Eighth International Conference on Computational Fluid Dynamics (ICCFD8), Chengdu, China, July 14-18, 2014

ICCFD8-2014-0344

Drag and drop simulation: from images to full 3D simulations

M. Bergmann[1,2], C. Galusinski[3], A. Iollo[1,2] & L. Weynans[1,2]

[1]Université de Bordeaux, France.
[2]INRIA Bordeaux sud ouest, France
[3]Université du Sud Toulon Var, France.

Corresponding author: michel.bergmann@inria.fr

Abstract : We present a suite of methods to achieve "drag and drop" simulation, i.e., to fully automatize the process to perform three-dimensional flow simulations around a body defined by actual images of moving objects. The overall approach requires a skeleton graph generation to get a level-set function from pictures, optimal transportation to get body velocity on the surface and then flow simulations thanks to a cartesian method based on penalization. We illustrate this paradigm by simulating the swimming of a mackerel fish.

Keywords : Drag and Drop Simulation, Cartesian mesh, Penalization, Level Set, Skeleton, Optimal transportation.

1 Introduction

The aim of this study is to devise a methodology for drag and drop simulations. The idea is to perform full three dimensional fluid flow simulations around deforming bodies defined by some images (here, a mackerel fish). The flow solver is based on cartesian grids, the body is defined by the zero of a level set function [1] and it is modeled thanks to a second order penalization method [2, 3] with the body velocity. Several numerical simulations of deforming bodies where the geometry is analytically prescribed has been done ([3, 4]). As we will explain in what follows, it is necessary to know the body level set function and surface deforming velocity at each simulation time step.

2 Modeling and numerical method

The flow configuration is given in figure 1 where Ω_i is domaine defined by body i, Ω_f is the domain filled by the fluid, and $\Omega = \Omega_f \cup \Omega_s^i$ is the whole domain. In what follows, we will simplify the problem by only considering one body.

The flow can be modeled using the incompressible Navier-Stokes equations written in the fluid domain Ω_f, where $D(\boldsymbol{u}) = \frac{\boldsymbol{\nabla u} + \boldsymbol{\nabla}^T \boldsymbol{u}}{2}$:

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

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

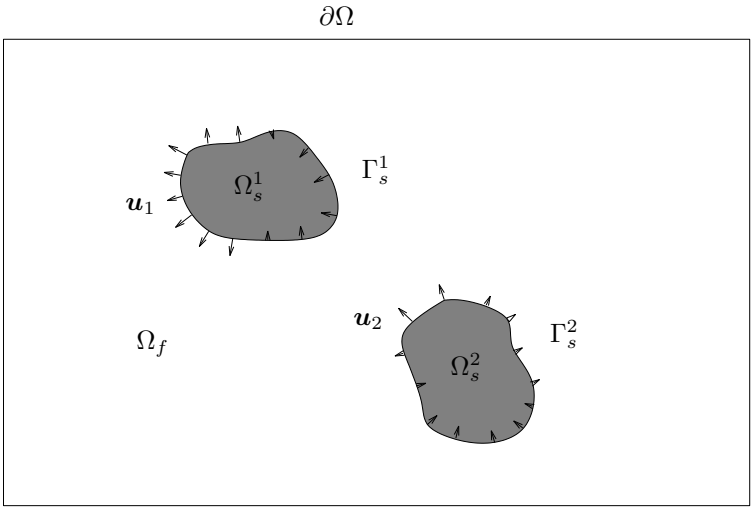


FIGURE 1 – Sketch of the general flow configuration.

with boundary conditions on the body boundary

$$\boldsymbol{u}(\boldsymbol{x},\, t) = \widehat{\boldsymbol{u}}(\boldsymbol{x},\, t) \text{ on } \Gamma_s. \tag{1c}$$

with initial condition in Ω_f and boundary conditions on $\partial\Omega$. Imposing the unsteady boundary condition on the body boundary is not straightforward. On the other hand, it is simple if the equations are discretized in space using a body fitted mesh. However, these kind of meshes are difficult to generate on complex geometries and mesh adaptation is required when the body moves. In this study we will consider the penalized Navier-Stokes equations in the whole domain Ω (fluid and body)

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

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0 \quad \text{in } \Omega, \tag{2b}$$

Since the boundary conditions on the body boundary are not explicitely imposed, a fixed mesh can been used and we chose to use an uniform cartesian mesh. The boundary conditions on the body boundary are implicitely imposed through the penalty term $\chi\lambda(\hat{\boldsymbol{u}} - \boldsymbol{u})$, where χ is a characteristic fonction ($\chi(\boldsymbol{x}) = 1$ if $\boldsymbol{x} \in \Omega_s$ and $\chi(\boldsymbol{x}) = 0$ elsewhere), λ is a large penalty parameter (we chose $\lambda = 10^8$). In what follows, the characteristic fonction will be computed from the sign distance fonction ψ, so that $\chi = H(\psi)$ where H denotes the Heaviside function.

It has been proven the solution of the system (2) tends to the solution of the system (1) with respect to the parameter $1/\lambda$.

The resolution of system (2) requires knowledge of both the unsteady characteristic function χ and the moving velocity $\hat{\boldsymbol{u}}$. The following section will explain how to compute these quantities from a series of images.

The system (2) is solved in time using a projection method ([5, 6]) and is discretized in space on a uniform cartesian mesh using second order centered finite differences up the third-order upwind finite differences for the convective term.

3 Methodology to get the level set and the deforming velocity

The aim of this section is to present the methodology to perform numerical simulations from body geometry snapshots. Here snapshots are given by some pictures of a real fish. The path to perform fully three-dimensional simulations form two-dimensional images (photographies) is the following. Firstly, we will generate a 3D profile from 2D images. To do this we use the skeleton approach. From the skeleton (that

we can deform using a swimming law) we can build a level set function using some realistic assumptions. We can obtain several 3D level set functions corresponding to a swimming profile deforming the skeleton (backbone) with a swimming law. Secondly, the velocity transformation form a level-set function to the next one is also required for penalization. In general, we only have a limited number of level-set snapshot and it is thus necessary to build intermediate ones. To summarize this step we need the transformation velocity and intermediate level set functions. Optimal transport is an adapted method to get this.

A full three-dimensional numerical simulation can be achieved knowing the level set function and the penalization velocity. Once we will get the level-set function and the penalization velocity we will be able to perform a full three dimensional numerical simulation.

3.1 Skeleton technics to build 3D level set functions

The skeleton method used here is described in detail in ([7]). We only present the main steps. We start from pictures (front, top and sides views) for a mackerel fish (see figure 2). We then perform a segmentation to extract the contour of each image. From these contours we can build level set functions.

If necessary, top and face views can be recentered as illustrated in figure 3. Skeleton method requires

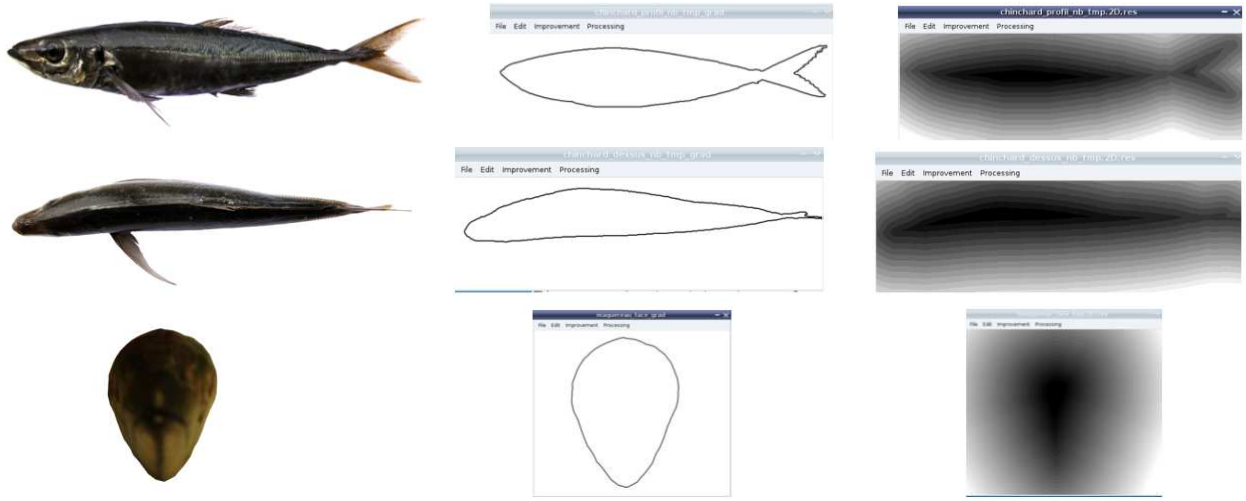


FIGURE 2 – Skeleton steps : photographies (left), segmentation and contours extraction (center), and associated level set (right).

a symmetry for top and front views, and the lateral tails are then removed. The undeformed profile is

FIGURE 3 – Contour recentering (top view).

represented as beeing the isocontour zero of the level set function (see figure 4). The swimming profile is obtained deforming the backbone (median skeleton, see figure 3) using the swimming law (3) :

$$y(x,\, t) = a(x)\sin(2\pi(x/\lambda + ft)), \tag{3a}$$

with

$$a(x) = c_0 + c_1 x + c_2 x^2. \tag{3b}$$

We chose the same swimming parameters as in [8]. Several level set functions corresponding to different time

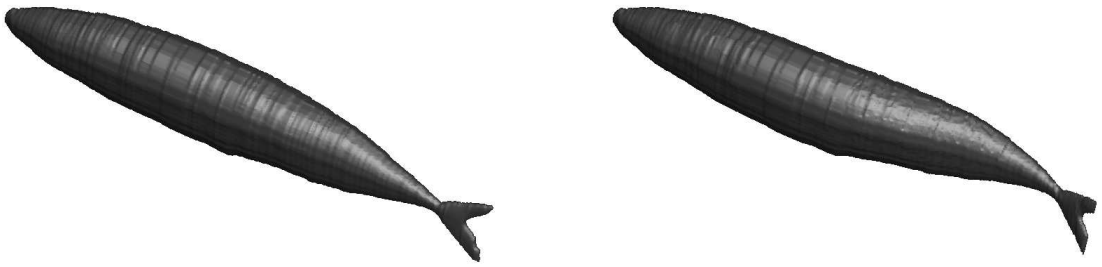


FIGURE 4 – Examples of the 3D profils. Undeformed (left) and deformed (right).

t can be generated using (3). In what follows we generate 20 snapshots for the level set function uniformly taken over une swimming stroke. We took care to remove the artificial deplacement of the center of mass due to deformation.

3.2 Optimal tranport

We get thus 20 snapshots of the level set function $\left\{\psi^k\right\}_{k=0,\ldots,19}$. Without loss of generality, the method is illustrated with ψ^0 and ψ^1. We look for the transformation (velocity field) from ψ^0 to ψ^1. This field will also allow to build some intermediate level set functions ψ^ℓ for $0 < \ell < 1$ corresponding to the temporal discretization of the Navier-Stokes equations. This problem had initially been introduced by Monge [9]. The numerical resolution of that kind of problem is difficult and we chose the lagrangian method introduced in [10].

We study the L^2 solution of the Monge-Kantorovich Problem (MKP). Let $\rho_0(\xi), \rho_1(x)$ two positive function with compact supports Ω_0 and Ω_1, where $\xi, x \in \mathbb{R}^d$ and d is the dimension of the physical problem ($d = 3$ in this study). We suppose that

$$\int_{\Omega_0} \rho_0(\xi) d\xi = \int_{\Omega_1} \rho_1(x) dx.$$

Let $X : \Omega_0 \to \Omega_1$ be a regular transform such that $X(\xi)$ is a transfert between ρ_0 and ρ_1. We then obtain the Jacobian equation :

$$\rho_0(\xi) = \det(\nabla X(\xi)) \rho_1(X(\xi)).$$

This equation is under determined with respect to $X(\xi)$ and we chose the transformation inducing the Kantorovich-Wasserstein L^2 distance :

$$\inf_X \int_{\Omega_0} \rho_0(\xi) |X(\xi) - \xi|^2 d\xi. \tag{4}$$

This L^2 MKP corresponds to determine the transformation X^* satisfying (4). It has been proven ([11, 12, 13]) that this problem has a unique solution that is the gradient of a convexe function $\Psi : \Omega_0 \to \mathbb{R}$:

$$X^*(\xi) = \nabla \Psi(\xi).$$

The first class of methods to solve this kind of problem is based on the solution of the Monge-Ampere Equation (MAE) :

$$\rho_0(\xi) = \det(\nabla^2 \Psi(x)) \rho_1(\nabla \Psi(x)).$$

The problem of the equation is that the boundary conditions are still unknown.

The second class of methods is based on arguments form continuous mechanics ([14]). A ficticious time [1] is introduced and we have $\Pi : [0,1] \times \Omega_0 \to \mathbb{R}^d$, with $\Pi(0,\xi) = \xi$, $\Pi(1,\xi) = X(\xi)$, $x = \Pi(t,\xi)$ and $\partial_t \Pi = v(t,x)$, and MKP is solution of

$$\inf_{\rho,v} \int_{\mathbb{R}^d} \rho(t,x)|v(t,x)|^2 \, dx,$$

where the minimum is look over all densities $\rho(t,x) \geq 0$ and velocity fileds$v(t,x) \in \mathbb{R}^d$ satisfying the continuity equation

$$\partial_t \rho + \nabla \cdot (\rho v) = 0, \tag{5}$$

with initial and final conditions :

$$\rho(0,\cdot) = \rho_0, \ \rho_1(1,\cdot) = \rho_1.$$

The resolution of this minimization problem can be expensive in the numerical point of vue. We then chose a lagrangian method [10] based on Picard iterations. We can start from a transformation fields that is a perturbation of the optimal one, and then linearize and iterate near the field. If the two images are quite close one to each other, we observed that the field zero is acceptable. We thus initially chose $\Psi^0 = 0$ and $X^0(\xi) = \xi$, where ξ is the coordinates of the fixed mesh where the images are defined. The algorithm is the following :

Algorithm : for a given iteration n, we compute the image $\rho_0^n(\xi)$ of the quantity $\rho_1(X^n(\xi))$ via the transformation $X^n(\xi)$

$$\rho_0^n(\xi) = \rho_1(X^n(\xi)) \det \nabla_\xi X^n(\xi).$$

We should obtain $\rho_0^n(\xi) = \rho_0(\xi)$ for $n \to \infty$. At the first iteration $n = 0$, we have $\rho_0^n(\xi) = \rho_1(X^n(\xi))$. We defined a convergence criterion with a given norm $|\rho_0^n(\xi) - \rho_0(\xi)| \leq \epsilon$. If this criterion is not satisfied we computed a correction Ψ^n solving :

$$\rho_0^n(\xi) - \rho_0(\xi) = \nabla_\xi \cdot (\rho_1(X^n(\xi)) \nabla_\xi \Psi^n), \tag{6}$$

and then we update the transformation

$$X^{n+1} = X^n - \alpha \nabla \Psi^n.$$

The parameter $0 \leq \alpha \leq 1$ depends on the images. If two images are quite close from each other we can chose $\alpha = 1$. In our case we chose $\alpha = 0.5$. We let $n = n + 1$ and the algorithm is stopped when the convergence criterion is satisfied. To get a well posed problem (6) we regularize the densities ρ_0 and ρ_1 adding a small value ε to avoid a zero value.

Choice of the densities : Since the level set functions (sign distance functions) are not strictly positive they are not adapted for optimal transport. We have also tested a mask function. However this choice is not adapted as well since the caudal take is almost not taken into account. We then consider a gaussian functions near the interface and we regularize these functions with $\varepsilon = 10^{-4}$.

Deformation velocity : Let Δt_s by the time step between two successive images. The deformation velocity from image ρ_0 to image ρ_1 is $\widetilde{\boldsymbol{u}}^0 = X^{opt}/\Delta t_s$. The exponent 0 highlights the fact that the center of mass do not move during the deformation.

Construction of intermediate images : For $X = X^{opt} + \xi$ we have ρ_0 and for $X = \xi$ we have ρ_1. Let τ be a parameter defining the transfert form ρ_0 to ρ_1. The transformation for a fixed τ is $X_\tau = \xi + (1-\tau) X^{opt}$. The image is then ρ_τ computed with

$$\rho_\tau(\xi) = \rho_1(X_\beta(\xi)) \det \nabla_\xi X_\beta(\xi).$$

1. This time will not be ficticious ion our study and it will correspond to the time between two successive images..

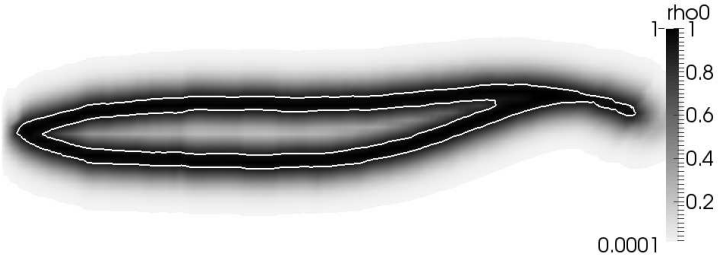


FIGURE 5 – Choice of the density. The profile is defined by the external blank lign. 2D visualization in the plan $z = 0$.

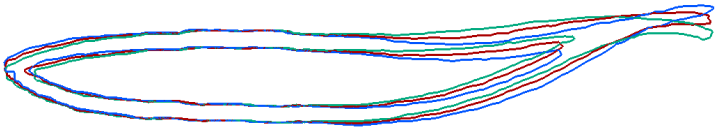


FIGURE 6 – Reconstruction of an intermediate image for $\tau = 0.5\Delta t_s$ (red) between ρ_0 (green) et ρ_1 (blue). 2D visualization in the plan $z = 0$.

Generalization : We show how to compute the deformation velocity form density ρ_0 to density ρ_1, and then to build intermediate densities ρ_τ. This can be generalized to a series of densities $\{\rho_i\}_{i=0}^{Ni}$ computing the deformation velocities from ρ_i to ρ_{i+1}. From the density $\rho_\tau(\xi,\, t)$ we can compute a mask and a level set function (by redistanciation). In what follows we note $\psi^0(\xi,\, t)$ the level set functions of the deformed profile and we note $\widetilde{\boldsymbol{u}}^0(\xi,\, t)$ the deformation velocity.

4 Numerical simulation of swimming

Computation of the position : The position of the self-propelled fish is characterized by the level-set function $\psi(\xi,\, t)$ that is the image of the level set function $\psi^0(\xi,\, t)$ using the transformation of the rigid motion (translation plus rotation) noted $X^r(\xi,\, t)$. We thus have :

$$\psi(\xi,\, t) = \psi^0(X^r(\xi,\, t)).$$

In the same way the velocity of the body is $\boldsymbol{u}^0(\xi,\, t) = \overline{\boldsymbol{u}}(\xi,\, t) + \boldsymbol{u}^\theta(\xi,\, t) + \widetilde{\boldsymbol{u}}^0(\xi,\, t)$. Since the position has been modified, this velocity has to act on the modified position and we have :

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

The transformation $X^r(\xi)$, and the translation $\overline{\boldsymbol{u}}(t)$ and rotation $\boldsymbol{u}^\theta(\xi,\, t)$ velocities are computed from Newton's laws computing the forces and the torques exerted by the flow on the body.

Computation of forces and torques We consider an arbitrarily domain Ω_{f_i} containing the obstacle Ω_s (see figure 7) such that :

$$\begin{aligned}\boldsymbol{F}_i = -\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_{f_i}(t)} \boldsymbol{u}\,\mathrm{d}V + \int_{\partial\Omega_{f_i}(t)} (\mathbb{T} + (\boldsymbol{u}-\boldsymbol{u}_i)\otimes\boldsymbol{u})\,\boldsymbol{n}_i\,\mathrm{d}S \\ - \int_{\partial\Omega_i(t)} ((\boldsymbol{u}-\boldsymbol{u}_i)\otimes\boldsymbol{u})\,\boldsymbol{n}_i\,\mathrm{d}S.\end{aligned} \tag{7a}$$

$$\begin{aligned}\boldsymbol{\mathcal{M}}_i = -\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_{f_i}(t)} \boldsymbol{r}_i\wedge\boldsymbol{u}\,\mathrm{d}V + \int_{\partial\Omega_{f_i}(t)} \boldsymbol{r}_i\wedge(\mathbb{T} + (\boldsymbol{u}-\boldsymbol{u}_i)\otimes\boldsymbol{u})\,\boldsymbol{n}_i\,\mathrm{d}S \\ - \int_{\partial\Omega_i(t)} \boldsymbol{r}_i\wedge((\boldsymbol{u}-\boldsymbol{u}_i)\otimes\boldsymbol{u})\,\boldsymbol{n}_i\,\mathrm{d}S.\end{aligned} \tag{7b}$$

The integrals over $\partial\Omega_i$ in (7a) and (7b) vanish in most of applications ($\boldsymbol{u} = \boldsymbol{u}_i$), except if blowing and/or suction is applied on that boundary.

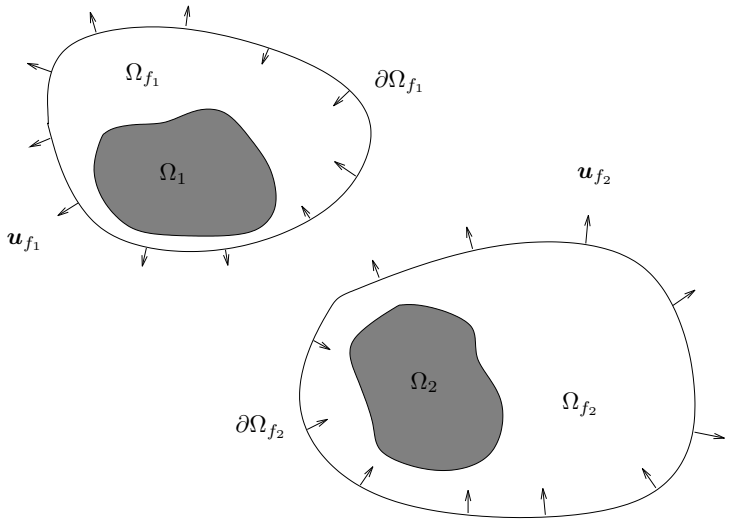


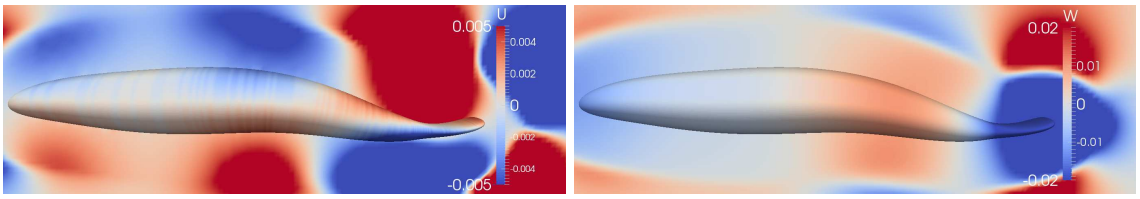
FIGURE 7 – Sketch of the domain used to compute the forces and the torques.

Validation : we compare two numerical simulations where the geometry of the fish is known at each time step. :

- Case I ("Lagrangian") : the level set function and the deformation velocities are given at each time step
- Case II ("Eulerian") : we suppose that we only have 20 snapshots of the geometry over one swimming stroke. We thus have to compute the transformation velocity (from one image to the following one) and the missing geometries using optimal transport.

In both cases the fish length is $\ell = 10\,cm$ and we chose the swimming law (3) with $f = 4\,Hz$, $c_0 = -0.002$, $c_1 = -0.12$, $c_2 = 2$ and $\lambda = \ell$.

A comparison of deformation velocity fields in case I (lagrangian) and case II (optimal transport) in presented in figure 8. Velocity componants U (in the swimming direction) and V (direction normal to the swimming) are presented for a 2D section in the plan $z = 0$. A good agreement is observed for the normal velocity V. The agreement is not so clear for the the component U. Due to the geometry of the profil, we can consider that the component U is approximatively tangential to the profil. Indded, optimal transport is not able to model the rotation (since deformation is potential). Other informations can helped the situation, as for instance repartition of the non constant density on the profil. Another solution could be to add constraints directly in the problem formulation. The swimming velocities obtained in both cases I and II are compared



(a) Velocity in the swimming direction U (b) Velocity normal to the swimming direction V

FIGURE 8 – comparison of deformation velocity fields in case I (lagrangian, inside the profil) and case II (optimal transport, outside the profil).

in figure 9. The normal velocity V shows good agreements in both cases. The velocity U is influenced by the differences observed in figure 8. However, these differences are quite small. After this validation we present

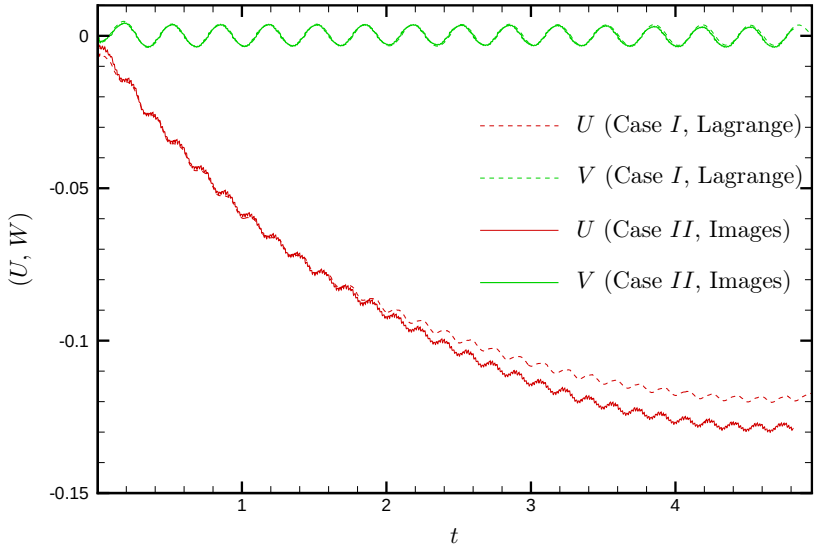


FIGURE 9 – Comparison of the swimming velocities obtained in lagrangian and eulerian ways.

a numerical simulation for a mackerel fish obtained from photographies.

Numerical results for a real mackerel fish Fgure 10 present the wake generated by a mackerel fish. The parameters are the same as those used in the previous section, the length is $10\,cm$ and the swimming frequency is $f = 4\,Hz$. The velocity of the mackerel is higher than those obtained in the previous section ($U \approx$ -0.15). This increase can be explained by a different profile and by the lunate tail. Further analysis are still necessary.

5 Conclusions and future work

The numerical simulations we show need to be improved when higher Reynolds numbers are considered. The next step is to perform mesh refinement and adaptation (patches or octrees).

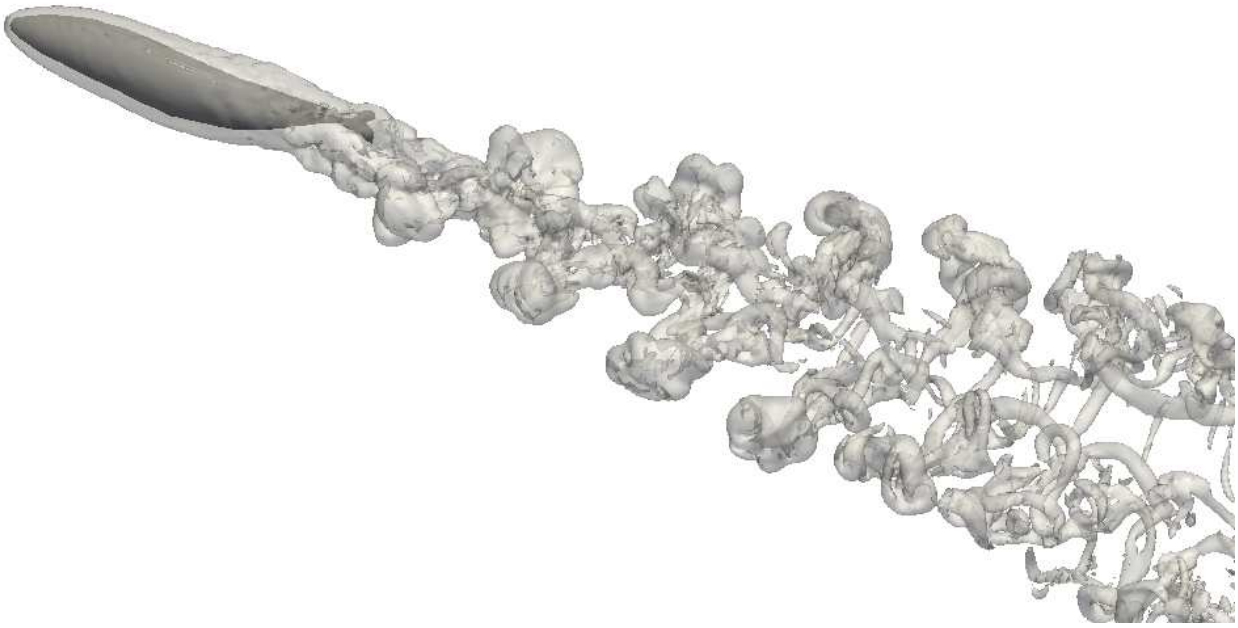


FIGURE 10 – Numerical simulation of a mackerel built from images. Iso contour of the norm of vorticity.

Références

[1] J. A. Sethian. *Level Set Methods and Fast Marching Methods.* Cambridge University Press, Cambridge, UK, 1999.

[2] P. Angot, C.H. Bruneau, and P. Fabrie. A penalization method to take into account obstacles in a incompressible flow. *Num. Math.*, 81(4) :497–520, 1999.

[3] M. Bergmann and A. Iollo. Modeling and simulation of fish-like swimming. *Journal of Computational Physics*, 230(2) :329 – 348, 2011.

[4] M. Bergmann, J. Hovnanian, and A. Iollo. An accurate cartesian method for incompressible flows with moving boundaries. *Communications in Computational Physics*, 15(5) :1266–1290, 2014.

[5] A.J. Chorin. Numerical solution of the Navier-Stokes equations. *Math. Comp.*, 22 :745–762, 1968.

[6] R. Temam. Sur l'approximation de la solution des equations de navier-stokes par la méthode des pas fractionnaires ii. *Archiv. Rat. Mech. Anal.*, 32 :377–385, 1969.

[7] J. Hovnanian. *Méthode de Frontières Immergées pour la Mécanique des Fluides. Application à la Simulation de la Nage.* PhD thesis, Université de Bordeaux 1, Bordeaux, France, 2012.

[8] D.S. Barrett, M.S. Triantafyllou, D.K.P. Yue, M.A. Grosenbauch, and M.J. Wolfgang. Drag reduction in fish-like locomotion. *J. Fluid Mech.*, 392 :182–212, 1999.

[9] G. Monge. Memoire sur la theorie des des déblais et des remblais. *Histoire de l'Academie des Sciences de Paris*, 1781.

[10] A. Bouharguane, A. Iollo, and L. Weynans. Numerical solution of the monge-kantorovich problem by picard iterations. Inria research report, RR-8477, 2014.

[11] Y. Brenier. Polar factorization and monotone rearrangement of vector–valued functions. *Communication in Pure and Applied Mathematics*, 64 :375–417, 1991.

[12] C. Villani. *Topics in optimal transportation.* American Mathematical Society, 1st edition, 2003.

[13] C. Villani. *Optimal Transport, old and new.* Springer-Verlag, 1st edition, 2009.

[14] J-D. Benamou and Y. Brenier. A computational fluid mechanics solution to the Monge–Kantorovich mass transfer problem. *Numerische Matematik*, 84 :375–393, 2000.