Modeling and numerical simulations of swimmers

Michel Bergmann

INRIA Bordeaux Sud-Ouest, project-team MC2

Institut de Mathématiques Appliquées de Bordeaux

33405 TALENCE cedex, France

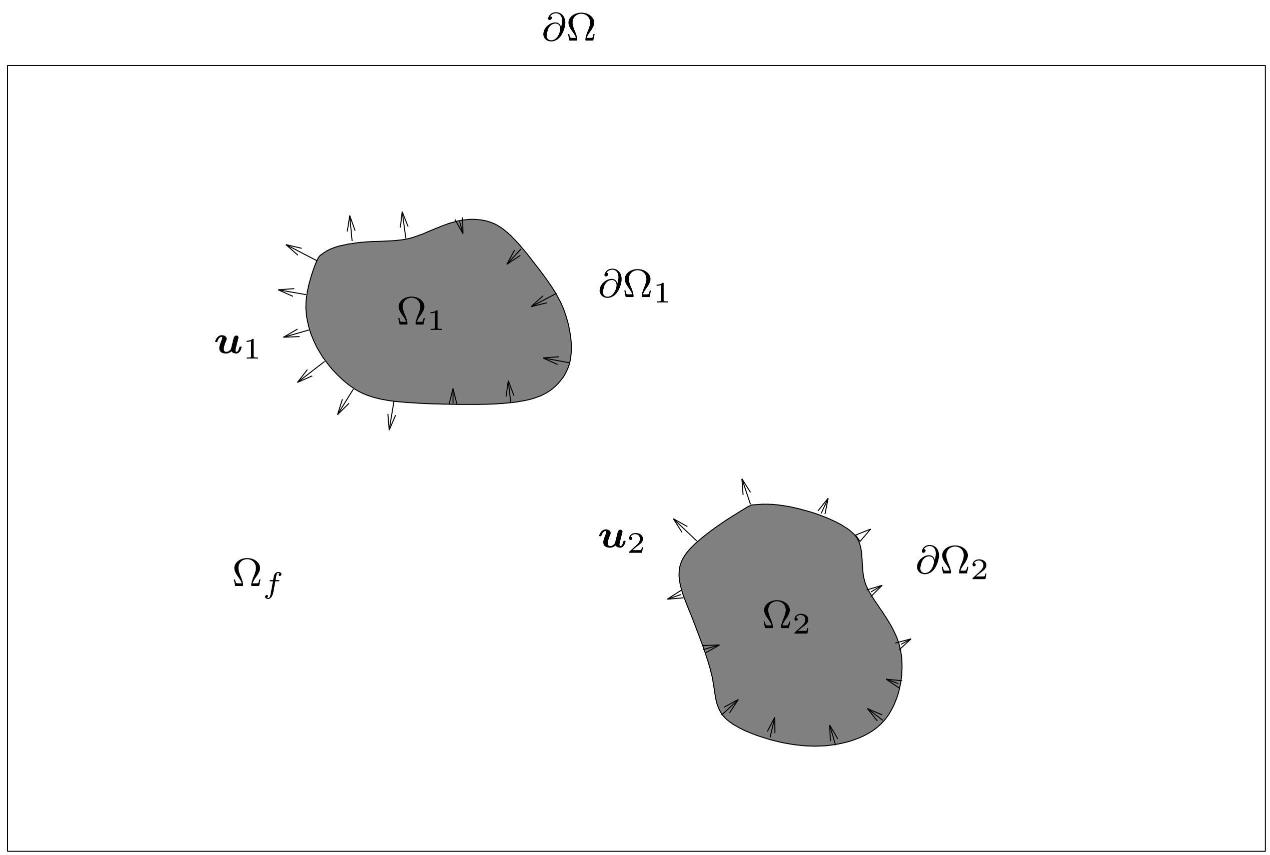
Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche BORDEAUX - SUD OUEST

Flow configuration



Ω_i : Domain "body" i

Ω_f : Domain "fluid"

$\Omega = \Omega_f \cup \Omega_i$: Entire domain

Flow modeling

- **Classical model:** Navier-Stokes equations (incompressible): $Re = \dfrac{\rho U_{ref} L}{\mu}$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla p + \frac{1}{Re}\Delta \boldsymbol{u} + \boldsymbol{g} \quad \text{in} \quad \Omega_f, \tag{1a}$$

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

$$\boldsymbol{u} = \boldsymbol{u}_i \quad \text{on} \quad \partial\Omega_i \tag{1c}$$

$$\boldsymbol{u} = \boldsymbol{u}_f \quad \text{on} \quad \partial\Omega \tag{1d}$$

Numerical resolution

Need of meshes that fit the body geometries

$\hookrightarrow$ Costly remeshing for moving and deformable bodies!!

Flow modeling

▶ **Penalization model:** penalized Navier-Stokes equations (incompressible):

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla p + \frac{1}{Re}\Delta \boldsymbol{u} + \boldsymbol{g} + \lambda \sum_{1=1}^{N_s} \chi_i(\boldsymbol{u}_i - \boldsymbol{u}) \quad \text{in} \quad \Omega, \tag{2a}$$

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

$$\boldsymbol{u} = \boldsymbol{u}_f \quad \text{on} \quad \partial\Omega. \tag{2c}$$

$\hookrightarrow \lambda \gg 1$ penalization factor $\rightarrow$ Solution eqs (2) tends to solution eqs (1) *w.r.t.* $\varepsilon = 1/\lambda \rightarrow 0$.

$\hookrightarrow \chi_i = H(\phi_i)$ where H is Heaviside function and ϕ_i the signed distance function ($\phi_i(\boldsymbol{x}) > 0$ if $\boldsymbol{x} \in \Omega_i$, $\phi_i(\boldsymbol{x}) = 0$, if $\boldsymbol{x} \in \partial\Omega_i$, $\phi_i(\boldsymbol{x}) < 0$ else if).

$$\frac{\partial \phi_i}{\partial t} + (\boldsymbol{u}_i \cdot \nabla)\phi_i = 0.$$

Numerical resolution

No need of meshes that fit the body geometries

$\hookrightarrow$ Cartesian meshes

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE — INRIA — centre de recherche BORDEAUX - SUD OUEST

Numerical approach | Method

- **Space:** Cartesian meshes,
  Centered FD 2nd order and upwind 3rd order for convective terms
  2^{nd} order penalization (immersed boundary like method)

- **Time:** 2^{nd} order Chorin scheme (Adams Bashforth/Crank Nicholson),
  Implicit penalization (larger λ)

$$\frac{\boldsymbol{u}^{(n+1)} - \boldsymbol{u}^{(n)}}{\Delta t} + \underbrace{(\boldsymbol{u}^{(n+\frac{1}{2})} \cdot \nabla)\boldsymbol{u}^{(n+\frac{1}{2})}}_{\text{Adams Bashforth}} = -\nabla p^{(n+\frac{1}{2}))} + \frac{1}{2\,Re}\underbrace{\left(\Delta \boldsymbol{u}^{(n+1)} + \Delta \boldsymbol{u}^{(n)}\right)}_{\text{Crank Nicholson}} + \boldsymbol{g}$$

$$+\lambda \sum_{1=1}^{N_s} \chi_i{}^{(n+1)}(\boldsymbol{u}_i{}^{(n+1)} - \boldsymbol{u}^{(n+1)}),$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u}^{(n+1)} = 0$$

$$(\boldsymbol{u}^{(n+\frac{1}{2})} \cdot \nabla)\boldsymbol{u}^{(n+\frac{1}{2})} = \frac{3}{2}(\boldsymbol{u}^{(n)} \cdot \nabla)\boldsymbol{u}^{(n)} - \frac{1}{2}(\boldsymbol{u}^{(n-1)} \cdot \nabla)\boldsymbol{u}^{(n-1)}$$

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE — INRIA — centre de recherche BORDEAUX - SUD OUEST

Numerical approach | Method

- **Chorin :** prediction (2nd order)

$$\frac{\boldsymbol{u}^{(*)} - \boldsymbol{u}^{(n)}}{\Delta t} = -\left[(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right]^{(n+\frac{1}{2})} - \nabla q + \frac{1}{2\,Re}\left(\Delta\boldsymbol{u}^{(*)} + \Delta\boldsymbol{u}^{(n)}\right)$$

$$A\boldsymbol{u}^{(*)} = \boldsymbol{b}\left(\boldsymbol{u}^{(n)},\, \boldsymbol{u}^{(n-1)}\right)$$

- **Chorin :** correction $\qquad \Delta\psi^{(n+1)} = \nabla\cdot\boldsymbol{u}^{(*)}$

$$\widetilde{\boldsymbol{u}}^{(n+1)} = \boldsymbol{u}^{(*)} - \nabla\psi^{(n+1)}$$

$$\widetilde{p}^{(n+1)} = q + \frac{\psi^{(n+1)}}{\Delta t} - \frac{\Delta t}{2\,Re}\Delta\psi^{(n+1)}$$

- **Compute body motion :**

$$\boldsymbol{u}_i^{(n+1)} = f(\widetilde{\boldsymbol{u}}^{(n+1)}, \widetilde{p}^{(n+1)}) \text{ and } \chi_i^{(n+1)} = \chi_i^{(n)} - \Delta t(\widetilde{\boldsymbol{u}}^{(n+1)}\cdot\nabla)\chi_i^{(n)}$$

- **Penalization step :** $\frac{\boldsymbol{u}^{(n+1)} - \widetilde{\boldsymbol{u}}^{(n+1)}}{\Delta t} = \lambda\sum_{1=1}^{N_s}\chi_i{}^{(n+1)}(\boldsymbol{u}_i{}^{(n+1)} - \boldsymbol{u}^{(n+1)})$

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche BORDEAUX - SUD OUEST

Numerical approach | Method

▶ **Compute body motion**

$$\boldsymbol{u}_i = \overline{\boldsymbol{u}}_i + \widehat{\boldsymbol{u}}_i + \widetilde{\boldsymbol{u}}_i$$

with:

$\overline{\boldsymbol{u}}_i$ translation velocity

$\widehat{\boldsymbol{u}}_i$ rotation velocity

$\widetilde{\boldsymbol{u}}$ deformation velocity (imposed for the swim)

$\hookrightarrow$ Compute forces $\boldsymbol{F}_i$ and torques $\boldsymbol{\mathcal{M}}_i$

$$m\frac{\mathrm{d}\overline{\boldsymbol{u}}_i}{\mathrm{d}t} = \boldsymbol{F}_i + m\boldsymbol{g}, \qquad \overline{\boldsymbol{u}}_i \quad \text{translation velocity,} \quad m \quad \text{mass}$$

$$\frac{\mathrm{d}J\boldsymbol{\Omega}_i}{\mathrm{d}t} = \boldsymbol{\mathcal{M}}_i, \qquad \boldsymbol{\Omega}_i \quad \text{angular velocity,} \quad J \quad \text{inertia matrix}$$

Rotation velocity $\widehat{\boldsymbol{u}}_i = \boldsymbol{\Omega}_i \times \boldsymbol{r}_G$ with $\boldsymbol{r}_G = \boldsymbol{x} - \boldsymbol{x}_G$ ($\boldsymbol{x}_G$ center of mass).

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE INRIA centre de recherche BORDEAUX - SUD OUEST

Numerical approach | Method

▸ **Classic way**

Stress tensor $\mathbb{T}(\boldsymbol{u},\, p) = -p\mathbb{I} + \frac{1}{Re}(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)$ and n outward normal unit vector at s_i:

$$\boldsymbol{F}_i = -\int_{\partial\Omega_i} \mathbb{T}(\boldsymbol{u},\, p)\, \boldsymbol{n}\, \mathrm{d}\boldsymbol{x},$$

$$\boldsymbol{\mathcal{M}}_i = -\int_{\partial\Omega_i} \mathbb{T}(\boldsymbol{u},\, p)\, \boldsymbol{n} \times \boldsymbol{r}_G\, \mathrm{d}\boldsymbol{x}.$$

Evaluation of forces and torques

Cartesian mesh: no direct acces to $\partial\Omega_i$

$\hookrightarrow$ Not easy evaluation ....

Numerical approach | Method

▶ **Other way** : Arbitrarily domain $\Omega_{f_i}(t)$ surrounding body i.

Forces:

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

Torques:

$$\boldsymbol{\mathcal{M}}_i = -\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_{f_i}(t)} \boldsymbol{u}\times\boldsymbol{r}_G\,\mathrm{d}V + \int_{\partial\Omega_{f_i}(t)} (\mathbb{T} + (\boldsymbol{u}-\boldsymbol{u}_i)\otimes\boldsymbol{u})\,\boldsymbol{n}\times\boldsymbol{r}_G\,\mathrm{d}S - \int_{\partial\Omega_i(t)} ((\boldsymbol{u}-\boldsymbol{u}_i)\otimes\boldsymbol{u})\,\boldsymbol{n}\times\boldsymbol{r}_G\,\mathrm{d}S.$$

Evaluation of forces and torques

The term onto $\partial\Omega_i$ vanishes in our case (no transpiration)

$\hookrightarrow$ Easy evaluation!

Numerical approach | Method

- **Numerical resolution of Navier-Stokes equations (Cartesian mesh+Chorin)**

Developments and Experiments using Plafrim cluster

- Large scale 3D problems: more than 100 millions dofs

$\hookrightarrow$ Required parallel code

$\Rightarrow$ One solution: Message Passing Interface (MPI)

$\Rightarrow$ Other solution with higher abstraction level (more simple):

Portable, Extensible Toolkit for Scientific Computation (PETSc)

`http://www.mcs.anl.gov/petsc/petsc-as/`

$\hookrightarrow$ PETSc gives:

$\Rightarrow$ structures for parallelism (DA *Distributed Arrays* to manage cartesian meshes)

$\Rightarrow$ librairies to solve linear systems in parallel (KSP *Krylov Subspace methods*)

F-GMRES, preconditioner ASM with ILU on each subdomain

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE INRIA centre de recherche BORDEAUX - SUD OUEST

Fish like swimming | Fish motion

▶ **Body velocity i:**

$$\boldsymbol{u}_i = \overline{\boldsymbol{u}}_i + \widehat{\boldsymbol{u}}_i + \widetilde{\boldsymbol{u}}_i.$$

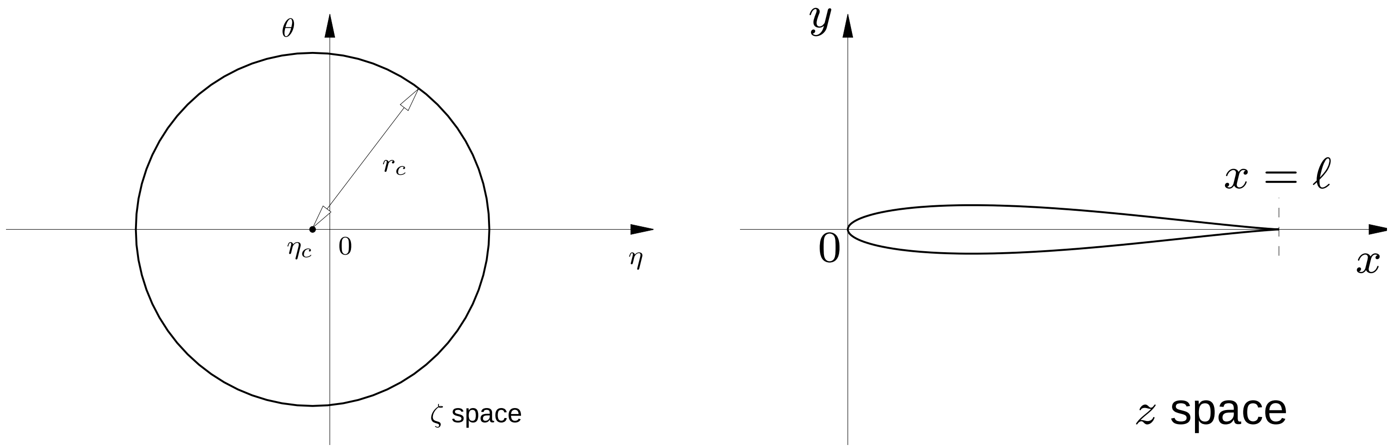
- Translation velocity $\overline{\boldsymbol{u}}_i$ is computed using forces $\boldsymbol{F}$
- Rotation velocity $\widehat{\boldsymbol{u}}_i$ is computed using torques $\boldsymbol{\mathcal{M}}$
- Deformation velocity $\widetilde{\boldsymbol{u}}_i$ is imposed for the swim

▷ Take care to not add artificial forces and torques!

1. Impose any backbone deformation,
2. Subtract body mass center motion due to deformation,
3. Rotate the body by the opposite angle generate by deformation,
4. Homothety for mass conservation

Fish like swimming | Steady fish shape

- **2D steady shape:** airfoil like profile



(a) Original shape (b) Steady shape

Fig. : *Sketch of the Karman-Trefftz transform. The z space is transformed to fit $0 \le x_s \le \ell$*

$$z = n \frac{\left(1+\frac{1}{\zeta}\right)^n + \left(1-\frac{1}{\zeta}\right)^n}{\left(1+\frac{1}{\zeta}\right)^n - \left(1-\frac{1}{\zeta}\right)^n},$$

$\Rightarrow$ **Only 3 parameters $\boldsymbol{b} = (\eta_c,\ \alpha,\ \ell)^T$**

$\triangleright$ $\alpha = (2-n)\pi$: tail angle

$\triangleright$ $\eta_c < 0$ circle center

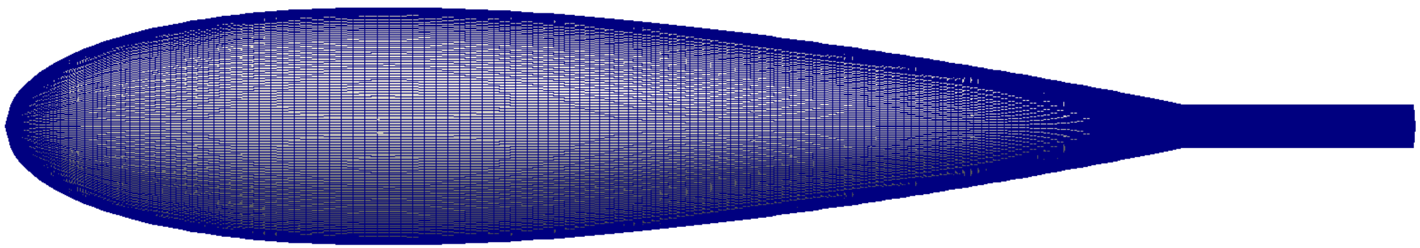
$\triangleright$ $\ell > 0$ fish length ($\ell = 1$)

Institut de Mathématiques de Bordeaux

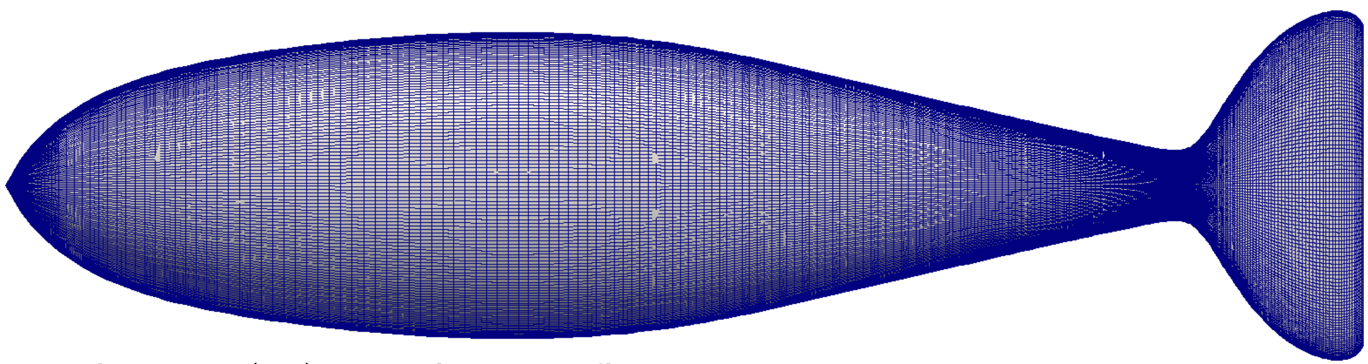
INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE INRIA centre de recherche BORDEAUX - SUD OUEST

Fish like swimming | Steady fish shape

- **3D steady shape:** ellipses centered on the backbone x_i, with axis $y(x_i)$ and $z(x_i)$.



$\hookrightarrow$ 2D shape: $y(x_i)$ is airfoil profil (+ tail)



$\hookrightarrow$ 3D shape: $z(x_i)$ B-splines profil

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE INRIA centre de recherche BORDEAUX - SUD OUEST

Fish like swimming | Swimming shape

$\hookrightarrow$ Backbone deformation in plan (x, y): $y(x,\, t) = (c_1 x + c_2 x^2)\sin(2\pi(x/\lambda + ft))$.

$\hookrightarrow$ 2D periodic, no artificial forces and torques,

$\hookrightarrow$ 3D each ellipse is orthogonal to the backbone $\Rightarrow$ mass conservation

Institut de
Mathématiques de
Bordeaux

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
BORDEAUX - SUD OUEST

Fish like swimming | Swimming shape

$\hookrightarrow$ Backbone deformation in plan (x, y): $y(x,\, t) = (c_1 x + c_2 x^2)\sin(2\pi(x/\lambda + ft))$.

$\hookrightarrow$ 2D periodic, no artificial forces and torques,

$\hookrightarrow$ 3D each ellipse is orthogonal to the backbone $\Rightarrow$ mass conservation

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

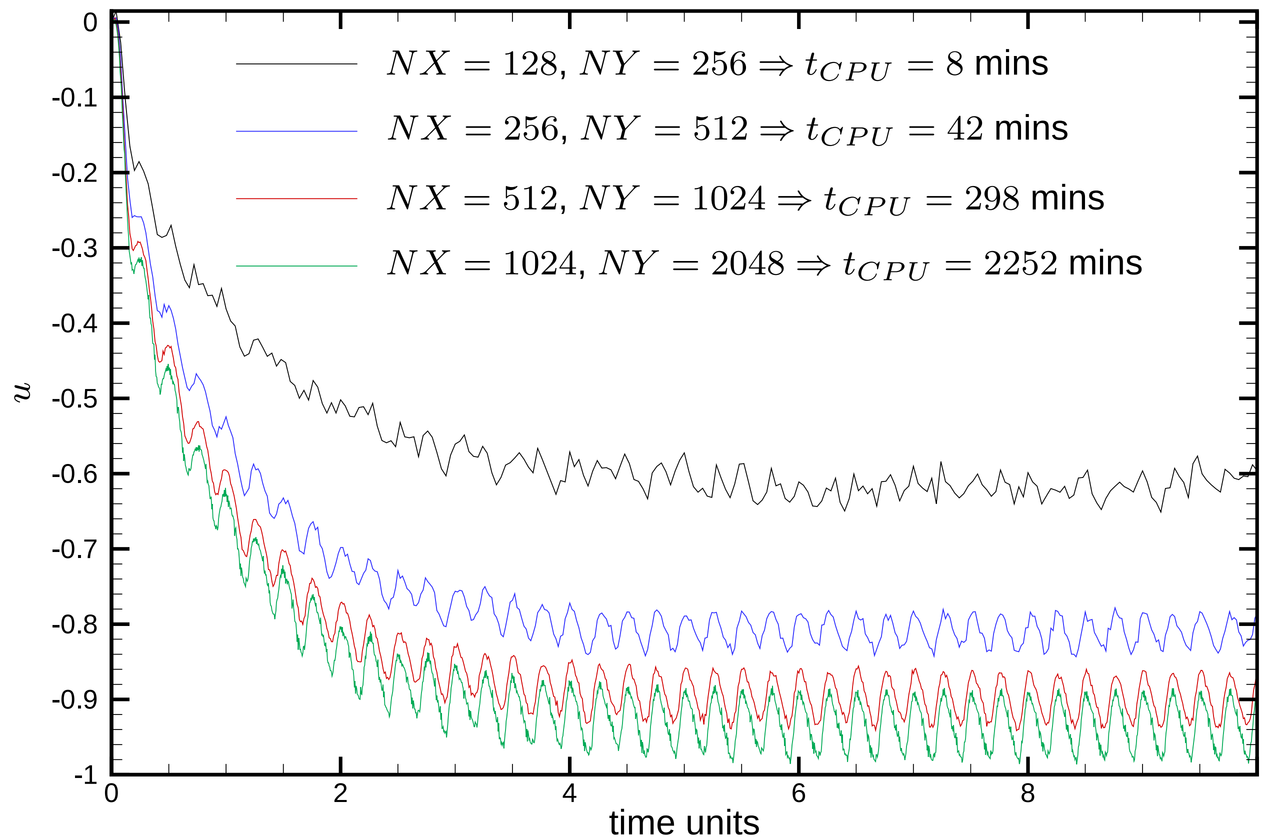
INRIA

centre de recherche BORDEAUX - SUD OUEST

Fish like swimming | Convergence study

- **Complex flow (deformable and moving body) $\Rightarrow$ convergence study (2D)**

  Parameter: u velocity depends on the forces, the torques, the mesh, ...



$\Rightarrow NX = 512$ and $NY = 1024$ seems good balance "precision vs.computation time".

$\hookrightarrow \Delta X = \Delta Y = 1/128$ are chosen in following simulations

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE INRIA centre de recherche BORDEAUX - SUD OUEST

Fish like swimming | Simulation

2D fish $Re = 1000$.

Fish like swimming | Simulation

3D fish $Re = 1000$*. Mesh* $1024 \times 128 \times 256 \approx 33 \cdot 10^6$ *nodes* $\Rightarrow 130 \cdot 10^6$ *unknowns.*

$\Rightarrow$ 3D and 2D wakes behavior are different

`S.Kern and P. Koumoutsakos, J Exp. Biology` **`209`**`, 2006.`

Fish like swimming | Turbulence simulation

Turbulence modeling: smagorinsky LES $Re = 10^6$

$\hookrightarrow$ Regulation of frequency and tail amplitude to reach a target swimming velocity $u_\tau = 0.56$.

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA centre de recherche BORDEAUX - SUD OUEST

Fish like swimming | Turbulence simulation

Turbulence modeling: smagorinsky LES $Re = 10^6$

$\hookrightarrow$ Regulation of frequency and tail amplitude to reach a target swimming velocity $u_\tau = 0.56$.

The optimized frequency $f \approx 1.1$ and double amplitude $A \approx 0.14$.

$\Rightarrow$ The Strouhal number $St = \frac{f\,A}{u_\tau} \approx 0.275$

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche BORDEAUX - SUD OUEST

Fish like swimming | Gray paradox?

Gray's paradox [1] :

"the power required for a dolphin of length $1.82m$ to swim at a speed of $10.1m/s$ is about seven times the muscular power available for propulsion (swimming more efficient than rigid body towed at same velocity)

$\hookrightarrow$ Paradox contested (J. Lighthill [2]) : fish power 3X higher

$\hookrightarrow$ Paradox "confirmed" experimentally at MIT (robot bluefin tuna) by Barret *et al.* [3]

[1] Gray J. (1936) : Studies in animal locomotion. VI. The propulsive power of the dolphin, *J. Exp. Biol.* **13** pp. 192-199.

[2] Lighthill, M.J. (1971) : Large amplitude elongated-body theory of fish locomotion, *Proc. R. Soc. Mech. B.* **179** pp. 125-138.

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

Fish like swimming | Gray paradox?

- **Power**

$$P(t) = \frac{\partial}{\partial t}\int_{\Omega_f} \frac{u^2}{2}\,\mathrm{d}\Omega + \frac{1}{Re}\int_{\Omega_f} \sigma'_{ij}\frac{\partial u_i}{\partial x_j},\mathrm{d}\Omega.$$

- **Propulsive index**

$$I_p = \frac{P_{tow}}{P_{swim}}.$$

- In turbulent case $Re = 10^6$ with $St = 0.275 \quad \Rightarrow \quad I_p = 0.35$, *i.e.* $P_{tow} < P_{swim}$

$\hookrightarrow$ No numerical evidence of Gray's paradox

$\hookrightarrow$ Possible reasons:

– is does no exist!

– numerical errors: LES model, cartesian meshes (boundary layers?) $\rightarrow$ error friction?, ..

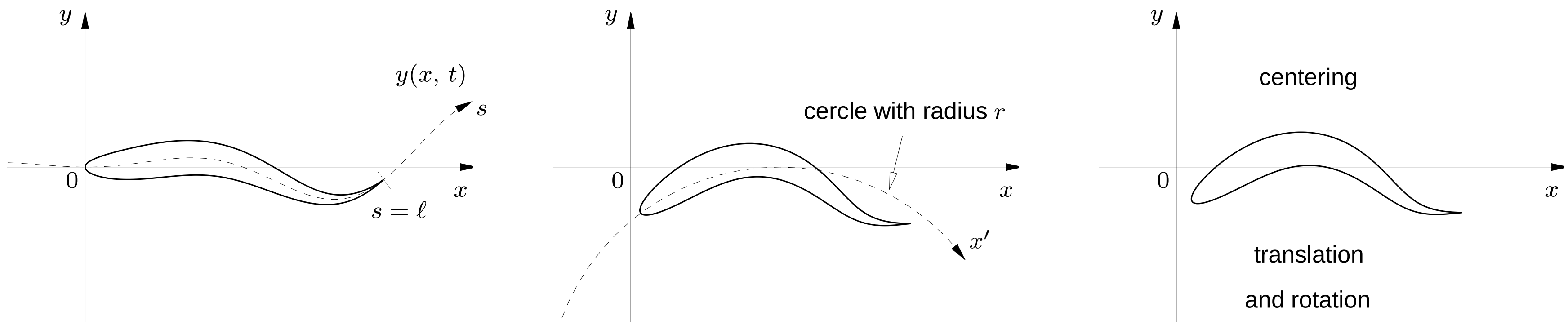
Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE INRIA centre de recherche BORDEAUX - SUD OUEST

Fish like swimming | Maneuvers

Example: predator/prey $\Rightarrow$ reach food

Method: add mean curvature r



(c) Swimming shape (d) Maneuvering shape (e) Real motion shape

Fig. : *Sketch of swimming and maneuvering shape.*

Question: adaptation of $r(t)$?

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche BORDEAUX - SUD OUEST

Fish like swimming | Maneuvers

Idea: adapt r using "angle of vision" θ_f, i.e. $r = r(\theta_f)$:

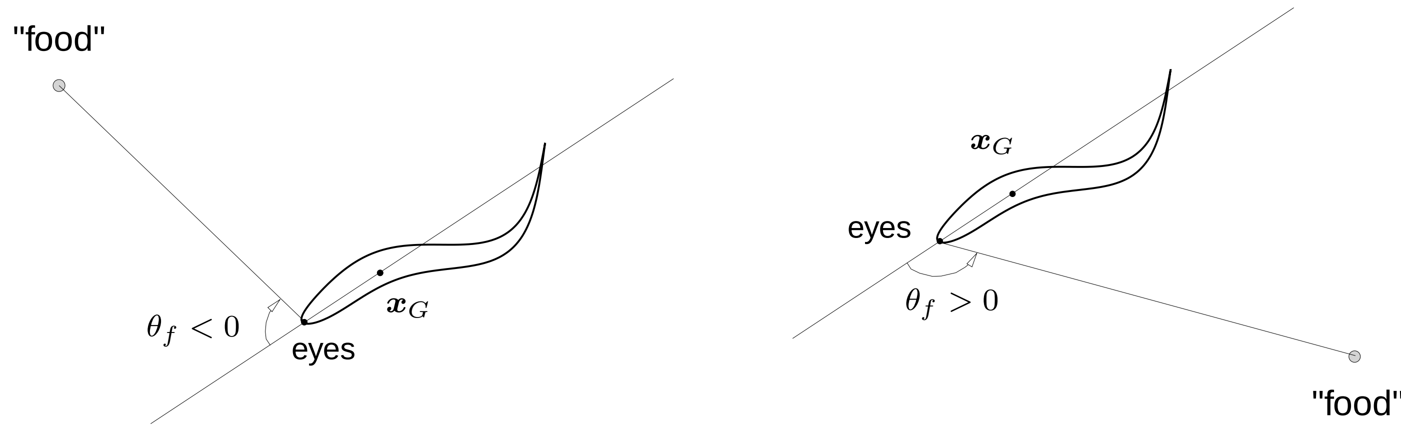


Fig. : *Sketch of the oriented food angle of vision.*

$$r(\theta_f) = \begin{cases} \infty & \text{if } \theta_f = 0, \\ \overline{r} & \text{if } \theta_f \geq \overline{\theta_f}, \\ -\overline{r} & \text{if } \theta_f \leq -\overline{\theta_f}, \\ \overline{r}\left(\dfrac{\overline{\theta}}{\theta_f}\right)^2 & \text{otherwise.} \end{cases}$$

We impose $|r| \geq \overline{r}$ and $|\theta_f| \geq \overline{\theta_f}$. We chose arbitrarily $\overline{r} = 0.5$ and $\overline{\theta} = \pi/4$.

Fish like swimming | Maneuvers

2D fish $Re = 1000$

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche BORDEAUX - SUD OUEST

Fish like swimming | Maneuvers

3D fish $Re = 1000$. *Mesh* $512 \times 128 \times 512 \approx 33 \cdot 10^6$ *nodes* $\Rightarrow 130 \cdot 10^6$ *unknowns.*

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE | INRIA | centre de recherche BORDEAUX - SUD OUEST

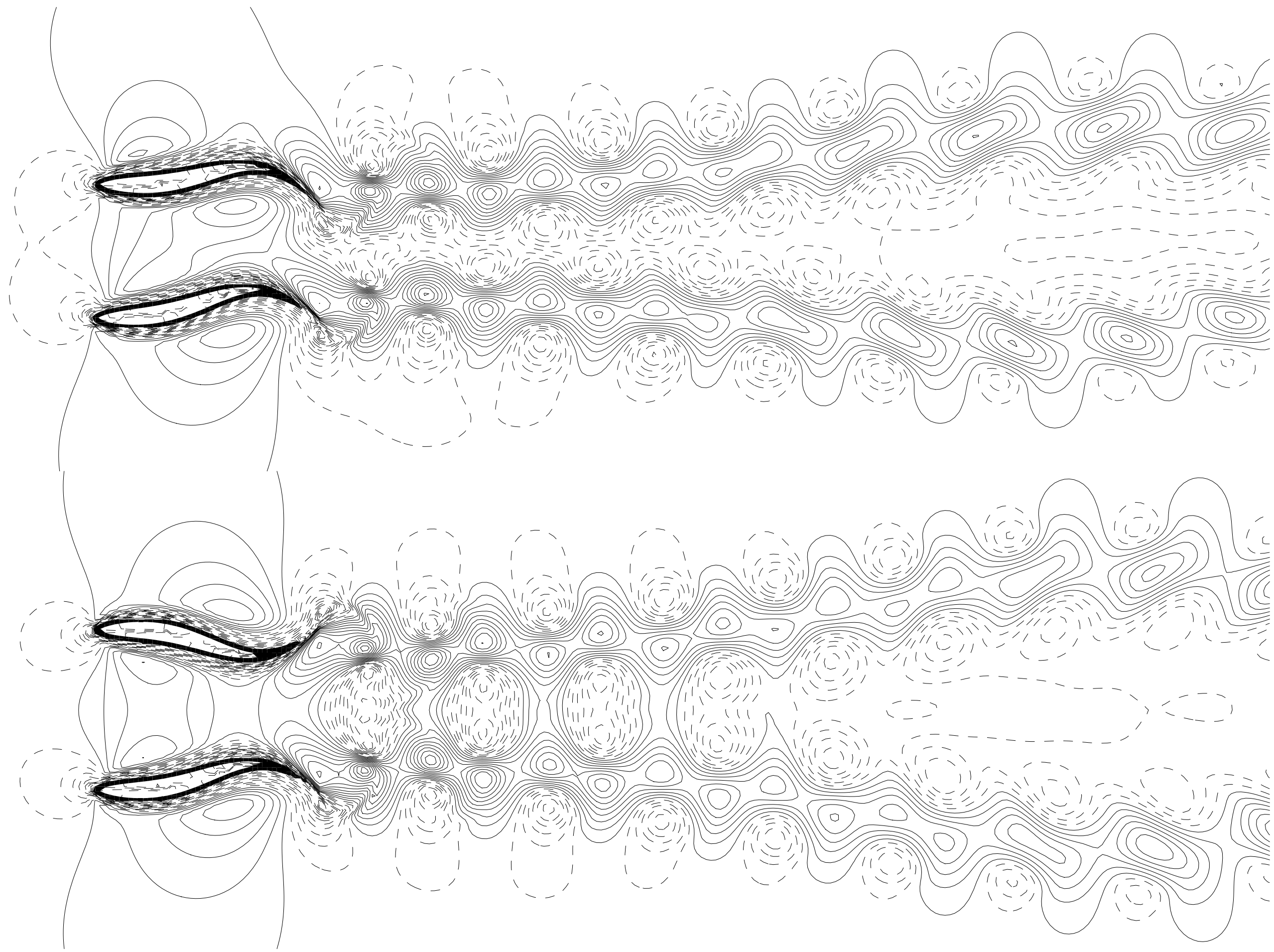
Fish like swimming | Schooling

- **Configuration:** school limited to 3 fishes

↪ **Preliminary study** 2 fishes with parallel swim

Velocity u
Phase

Velocity u
Anti-phase



Fish like swimming | Schooling

- **Observation:** existence of a zone in the wake where flow has same velocity sign than fishes velocities

- **Idea:** put a third fish in this zone with "potential benefits"

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche BORDEAUX - SUD OUEST

Fish like swimming | Schooling

- **Observation:** existence of a zone in the wake where flow has same velocity sign than fishes velocities
- **Idea:** put a third fish in this zone with "potential benifits"

Anti-phase. $\Rightarrow$ Quite efficient.

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE INRIA centre de recherche BORDEAUX - SUD OUEST

Fish like swimming | Schooling

- ▶ **Observation:** existence of a zone in the wake where flow has same velocity sign than fishes velocities
- ▶ **Idea:** put a third fish in this zone with "potential benifits"

Phase. $\Rightarrow$ Very efficient.

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche BORDEAUX - SUD OUEST

Fish like swimming | Schooling

- **Goal:** save energy

↪ adapt velocity of the third fish

(regulation of tail amplitude A to reach same velocity than two other fishes)

	Phase				Anti-phase			
L D	0.4	0.5	0.6	0.7	0.4	0.5	0.6	0.7
1.5	15.0	16.3	11.1	7.1	6.8	6.9	9.8	7.1
2.0	10.1	14.5	9.8	6.0	6.8	6.1	9.8	6.0
2.5	8.4	13.6	9.0	5.1	6.7	5.3	9.0	5.1
3.0	15.0	15.1	6.9	5.0	5.2	5.1	7.0	3.2
3.5	5.2	13.2	6.2	2.2	4.9	5.0	6.2	0.5

Tab. : *Percentage of energy saved for the three fishes school in comparison with three independent fishes.* $Re = 10^3$.

The 3 fishes school can save an amount around 15% of total energy!!

Fish like swimming | Schooling

3D fishes $Re = 1000$. *Mesh* $1024 \times 128 \times 256 \approx 33 \cdot 10^6$ *nodes* $\Rightarrow 130 \cdot 10^6$ *unknowns.*
$\Rightarrow$ No efficient effect for 3^{rd} fish. 3D wake $\neq$ 2D wake (no inverted VK street)

Institut de Mathématiques de Bordeaux

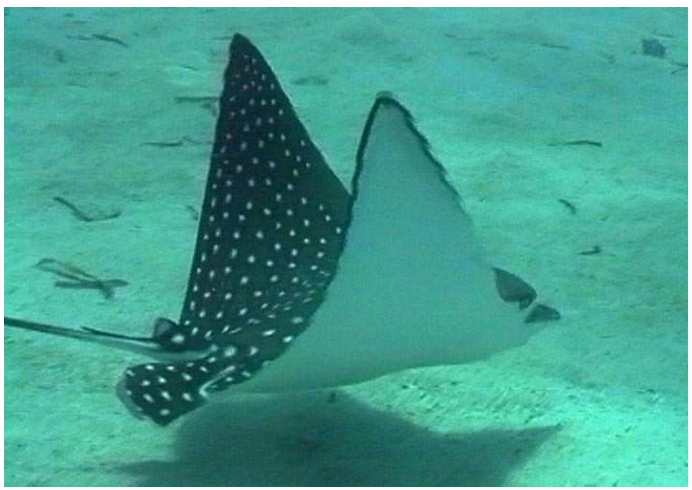
INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE — INRIA — centre de recherche BORDEAUX - SUD OUEST

Fish like swimming | First ray model

Fish like swimming | First ray model

3D ray-like swimming with flapping wings at $Re = 10^4$.

Fish like swimming | Second ray model



Fish like swimming | Second ray model

3D ray-like swimming with flapping wings at $Re = 10^4$.

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche BORDEAUX - SUD OUEST

Fish like swimming | Simplified Jellyfish model

Fish like swimming | Simplified Jellyfish model

3D jellyfish $Re = 1000$*. Mesh* $256 \times 256 \times 512 \approx 33 \cdot 10^6$ *nodes* $\Rightarrow 130 \cdot 10^6$ *unknowns.*

$\Rightarrow$ Velocity very close to 2D case (quasi axisymmetric)

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche BORDEAUX - SUD OUEST

Other 3D simulations | Wind turbine

Wind turbine: photo

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE
INRIA
centre de recherche BORDEAUX - SUD OUEST

Other 3D simulations | Wind turbine

Wind turbine: modeling (size 3m)

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche BORDEAUX - SUD OUEST

Other 3D simulations | Wind turbine

Wind turbine: rotation (20 rpm)

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche BORDEAUX - SUD OUEST

Other 3D simulations | Wind turbine

Wind turbine: numerical simulation (wind 3 m/s, $Re = 10^4$)

Mesh $350 \times 350 \times 350 \approx 42 \cdot 10^6$ *points* $\Rightarrow 170 \cdot 10^6$ *unknowns.*

Institut de
Mathématiques de
Bordeaux

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
BORDEAUX - SUD OUEST

Other 3D simulations | Wind turbine

Wind turbine: numerical simulation (wind 3 m/s, $Re = 10^6$, Smagorinsky turbulence model)

Mesh $350 \times 350 \times 350 \approx 42 \cdot 10^6$ *points* $\Rightarrow 170 \cdot 10^6$ *unknowns.*

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche BORDEAUX - SUD OUEST

Next ....

- **Fluid-Structure interactions & elasticity** (eulerian, post doc Thomas Milcent)

  $\hookrightarrow$ Model the tail/fins

  $\rightarrow$ Example: cylinder motion imposed by penalization with free motion of the "tail"

Institut de Mathématiques de Bordeaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche BORDEAUX - SUD OUEST