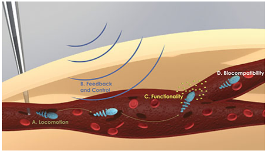
Navigation of Flagellated Micro-Swimmers

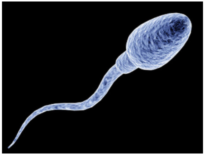
Laetitia Giraldi
with C. Van Landeghem, L. Berti, C. Prud'homme, R. Chesneaux, Z. El Khiyati and J. Bec

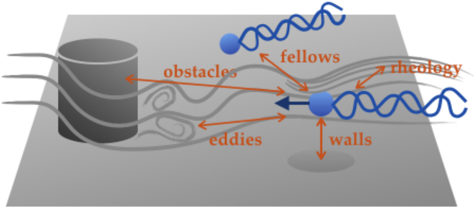
July, 2023

Challenges of micro-swimming



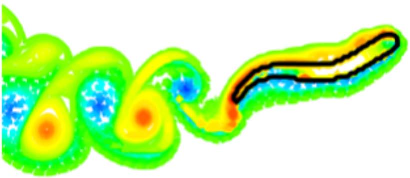
- Medical applications : driving micro-robots through our body
- Multidisciplinary fields (Maths, Physics , Biology and Robotic)
- Biomimetism





Controlling

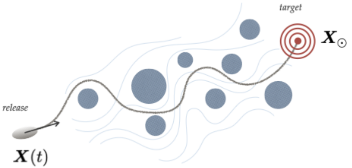
Locomotion : Design realistic, complicated model (elasticity, fluid-structure interactions, etc.)



⇒ find optimal gait for displacement

Alouges et al. (2013), Ghaffari et al. (2015), Loheac et al. (2015), Berti et al. (2021) ...

Navigation : travel in a complex environment with simple models for displacement



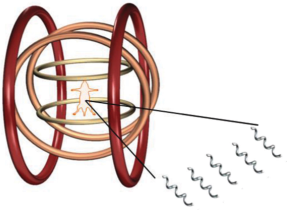
⇒ find optimal path to rapidly reach a target

Gustavsson et al. (2017), Biferale et al. (2019), Liebchen and Lowen (2019), Alaghesan et al. (2020) ...

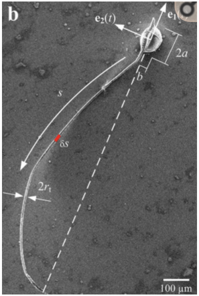
Usually treated separately, assuming large separation of scales

Flagellated locomotion

- Breakthrough $\rightarrow$ in vivo experiments
- Problems : rigid robots
- The hope of the flagellated locomotion ?

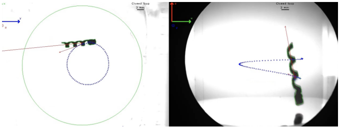


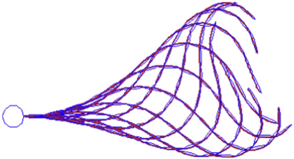
[B. Nelson et al., 2015]



[I. S. M. Khalil et al. 2019]

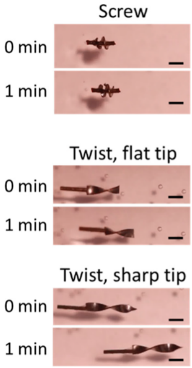
Helicoidal versus flagellated locomotion



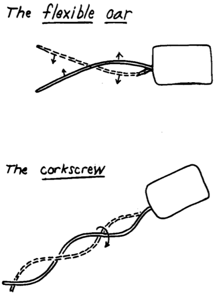


Flagellated locomotion

- The hope of the flagellated locomotion ?
  - Capability to adapt the strategy
- How to control a flagellated robot ?
  - Mathematical modeling
  - Control - optimization tools



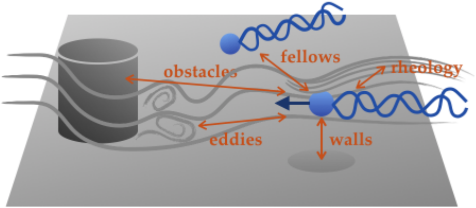
[T. Qiu et al., 2019]



[Purcell 1977]

How to control (optimally) a flagellated micro-swimmer in a bodily fluids ?

- Locomotion and navigation occur at comparable scales.



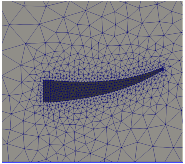
Content

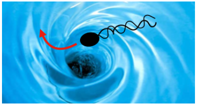
1. Locomotion
   - Numerical methods for solving a swimmer dynamics

   with C. Van Landeghem, L. Berti and C. Prud'homme

2. Locomotion + Navigation
   - Optimization through reinforcement learning

   with R. Chesneaux, Z. El Khiyati and J. Bec





Numerical framework for micro-swimming

- [1] Modelling and finite element simulation of multi-sphere swimmers. Comptes Rendus. Mathématique, 2021
- [2] Reinforcement learning with function approximation for 3-spheres swimmer IFAC 2022
- [3] Mathematical and computation framework for moving and colliding rigid bodies in a Newtonian fluid. Submitted

- Feel++ software

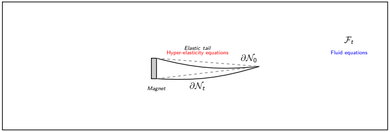
joint work with C. Van Landeghem, L. Berti, V. Chabannes, C. Prud'homme, A. Chouippe, Y. Hoarau

Prescribed deformation - Navier-Stokes

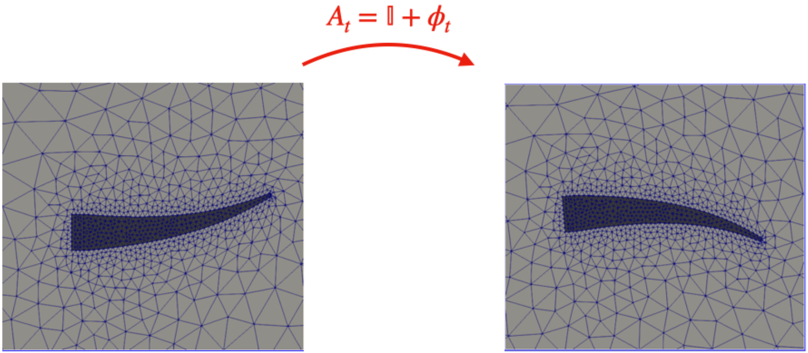
$$
\begin{cases}
\rho(\partial_t u|_{\mathbb{A}_t} + (u - u_{\mathbb{A}_t}) \cdot \nabla u) - \mu \Delta u + \nabla p = 0, & \text{in } \mathcal{F}_t, \\
\nabla \cdot u = 0, & \text{in } \mathcal{F}_t, \\
u = U \circ \mathbb{A}_t, & \text{on } \partial \mathcal{N}_t \ , \\
m\dot{\mathbf{v}} = -F_{fluid} + F^{int} \ , & \\
J\dot{\Omega} = -M_{fluid} + M^{int} \ , &
\end{cases}
$$

$$
U := \underbrace{\mathbf{v} + \Omega \times (x - x^{CM}(t))}_{\text{rigid motion}} + \overbrace{u_d(t)}^{\text{deformation}}
$$

- For u_d prescribed
  $\Rightarrow$ the rigid motion ($\mathbf{v}$ and Ω) are computed.



Arbitrary Lagrangian Eulerian framework

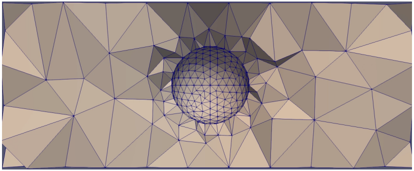


ϕ_t :: swimmer's deformation and an harmonic extension elsewhere

ALE map describing the evolution of the computational domain

Numerical methods

- Classical ALE framework [Maury, 2000]
- Time discretization
- Spatial discretization → conforming Lagrange finite elements
- Moving domain → Arbitrary-Lagrangian-Eulerian technics
- Mesh (using MMG)
  - Mesh quality indices
  - Re-meshing metric → distance to the swimmer
  - Interpolating
- Feel++



Algebraic strategy

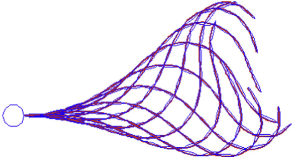
$$\underbrace{\mathcal{P}^T A \mathcal{P}}_{\mathcal{A}} \begin{bmatrix} \mathbf{u}_I \\ \mathbf{u}_\Gamma \\ \mathbf{U} \\ \omega \\ p \end{bmatrix} = \mathcal{P}^T \begin{bmatrix} G_I \\ G_\Gamma \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

- Requires efficient implementation of $\mathcal{P}^T A \mathcal{P}$ in parallel
- Use a block preconditioner of type PCD or PMM

Micro-swimmers deformation gait

The deformation can be modeled by defining the deformation velocity u_d on the boundary of the swimmer.

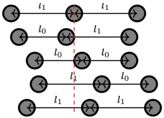
Spermatozoon : u_d for a sinusoidal pl



$$u_d(t,x) = \begin{bmatrix} A_1 \cos(4\pi(t-x)) \\ A_2 \cos(2\pi(t-x)) \\ 0 \end{bmatrix}$$

where A_1, A_2 define the amplitudes.

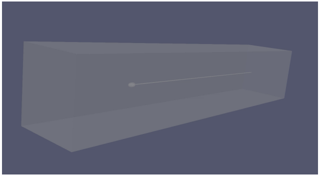
Multi-sphere swimmers :



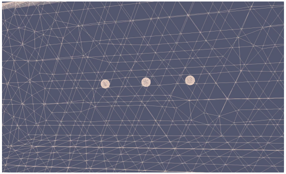
u_d : : relative speed between the spheres.

Numerical simulations

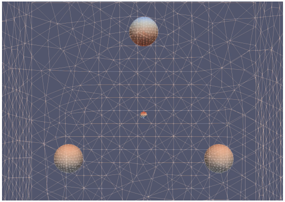
Spermatozoon



Three-sphere swimmer

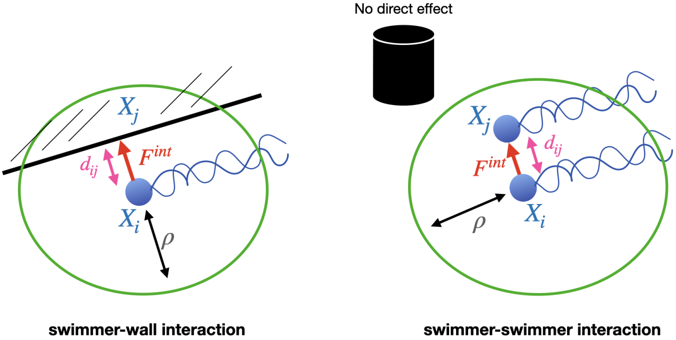


Three-sphere swimmer planar



Feel++ Youtube channel [https ://www.youtube.com/@FeelppOrgChannel]
Feel++ [https ://github.com/feelpp/feelpp]

Lubrification forces and torques

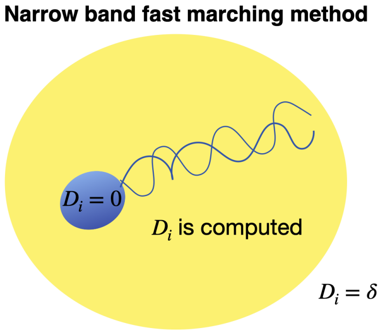


- Short-range repulsive force [R. Glowinski et al.]
- Activated when the distance between two bodies is less than ρ
- Prevent the bodies overlapping and direct contacts

$$F^{int} = \begin{cases} 0\,, & \text{for} \quad d_{ij} > \rho, \\ \frac{1}{\varepsilon}\overrightarrow{X_iX_j}(\rho - d_{ij})^2\,, & \text{for} \quad 0 \le d_{ij} \le \rho. \end{cases} \qquad T^{int} = -\overrightarrow{X^{CM}X_i} \times F^{int},$$

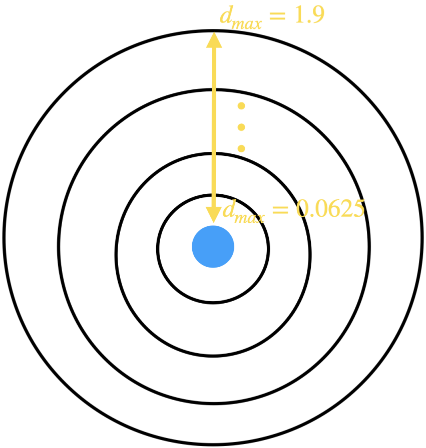
Computation of the distance function

- Distance $D(x) : \mathbb{R}^3 \longrightarrow \mathbb{R}$ compute on the grid
- $\Rightarrow$ Costly ! !
- Narrow band fast marching method
- D is computed only near the neighborhood around front



Fast marching method performance

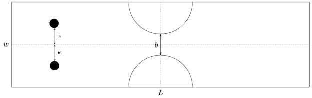
Threshold d_{max}	Execution time in 2D	Speedup
1.9	1.959 s	/
1.0	0.722 s	2.71
0.5	0.411 s	4.76
0.25	0.316 s	6.20
0.125	0.276 s	7.09
0.0625	0.261 s	7.50
Threshold d_{max}	**Execution time in 3D**	**Speedup**
1.9	22.851 s	/
1.0	5.869 s	3.89
0.5	2.368 s	9.65
0.25	1.815 s	12.59
0.125	1.759 s	12.99



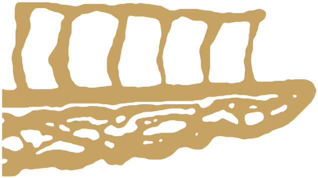
- $\Rightarrow$ The band should be narrow !

Collision simulations

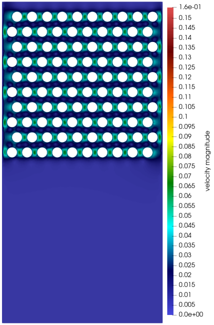
Particles in a 2D symmetric stenotic artery



Zebra fish artery network



Multi-sphere falling down



Sum up

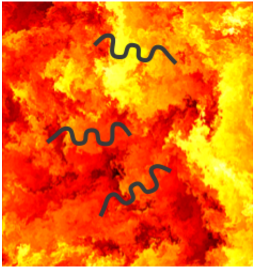
- Swimmer dynamics in a complex environment is very costly
- Optimize in this framework is a challenging task
- Less accuracy dynamics could help
- The importance of selecting the right optimization tools

Could machine learning help ?

2. Locomotion + Navigation through Reinforcement Learning

Steering undulatory micro-swimmers in a moving fluid through reinforcement learning

Joint work with Raphael Chesneaux, Zakarya El Khiyati and Jérémie Bec

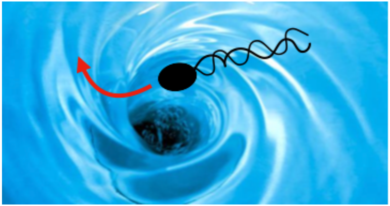


Main idea

How to swim throught eddies ?

Process

- Learn into a "simple fluid flow"
  - Stationary flow with same shape vortices
- Select the best swimmer's (strategies)
- Swim into turbulence with these strokes

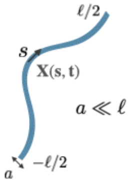


Simpler flagellated swimmer's dynamics

- Cosserat equation (semi-flexible, inextensible, slender body)

$$\sigma\partial_t^2\mathbf{X} = \underbrace{\zeta\mathbb{D}^{-1}[\partial_t\mathbf{X} - \mathbf{u}(\mathbf{X},\mathbf{t})]}_{\text{viscous drag}} + \underbrace{\partial_s(T\partial_s\mathbf{X})}_{\text{tension}} - \underbrace{EI\partial_s^4\mathbf{X}}_{\text{elasticity}} + \underbrace{f(s,t)}_{\text{locomotion force}}$$

$$\mathbb{D} = \mathbb{I} + \partial_s\mathbf{X}\partial_s\mathbf{X}^\mathbf{T}, \qquad \zeta = \frac{\mathbf{8}\pi\mu}{\mathbf{1} + \mathbf{2}\log(\mathbf{l}/\mathbf{a})}$$

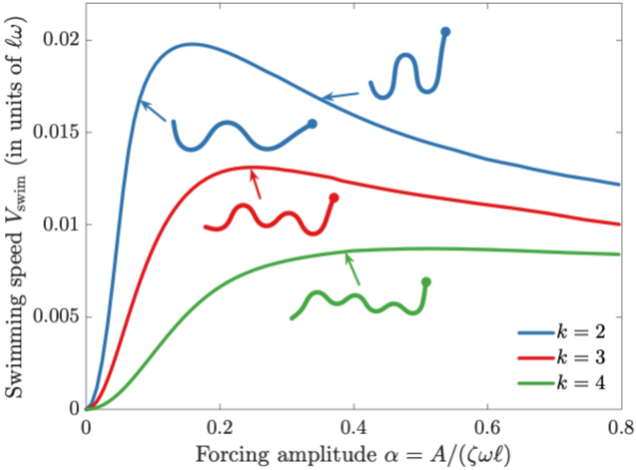


- Inextensibility constraint $\longleftrightarrow \partial_s\|\mathbf{X}\| = \mathbf{1}$

- Solving using semi-implicit second-order centred finite difference scheme (see [Tornberg and Shelley (2004)])

Active locomotion force without an external flow

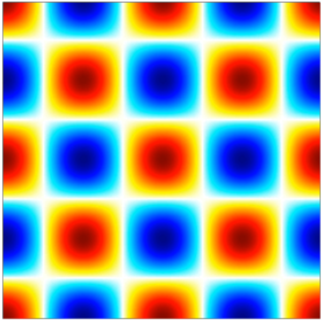
- Planar undulation : $f(s,t) = A\cos(\nu s - \omega t)\mathbf{p}_\perp$
- No net momentum : $\nu = \frac{2\pi k}{l}, \quad k \in \mathbb{Z}$



A simple flow : 2D cellular flow

$$\mathbf{u} = \nabla^{\perp}\mathbf{F} \quad \text{where} \quad \mathbf{F}(\mathbf{x},\mathbf{t}) = \frac{\mathbf{LU}}{\pi}\cos(\pi\mathbf{x_1}/\mathbf{L})\cos(\pi\mathbf{x_2}/\mathbf{L})$$

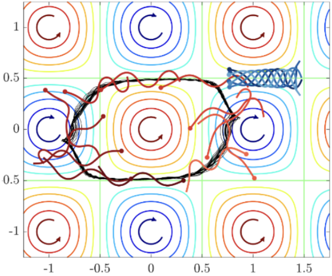
$Re_\alpha \lesssim 1$



Stable, stationary (cellular/BC/Taylor-Green)

$\boldsymbol{u} \propto \nabla^{\perp} F$

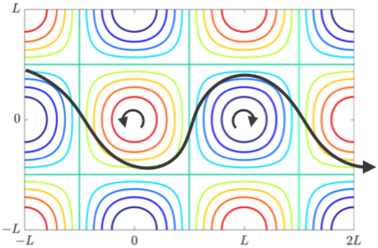
Without any strategy - Trap swimmers



- Red one cycles across several cells
- Blue one is stuck between two cells swimming against the flow

Settings

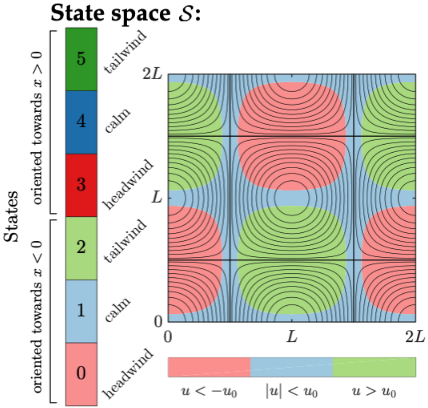
- Locomotion force : controlling the intensity and the wave direction
- Navigation problem : move toward $x > 0$
- Optimization problem : maximize the speed during a large time

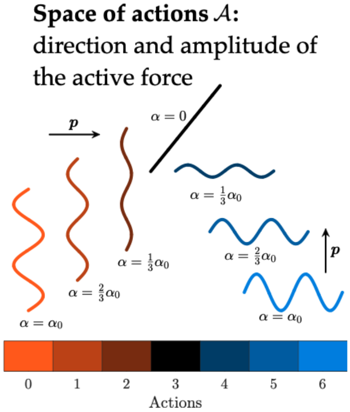


Surfing

Difficulties : Although the fluid flow is stationary, the swimmer's dynamics is chaotic.

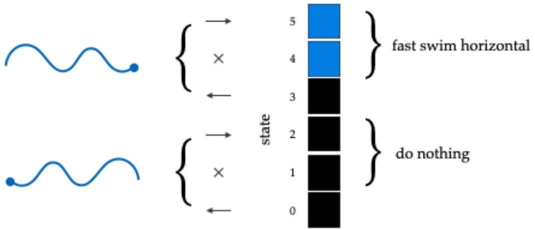
Discretize the optimization problem

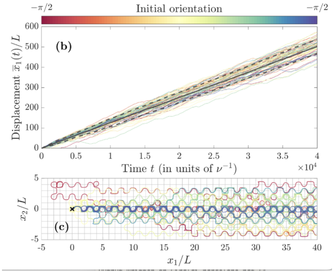




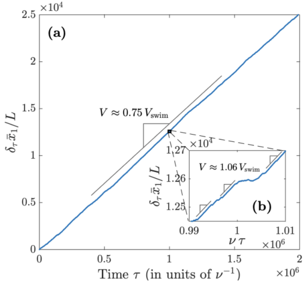
Policy : choose action given current state $(a, s) \in \mathcal{A} \times \mathcal{S} \to \pi(a, s)$

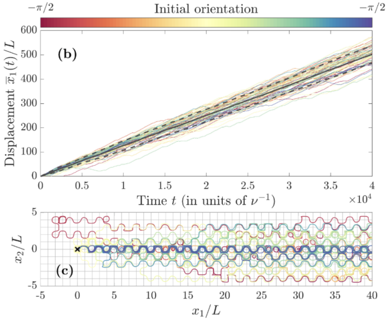
Naive strategy





Performance of the naive strategy





- Trapping events are no more stable !
- $\Longrightarrow$ Naive strategy leads to a positive displacement

Q-learning

- The agent in a state $s(t) \in \mathcal{S}$ decides the action $a(t) \in \mathcal{A}$ $\Rightarrow$ reward $r(t)$ and enters in $s(t + \Delta t)$
- The algorithme maximises an expected future reward $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$

$$a(t) = \begin{cases} a_* = \arg\max_{\mathcal{A}} Q(s(t), a) & \text{with prob. } 1 - \varepsilon \\ a \neq a_* \quad \text{with prob. } \varepsilon \end{cases}$$

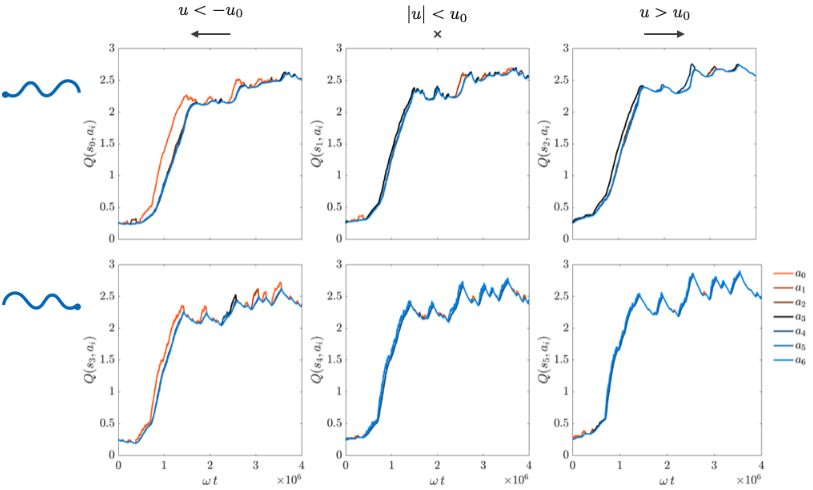
- The Q-function is updated at each step according to

$$Q^{\text{new}}(s(t), a(t)) = (1 - \lambda) Q(s(t), a(t)) + \lambda \Big(r(t) + \gamma \max_{\mathcal{A}} Q(s(t + \Delta t), a) \Big)$$

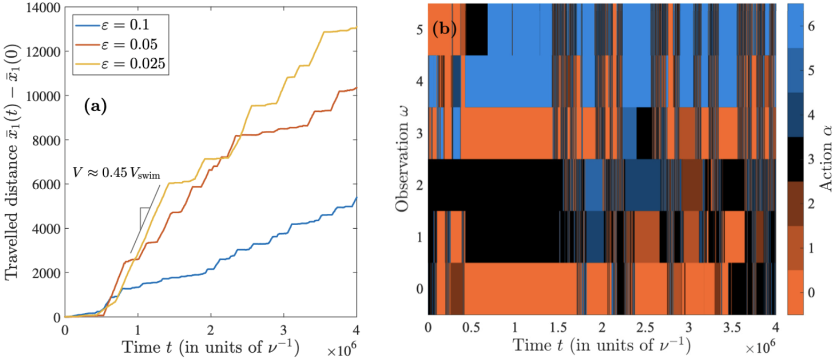
- λ : : learning rate $\leftrightarrow$ typical time to travel 1 cell
- γ : : discount factor $\leftrightarrow$ typical time to travel 10 cells
- Convergence ensured for Markovian decision process with conditions

Q-learning : no convergence

- coarse discretization into state and action $\rightarrow$ Partially observable markov decision process



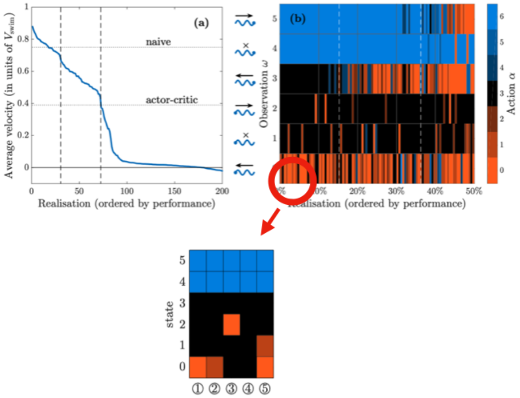
Q-learning : no convergence



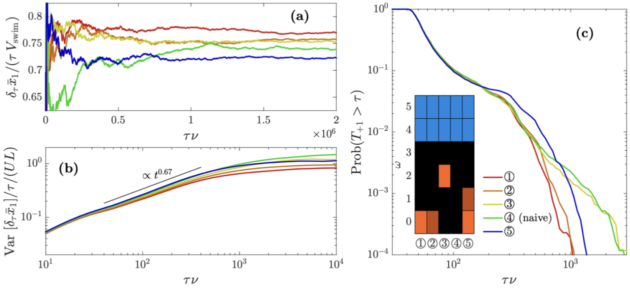
- Succession of surfing and trapping events

Select a set of admissible policies

- Various methods of reinforcement learning (SARSA, Actor-critic) fails
- Q-learning allows to select "admissible" policies



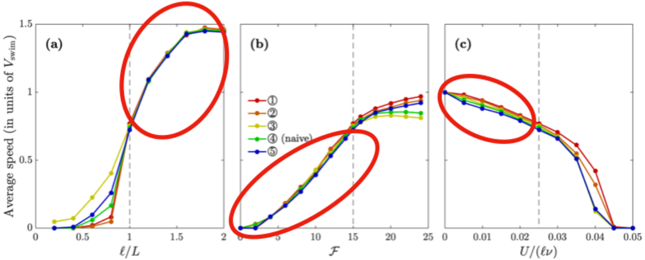
Performance a set of admissible policies



- Mean and Variance of the displacement relatively similar
- Proba to be trap into one cell is much smaller for 1 and 2

Robustness with respect to physical parameters

- Lenght ratio - Flexibility and Amplitude of the fluid flow



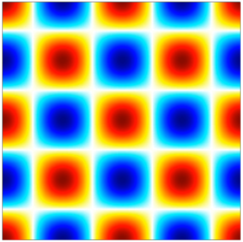
⇒ The performance ranking is not affected in the "red zone"

Into turbulent flow

2D Navier-Stokes $\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla \mathbf{p} + \mu \nabla^2 \mathbf{u} - \alpha \mathbf{u} + \nabla^\perp \mathbf{F}\,, \quad \nabla \mathbf{u} = \mathbf{0}$

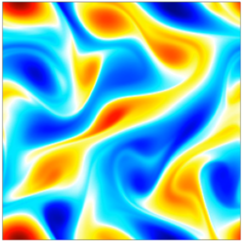
$$Re_\alpha = \frac{U}{\alpha L}$$

$Re_\alpha \lesssim 1$



Stable, stationary (cellular/BC/Taylor-Green)
$\boldsymbol{u} \propto \nabla^\perp F$

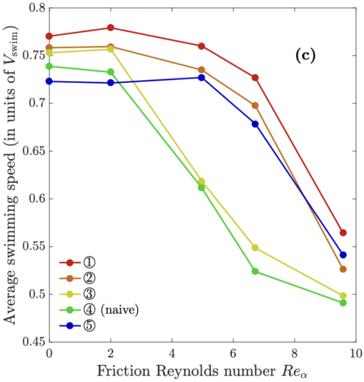
$Re_\alpha \gtrsim 1$



Unstable, turbulent

- Transition to turbulent by increasing Re_α
- Numerical simulation made by a speudo-spectral solver

Robustness into turbulent flow



- Learned strategies globally work but with increasing effects of trapping

Summary and perspectives

- Convergence of $Q-$learning algorithms is not granted, certainly due to both non-Markovianity and chaoticity
- $Q-$learning algorithms allow to select some "good" strategies
- Towards more realistic models
- $\rightarrow$ Complex flow
- $\rightarrow$ Complex environment
- ANR NEMO

- Thanks for your attention.