A hybrid high and low dimensional approach for ocean wave energy converters

Michel Bergmann

INRIA Bordeaux Sud-Ouest - MEMPHIS team
Institut de Mathématiques Appliquées de Bordeaux

michel.bergmann@inria.fr
http://www.math.u-bordeaux1.fr/~mbergman/
https://team.inria.fr/memphis/

Context and motivations - Inria MEMPHIS Team

MEMPHIS - "Modeling Enablers for Multi-PHysics and InteractionS"

Angelo Iollo (Pr UB, Team Leader): *ROM + compressible*

MB (CR Inria): *ROM + incompressible + elasticity*

Afaf Bouharguane (MC UB): *Optimal transportation*

Charles-Henri Bruneau (Pr UB): *DNS turbulence*

+

Several PhDs - Post-docs - Engineers

"We aim at a step change in numerical modeling for science and engineering. We do that by developing two fundamental enablers: reduced-order models and monolithic numerical models on hierarchical Cartesian grids. Thanks to these enablers it will be possible to transfer complexity handling from engineers to computers, providing fast, on-line numerical models for simulation."

Institut de Mathématiques de Bordeaux

informatics mathematics

Context and motivations - Inria MEMPHIS Team

Energy systems: windturbines, VALEOL (2 cifre PhDs)

- **Fluid-Structure (elastic) interactions**

Source: youtube

Context and motivations - Inria MEMPHIS Team

Wave Energy Converters: ISWEC, W4E

- **Fluid-Fluid-Structure (rigid) interactions**

Source: W4E

Context and motivations - Inria MEMPHIS Team

Aeroelastic problems: H2020 AEROGUST + VALEOL

- **Fluid-Structure (elastic) interactions**

Source: Courtesy NASA, Stillman Fires Collection, Prelinger Archives www.cs.wright.edu

Institut de Mathématiques de Bordeaux

Context and motivations - Inria MEMPHIS Team

Biomimetic and bioinspirations: MRGM

- **Fluid-Structure (deformable) interactions: optimal mass transportation**

Source: 3 days larvae: Video from MRGM (P. Babin and A.M Knoll-Gellida)

Context and motivations - Inria MEMPHIS Team

Biomimetic and bioinspirations: CorWave LVAD (cifre PhD)

- **Fluid-Structure (deformable) interactions** fish-like swimming

Source: http://www.corwave.com/presentation/therapy-lvad

Context and motivations - Inria MEMPHIS Team

Flows with particles: CRPP-LOF, LOMA (PhD + projet region)

- **Example** ciment, interactions/collisions

Numerical simulation

Context and motivations

$\hookrightarrow$ **Try to solve all these phenomena in a single monolithic framework**

Goal: perform massive parallel numerical simulations for complex flows with interfaces

$\Rightarrow$ requires HPC (*High Performance Computing*)

Basically, we need:

- $\hookrightarrow$ precise schemes,
- $\hookrightarrow$ simple and robust schemes,
- $\hookrightarrow$ **fast and easy set up of the simulations,**
- $\hookrightarrow$ massive parallel computations.

$\Rightarrow$ **Engineer time $\rightarrow$ CPU time**

"The simplest for the user"

Institut de Mathématiques de Bordeaux

Outline

- **Modeling and numerical methods**

  $\hookrightarrow$ Cartesian/Hierarchical (octree) mesh, penalization, level-set

  $\hookrightarrow$ Research code for applications, // 10^4 CPUs

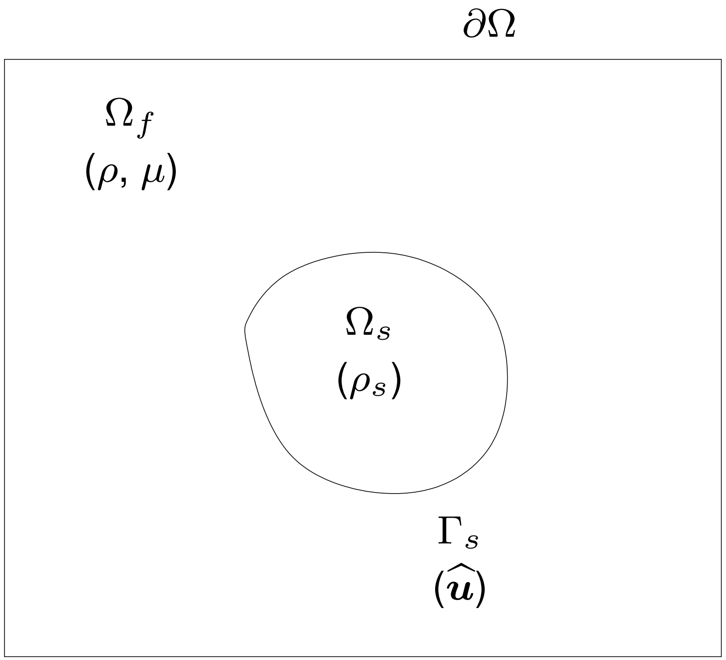
- **Wave Energy Converters**

  $\hookrightarrow$ Rigid body (Wave For Energy (W4E)/Optimad)

  $\hookrightarrow$ Elastic body (Pelamis like WEC)

Modeling and numerical methods | Fluid/Structure

- **Modeling of open flows around deformable bodies**



Navier-Stokes equations in domain Ω_f :

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

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

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

$+$ *initial conditions* $+$ *boundary conditions on external boundary* $\partial\Omega$.

$\hookrightarrow$ **How to manage numerically the unsteady boundary conditions on Γ_s?**

$\hookrightarrow$ **What kind of boundary conditions on external $\partial\Omega$ for open flows?**

Modeling and numerical methods | Fluid/Structure

- **Unsteady mesh adaptation to accurately track interfaces**

Cécile Dobrzynski (IMB)

+ Very efficient: precision
+ Can be quite fast: only fine mesh near bodies
- Not an easy set up: mesh generation
- Can also be very costly: mesh adaptation
- Complicated numerical schemes (FEM)

Institut de Mathématiques de Bordeaux

informatics mathematics

Modeling and numerical methods | Fluid/Structure

- **Embedded interfaces**

+ Simple grids (Cartesian/octrees) $\Rightarrow$ simple numerical schemes (Finite Volumes)
+ Easy parallel computing: simple domain decomposition
- Precision near interfaces (boundary layers)

$\hookrightarrow$ Indeed, the bodies interfaces do not match the cartesian fluid mesh

$\hookrightarrow$ How to capture the interface? Two ways:

$\triangleright$ Eulerian: capture interface with scalar function $\phi(\boldsymbol{x}, t)$ transported with $\widehat{\boldsymbol{u}}$

$$\frac{\partial \phi}{\partial t} + \widehat{\boldsymbol{u}} \cdot \boldsymbol{\nabla} \phi = 0. \tag{1}$$

$\hookrightarrow$ Large deformations $\Rightarrow$ interfaces fluid/fluid

$\triangleright$ Lagrangian: markers on boundary

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \widehat{\boldsymbol{u}}. \tag{2}$$

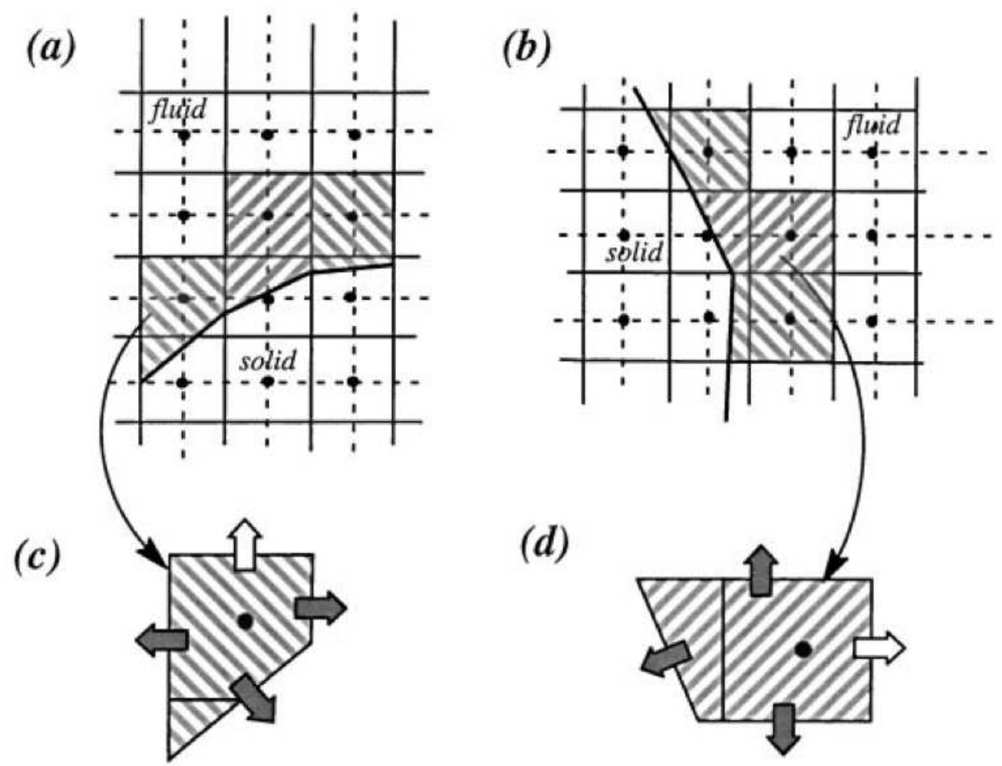
$\hookrightarrow$ Small deformations $\Rightarrow$ interfaces fluid/structure

Institut de Mathématiques de Bordeaux

informatics mathematics

Modeling and numerical methods | Fluid/Structure

- **Embedded interfaces** Cut cell method



Yee, Mittal, Udaykumar, Shyy 1999

+ Efficient: nice conservations
- For us, too difficult to manage in 3D!!

Institut de Mathématiques de Bordeaux

Modeling and numerical methods | Fluid/Structure

▶ **Embedded interfaces with penalization model:** Darcy-Brinkman porous model

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

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0 \quad \text{in} \quad \Omega.$$

↪ Usually $\widehat{\boldsymbol{u}}$ is the body velocity: imposed on nodes inside the body (where $\chi = 1$)

↪ $\chi = H(\phi)$ where H is Heaviside function and ϕ the level set function
($\phi(\boldsymbol{x}) > 0$ if $\boldsymbol{x} \in \Omega$, $\phi(\boldsymbol{x}) = 0$, if $\boldsymbol{x} \in \partial\Omega$, $\phi(\boldsymbol{x}) < 0$ else if).

↪ $\lambda \gg 1$ penalization factor $\rightarrow$ Solution of penalized system tends to solution classical system *w.r.t.* $\varepsilon = 1/\lambda \rightarrow 0$.

+ Very Simple: no need of meshes that fit the body geometries nor cut cell)
+ No need to impose pressure BCs on body boundaries (projection method)
- Precision: only 1^{st} order in space ($\widehat{u} = u$ is not imposed exactly on the boundary)

Institut de Mathématiques de Bordeaux

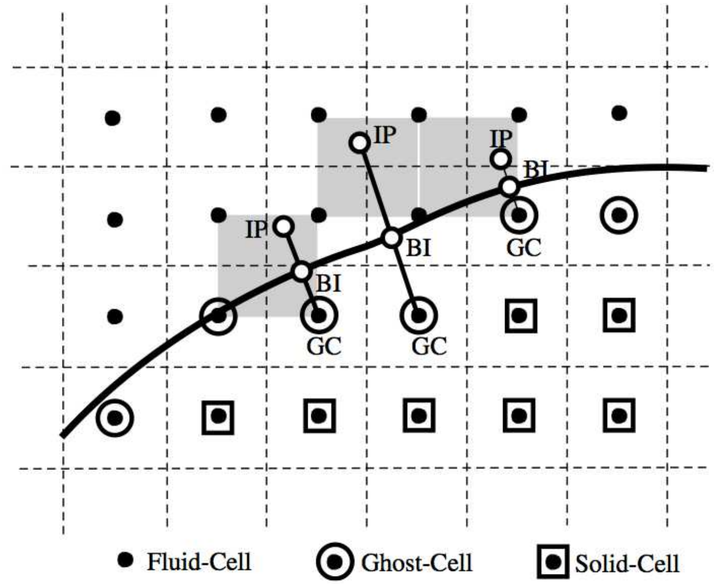
informatics mathematics

Modeling and numerical methods | Fluid/Structure

- **Improvement of penalization model:** discrete ghost fluid like method

$$\hookrightarrow \quad \widehat{\boldsymbol{u}}^n = \boldsymbol{u}_{GC} = 2\,\boldsymbol{u}_{BI} - \boldsymbol{u}_{IP}$$

IP coordinates determined by level set, velocity $\boldsymbol{u}_{IP}$ determined by bilinear interpolation



Ghias, Mittal, Dong 2007

+ Very efficient 2^{nd} order in space
+ Easy to implement!! (small linear systems to solve, see next slides)

Institut de Mathématiques de Bordeaux

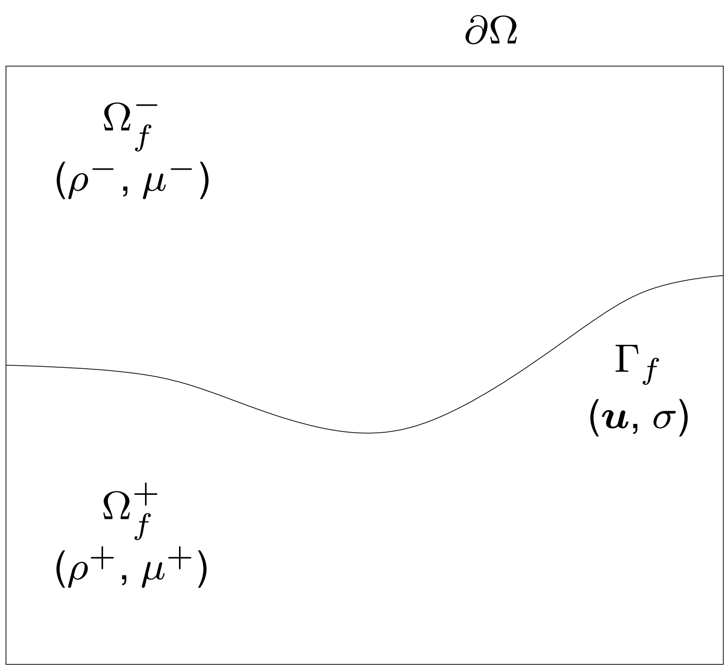
informatics mathematics Inria

Modeling and numerical methods | Fluid/Fluid

- **Flow modeling with interface fluid/fluid**

$$\rho(\psi_f)\left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u}\right) = -\boldsymbol{\nabla} p + \boldsymbol{\nabla}\cdot 2\mu(\psi_f)D(\boldsymbol{u}) + \rho(\psi_f)\boldsymbol{g} \text{ dans } \Omega_f^+ \text{ et } \Omega_f^-,$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0 \text{ in } \Omega_f^+ \text{ et } \Omega_f^-,$$



$$[\boldsymbol{u}(\boldsymbol{x}, t)] = 0 \text{ across } \Gamma_f,$$

$$[-pI + 2\mu D(\boldsymbol{u})]\cdot\boldsymbol{n} = \sigma\kappa\boldsymbol{n} \text{ across } \Gamma_f,$$

$$\rho(\psi_f) = \rho^+ + H(\psi_f)(\rho^- - \rho^+),$$

$$\mu(\psi_f) = \mu^+ + H(\psi_f)(\mu^- - \mu^+).$$

Transport of the level set function :

$$\frac{\partial \psi_f}{\partial t} + \boldsymbol{u}\cdot\boldsymbol{\nabla}\psi_f = 0 \text{ dans } \Omega.$$

Institut de Mathématiques de Bordeaux

Modeling and numerical methods | Fluid/Fluid

- **CSF: the surface tension is considered as being an extra volume force**

$\hookrightarrow$ The jump $[-pI + 2\mu D(\boldsymbol{u})] \cdot \boldsymbol{n} = \sigma\kappa\boldsymbol{n}$ across Γ_f, becomes

$$\rho(\psi_f)\left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u}\right) = -\boldsymbol{\nabla}p + \boldsymbol{\nabla}\cdot 2\mu(\psi_f)D(\boldsymbol{u}) + \sigma\kappa\delta(\psi_f)\boldsymbol{n} + \rho(\psi_f)\boldsymbol{g} \text{ dans } \Omega_f.$$

$\Rightarrow$ regularization of the Dirac over ε

$$\delta^\epsilon(\psi_f) = \frac{\mathrm{d}H^\epsilon(\psi_f)}{\mathrm{d}\psi_f} = \begin{cases} 0 & \text{si } |\psi_f| > \epsilon, \\ \frac{1}{2\epsilon}\left(1 + \cos(\frac{\pi\psi_f}{\epsilon})\right) & \text{si } |\psi_f| \leq \epsilon. \end{cases}$$

$\hookrightarrow$ ρ and μ are also regularized

Advantages: easy to implement

Drawbacks: unphysical velocities can appear near interface + new stability condition

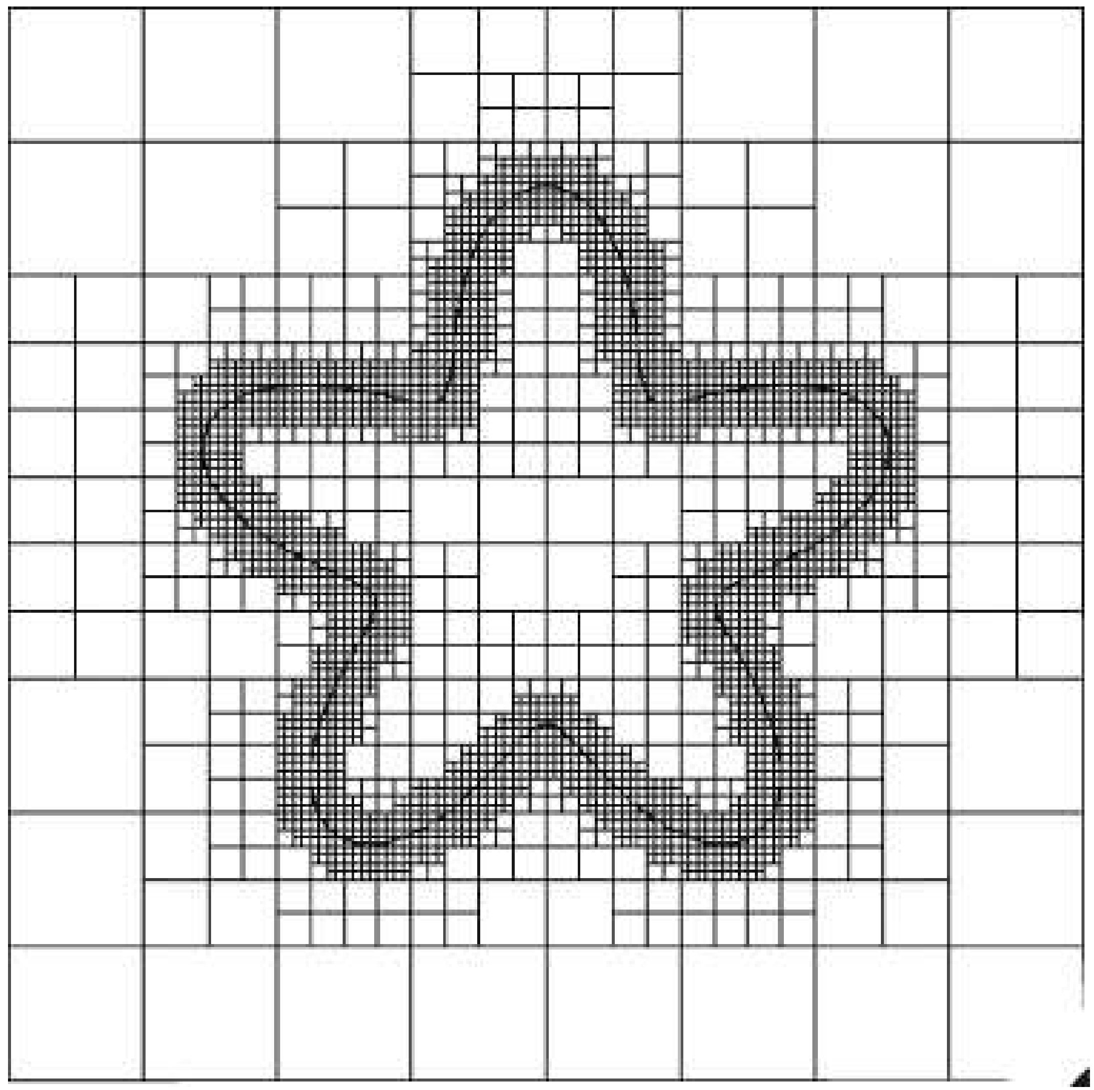
Under consideration: the sharp method developed by M. Cisternino and L. Weynans

Institut de Mathématiques de Bordeaux

informatics mathematics Inria

Modeling and numerical methods | Discretization

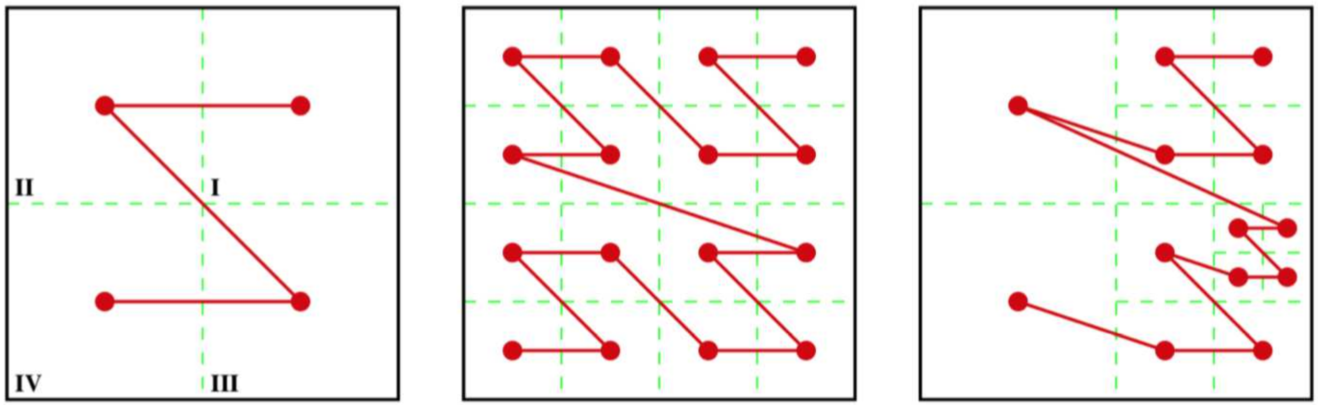
- **Octree grids**

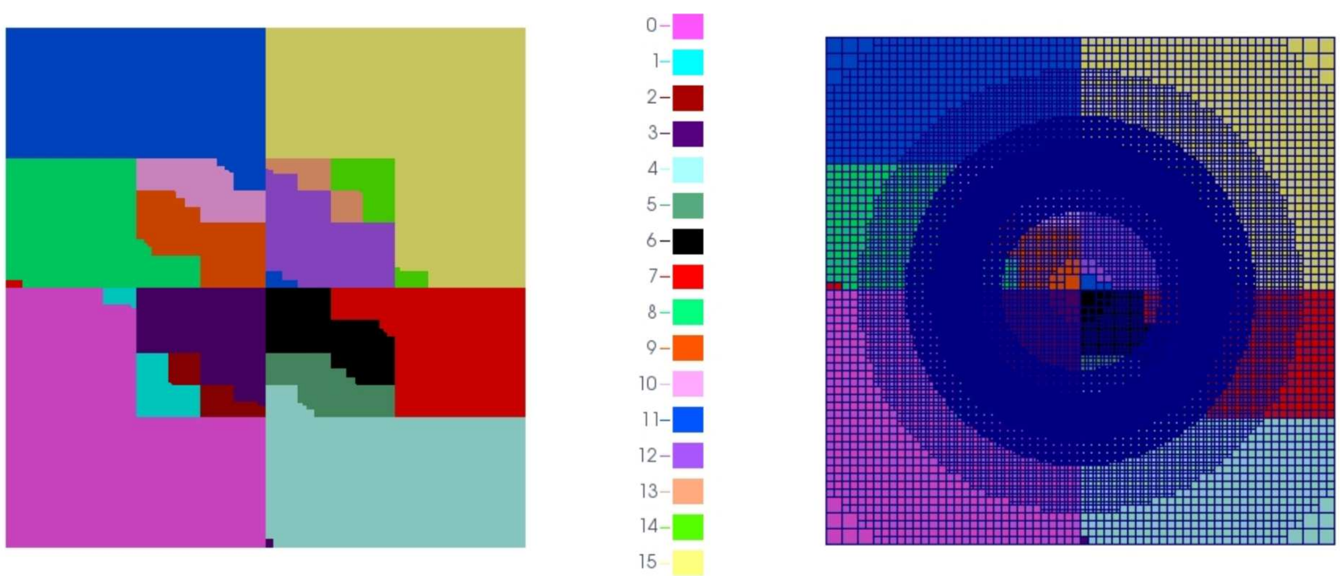


Modeling and numerical methods | Discretization

- **Octree grids**

With Z-ordering (alternative: Hilbert ordering)





Modeling and numerical methods | Discretization

Numerical schemes

- **In space:**
  - $\hookrightarrow$ Hierarchical grids (quadtrees/octrees)
  - $\hookrightarrow$ 2nd order finite volumes
  - $\hookrightarrow$ Penalization and IPC coupled in the Chorin temporal scheme
- **In time: Chorin Temam scheme**
  - $\hookrightarrow$ 2nd order Semi-Lagrangian scheme, implicit viscous term
  - $\hookrightarrow$ Implicit penalization (large penalty term)

$$
\begin{aligned}
\frac{\boldsymbol{u}_a^{(n+1)} - \boldsymbol{u}_d^{(n)}}{\Delta t} &= -\nabla p_a^{(n+1)} + \frac{1}{Re}\Delta \boldsymbol{u}_a^{(n+1)} + \boldsymbol{F} + \lambda \chi^{(n+1)}(\widehat{\boldsymbol{u}}_a^{(n+1)} - \boldsymbol{u}_a^{(n+1)}) \\
\boldsymbol{\nabla} \cdot \boldsymbol{u}_a^{(n+1)} &= 0
\end{aligned}
$$

Modeling and numerical methods | Discretization

Numerical schemes

↪ **Step 1:** Prediction starting from a pressure guess q

$$\frac{\boldsymbol{u}_a^* - \boldsymbol{u}_d^{(n)}}{\Delta t} = -\nabla p_d^{(n)} + \frac{1}{Re}\Delta \boldsymbol{u}_a^* + \lambda\chi^{(n)}(\overline{\boldsymbol{u}}_a^{(n)} - \boldsymbol{u}_a^*) + \boldsymbol{F}$$

Incremental method $q = p^n \longrightarrow$ "accurate" boundary conditions

$\boldsymbol{u}^{**} = \boldsymbol{u}_a^* + \Delta t\,(\nabla p)_{cc}$ $\quad cc$: cell center (stencil $2\Delta x$)

$\boldsymbol{U}^{**} = \gamma(\boldsymbol{u}^{**}),$ $\quad \gamma$ interpolation function

$\boldsymbol{U}^{*} = \boldsymbol{U}^{**} - \Delta t\,(\nabla p)_{fc}$ $\quad fc$: face center (stencil Δx)

↪ subscript a denotes arrival points, *i.e.* on the grid (cell centered)

↪ subscript d denotes departure points, not on the grids ⇒ obtained by interpolations

Institut de Mathématiques de Bordeaux

informatics mathematics

Modeling and numerical methods | Discretization

Numerical schemes

$\hookrightarrow$ **Step 2:** Correction: projection in divergence free space

▷ Poisson equation

$$\Delta\phi^{(n+1)} = \nabla \cdot \boldsymbol{U}^*$$

$\hookrightarrow$ *Not necessary to impose boundary conditions on interface (continuous pressure)*

▷ Correction

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

$$\widetilde{\boldsymbol{U}}^{(n+1)} = \boldsymbol{U}^* - (\nabla\phi)_{fc}^{(n+1)}$$

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

$\hookrightarrow$ face center gradients are obtained using *Diamants (DDFV-like) method*.

Institut de Mathématiques de Bordeaux

Modeling and numerical methods | Discretization

Numerical schemes

$\hookrightarrow$ **Step 3:** Computation of the body motion

$\triangleright$ Newton's laws

$$\boldsymbol{u}^{(n+1)} = f(\widetilde{\boldsymbol{u}}^{(n+1)}, \widetilde{p}^{(n+1)})$$

$\triangleright$ Transport of the distance function ψ: *2nd order semi-Lagrangian with 3rd order interpolations* $\Rightarrow$ ψ is *a priori* not a distance function anymore

$\triangleright$ Redistanciation to recover $|\nabla\psi| = 1$: *sub cell fix with HJ WENO*

$$\frac{\partial\psi}{\partial\tau} + sign(\widetilde{\psi}^{(n+1)})(|\nabla\psi| - 1) = 0 \text{ with } \psi(\boldsymbol{x}, \tau = 0) = \widetilde{\psi}^{(n+1)}.$$

$\hookrightarrow$ *After convergence we obtain $\psi^{(n+1)}$ and thus $\chi^{(n+1)} = H(\psi^{(n+1)})$.*

$\hookrightarrow$ **Step 4:** IPC, correction for 2nd order penalization

$$\frac{\boldsymbol{u}^{(n+1)} - \widetilde{\boldsymbol{u}}^{(n+1)}}{\Delta t} = \lambda\chi^{(n+1)}(\widehat{\boldsymbol{u}}^{(n+1)} - \boldsymbol{u}^{(n+1)})$$

$\Rightarrow$ **The whole system has to be closed with appropriate external boundary conditions!!**
$\Rightarrow$ **For next examples, simple periodic boundary conditions...**

Institut de Mathématiques de Bordeaux

informatics mathematics

Modeling and numerical methods

Applications: swimmers, wind turbines, ocean waves/boat interactions...

Developments and Experiments using local clusters PLAFRIM (1 and 2) and AVAKAS

- Large scale 3D problems: more than one billions dofs

  $\hookrightarrow$ Required parallel code: Very easy with cartesian mesh!!

  $\Rightarrow$ 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)

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

  F-GMRES, preconditioner ASM with ILU on each subdomain

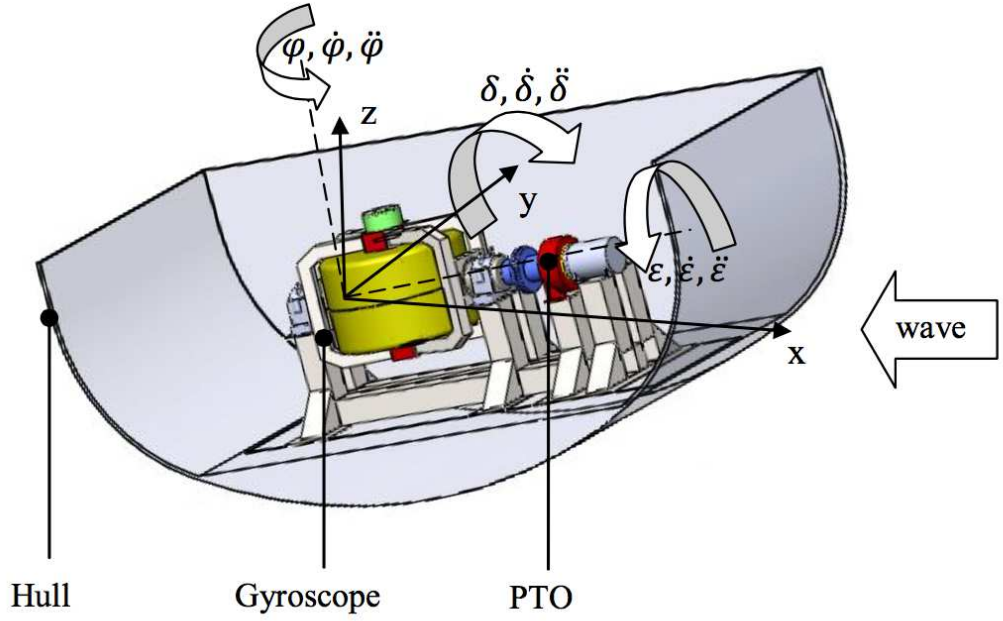
Institut de Mathématiques de Bordeaux

informatics mathematics

Wave Energy Converters | applications

↪ **We have to develop methods adapted to the applications!**

Ocean wave energy: (i) water snakes like pelamis and (ii) iswec



Institut de Mathématiques de Bordeaux

Wave Energy Converters | ocean wave

- **Snake model:** http://www.pelamiswave.com

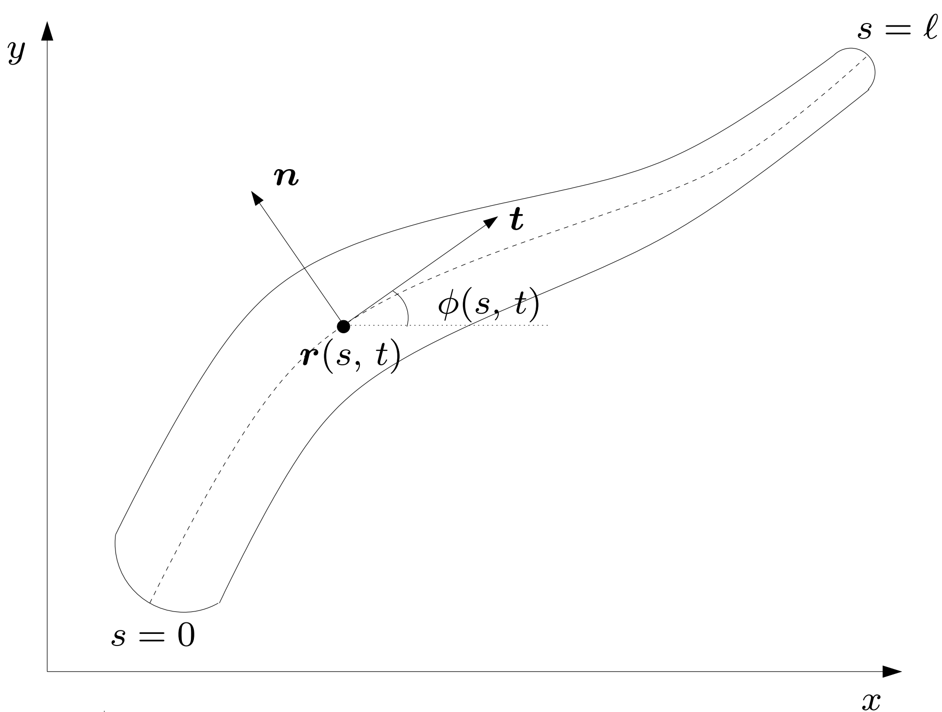
- **Interfaces fluid/fluid/body**
- **Elastic structure + interfaces F/F/S:** water snake

Wave Energy Converters | ocean wave

- **Elastic beam model:** water snake

$\Rightarrow$ linear elasticity

$$M^z = E\,I\left(\frac{\partial \phi}{\partial s} - \kappa\right) + \mu\,I\,\frac{\partial \dot{\phi}}{\partial s},$$



inextensibility

$$\frac{\partial x}{\partial s} = \cos(\phi)$$

$$\frac{\partial y}{\partial s} = \sin(\phi).$$

equations of motion

$$\rho S\,\ddot{x} = W^x + f^x,$$

$$\rho S\,\ddot{y} = W^y + f^y,$$

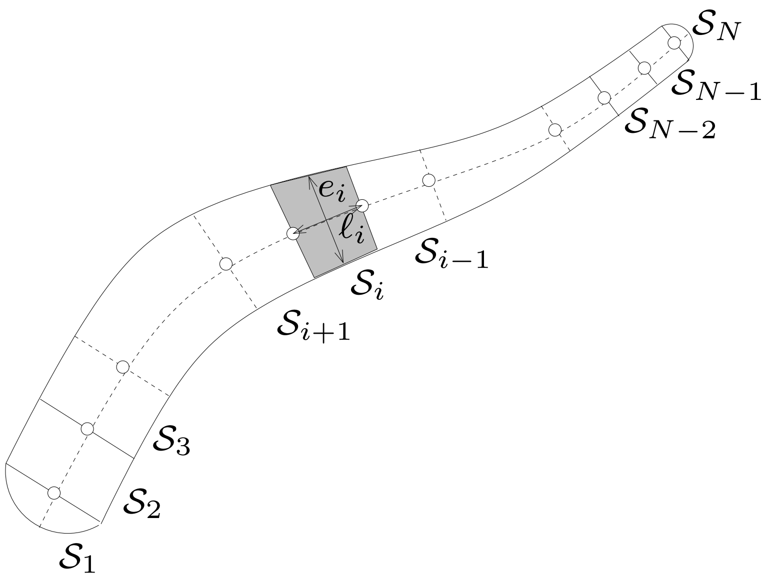
$$\rho I\,\ddot{\phi} = \frac{\partial M^z}{\partial s} + f^y\cos(\phi) + f^x\sin(\phi),$$

Institut de Mathématiques de Bordeaux

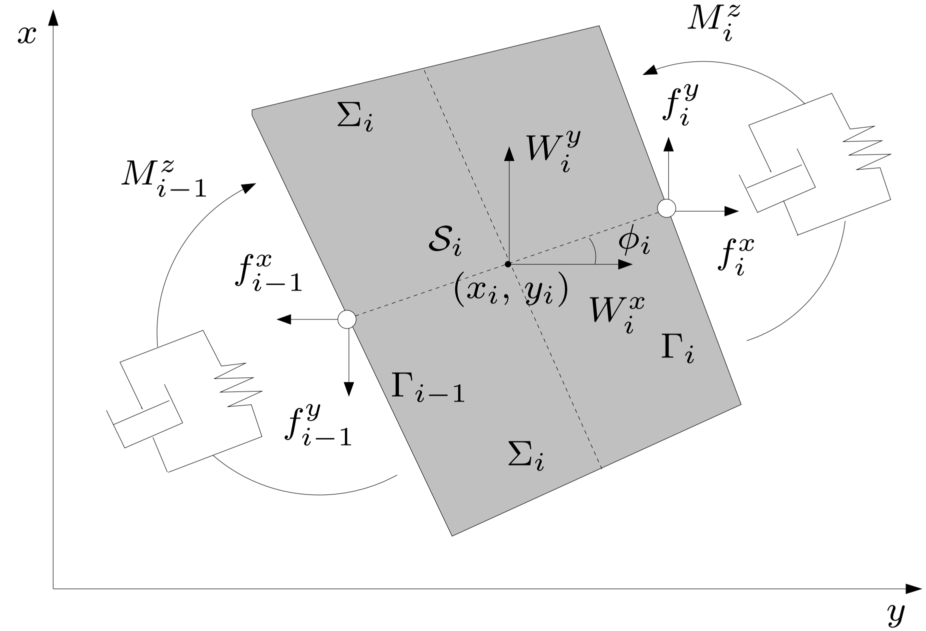
informatics mathematics

Wave Energy Converters | ocean wave

- **Solution:** le system is fully described by $\phi(s,\,t)$ and $(x(s=0,\,t),\,y(s=0,\,t))$



Discretization in sections $\mathcal{S}_i$



Forces and torques acting on section $\mathcal{S}_i$

Fully described knowing $x_1,\,y_1$ et $\phi_1,\ldots,\phi_N$

Dynamical system to solve (RK3)

$\hookrightarrow \ddot{\boldsymbol{z}} = G(\boldsymbol{z},\,\dot{\boldsymbol{z}},\,t)$ avec $\boldsymbol{z} = (x_1,\,y_1,\,\phi_1,\ldots,\phi_N)^T$

(can be expensive for large N!)

Institut de Mathématiques de Bordeaux

Wave Energy Converters | ocean wave

▶ **Numerical method to couple flow and structure solvers:** implicit "strong" coupling

$\hookrightarrow$ Stability: implicit forces have to be used to move the structure

$\hookrightarrow$ Complicated $\Rightarrow$ iterative algorithm

1. At time t^n the interface is captured with ϕ^n. We impose curvature κ^n for structure. Let $k = 0$ and the forces be $\widetilde{\boldsymbol{W}}_k^n = \boldsymbol{W}^n$.
2. Structure code: from ϕ^n, compute $\widetilde{\phi}_k^{n+1}$ with κ^n and $\widetilde{\boldsymbol{W}}_k^n$,
3. Fluid code: from ϕ^n and $\widetilde{\phi}_k^{n+1}$, computed forces $\widetilde{\boldsymbol{W}}_{k+1}^n$,
4. If $\boldsymbol{e} = \|\widetilde{\boldsymbol{W}}_{k+1}^n - \widetilde{\boldsymbol{W}}_k^n\|_\infty < \epsilon$, then $n = n + 1$ and go back to 1. Else if, $k = k + 1$ and go back to 2 to perform a new sub-iteration.

Remark: Usually, we use 3-4 sub-iterations

Institut de Mathématiques de Bordeaux

informatics mathematics Inria

Wave Energy Converters | ocean wave

- **Numerical simulations of the water snake**

3D Example with 10 cylinders, total length 1m
Mesh $1200 \times 300 \times 300 \approx 400\,000\,000$ unknows, 256 CPUs, 10 hours

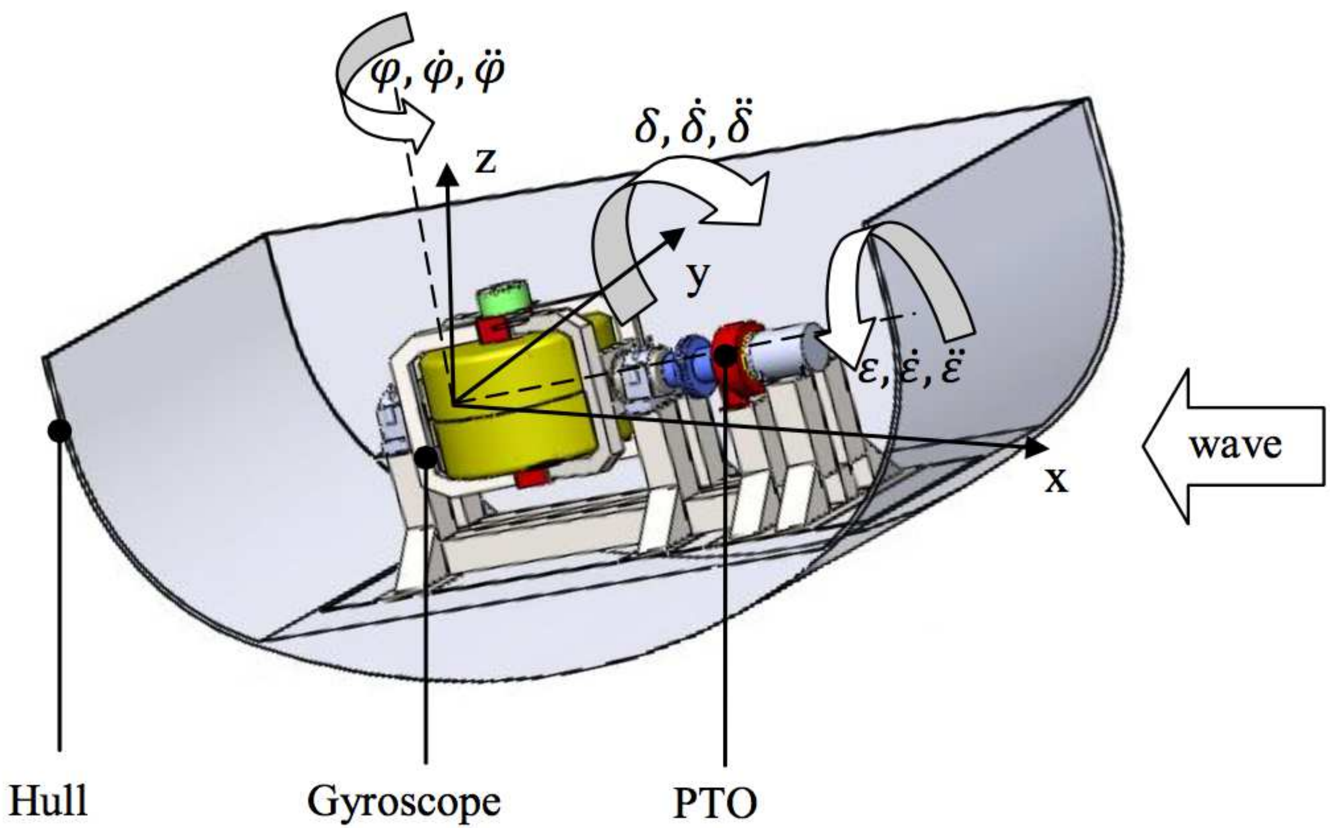
Institut de Mathématiques de Bordeaux

informatics mathematics

Wave Energy Converters | ocean wave

- **Another wave energy system: ISWEC (Inertial Sea Wave Energy Converter)**

  $\hookrightarrow$ **project SeaCure**



Wave Energy Converters | ocean wave

- **Another wave energy system: ISWEC (Inertial Sea Wave Energy Converter)**

  $\hookrightarrow$ **project SeaCure**

Wave Energy Converters | ocean wave

- **Another wave energy system: ISWEC (Inertial Sea Wave Energy Converter)**

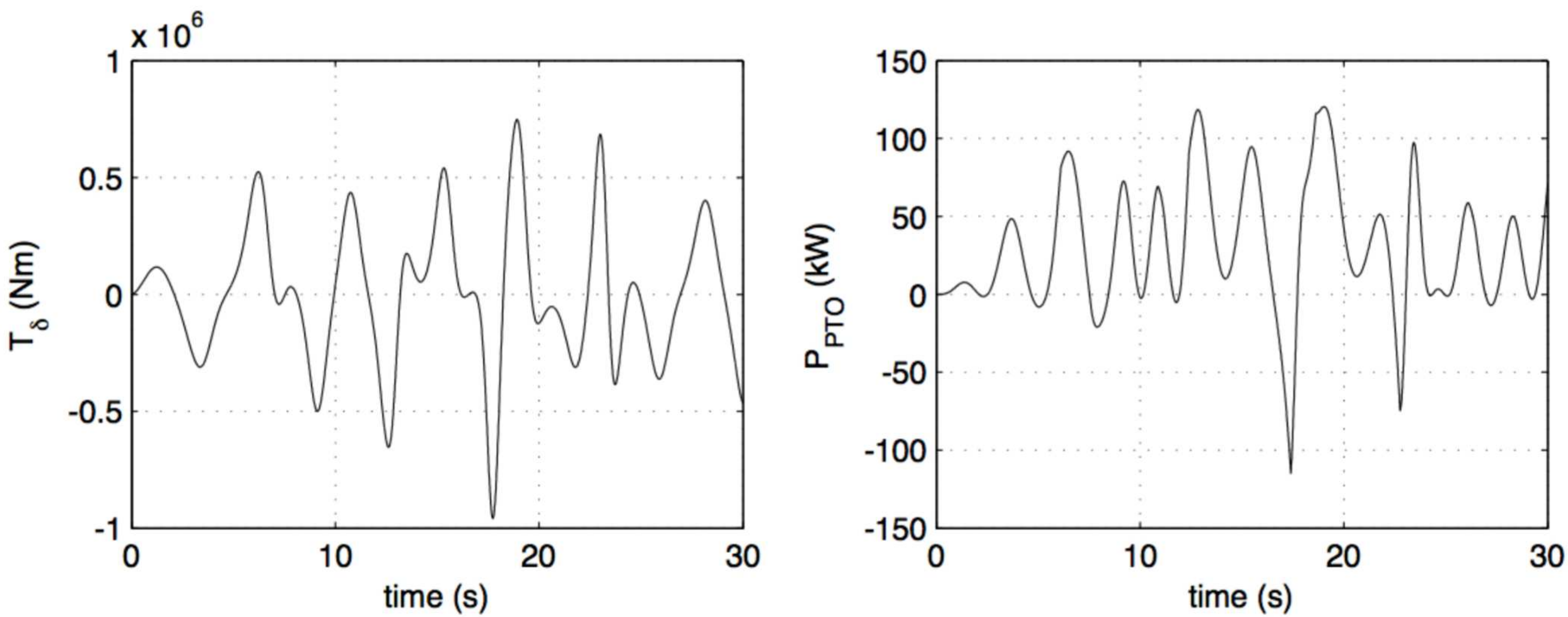
  $\hookrightarrow$ **project SeaCure**

3D Example: total length 15.4m. With mooring, Gyroscope and hydrodynamic effects.
Mesh $1200 \times 300 \times 300 \approx 400\,000\,000$ unknows, 256 CPUs, 10 hours

Wave Energy Converters | ocean wave

- Another wave energy system: ISWEC (Inertial Sea Wave Energy Converter)

  ↪ project SeaCure



Power extracted by the gyroscope

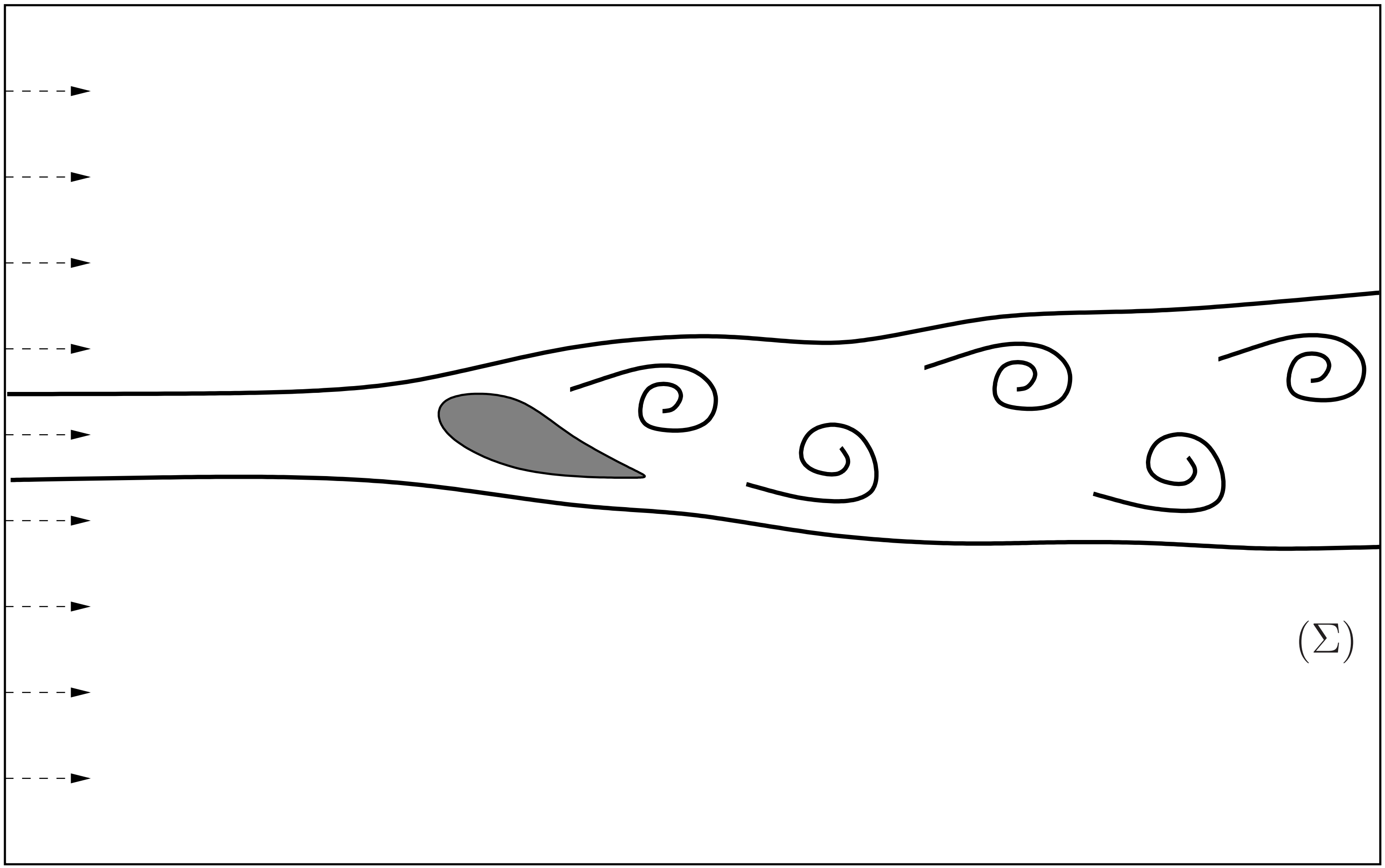
High and low dimensional models

- Boundary conditions are not physical (periodic)
- Not real ocean waves (dam break)
- Large domains are needed to impose more physical BCs
- Each simulations is costly
- Only few simulations can be done
- Coupling with low fidelity model (less accurate but faster!)

Institut de Mathématiques de Bordeaux

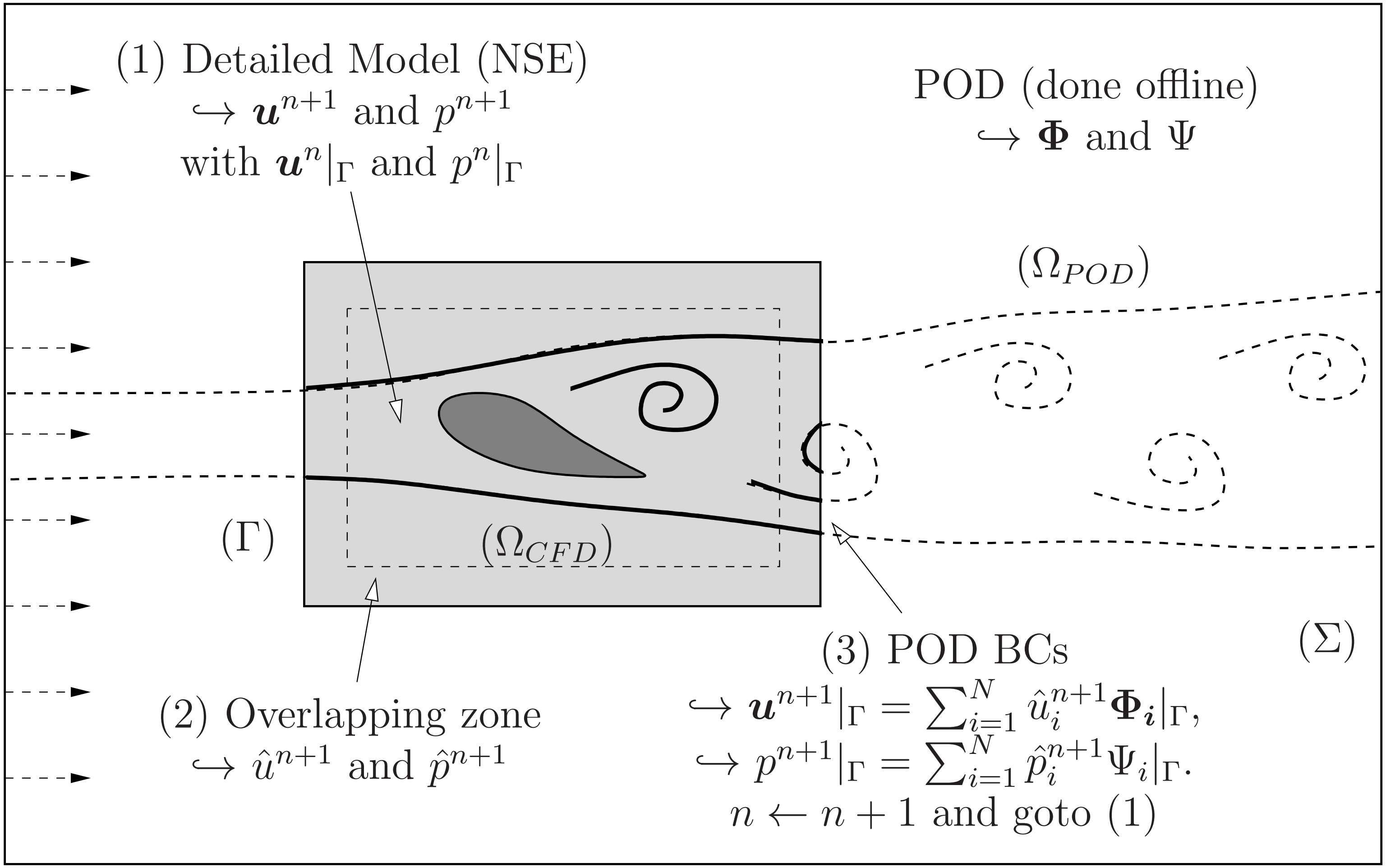
informatics mathematics

High and low dimensional models



Problem with outflow boundary conditions: we have to impose artificial BCs!
$\Rightarrow$ Far away, large domain!

High and low dimensional models



Look for BCs on a Proper Orthogonal Decomposition (POD) subspace computed offline

$\Rightarrow$ **how to compute a robust POD subspace if input parameters change?**

High and low dimensional models

Proper Orthogonal Decomposition (POD), Lumley (1967)

▷ Look for the flow realization $\boldsymbol{\Phi}(\boldsymbol{X})$ that is "the closest" in an average sense to realizations $\boldsymbol{U}(\boldsymbol{X})$.

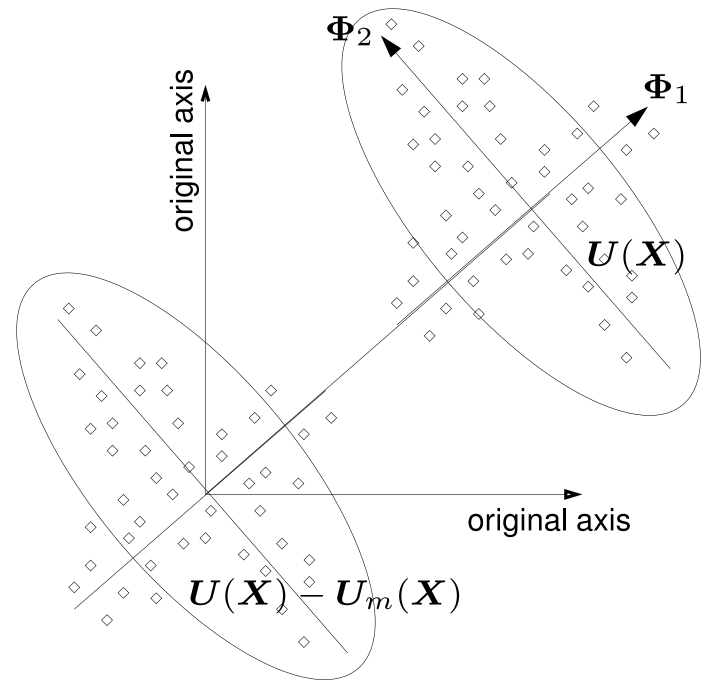
$$(\boldsymbol{X} = (\boldsymbol{x}, t) \in \mathcal{D} = \Omega \times \mathbb{R}^{+})$$

▷ $\boldsymbol{\Phi}(\boldsymbol{X})$ solution of problem:

$$\max_{\boldsymbol{\Phi}} \langle |(\boldsymbol{U}, \boldsymbol{\Phi})|^2 \rangle, \quad \|\boldsymbol{\Phi}\|^2 = 1.$$

▷ Optimal convergence *in L^2 norm* de $\boldsymbol{\Phi}(\boldsymbol{X})$

$\Rightarrow$ Dynamical reduction possible.



Lumley J.L. (1967) : The structure of inhomogeneous turbulence. *Atmospheric Turbulence and Wave Propagation*, ed. A.M. Yaglom & V.I. Tatarski, pp. 166-178.

Institut de Mathématiques de Bordeaux

informatics mathematics

High and low dimensional models

▷ Equivalent with Fredholm equation:

$$\int_{\mathcal{D}} R_{ij}(\boldsymbol{X},\boldsymbol{X'})\Phi_n^{(j)}(\boldsymbol{X'})\,d\boldsymbol{X'} = \lambda_n \Phi_n^{(i)}(\boldsymbol{X}) \qquad n = 1, .., N_s$$

$\hookrightarrow R(\boldsymbol{X},\boldsymbol{X'})$: *Space-time correlation tensor*

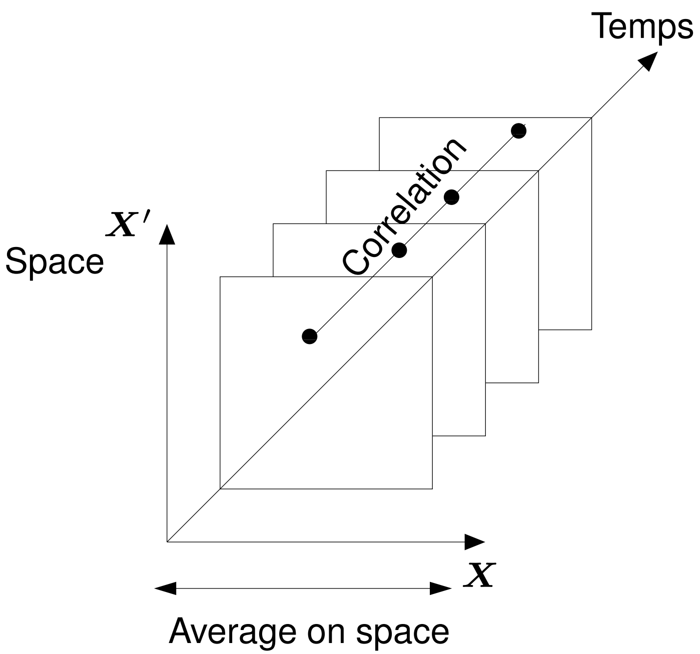
▷ Snapshots method, Sirovich (1987) :

$$\int_T C(t,t')a_n(t')\,dt' = \lambda_n a_n(t)$$

$\hookrightarrow C(t,t')$: *Temporal correlations*

▷ POD basis $\boldsymbol{\Phi}(\boldsymbol{X})$ for one set of input parameters space:

$$\boldsymbol{U}(\boldsymbol{x},t) = \sum_{n=1}^{N_s} a_n(t)\boldsymbol{\Phi}_n(\boldsymbol{x}).$$



Sirovich L. (1987) : Turbulence and the dynamics of coherent structures. Part 1,2,3 *Quarterly of Applied Mathematics*, **XLV** N° 3, pp. 561–571.

High and low dimensional models

▶ **How to perform an efficient sampling of input parameter space?**

- Uniform Sampling in a cartesian way? Problems: not optimal
  $\hookrightarrow$ distance in parameter space $\neq$ "distance" in solution space
- Leave one out (quadtree refinements)? Problems: lot of sampling points
- Stochastic way? Problem: no guarantees

$\hookrightarrow$ Need something else...

▶ **Iterative sampling based on an error criterion**

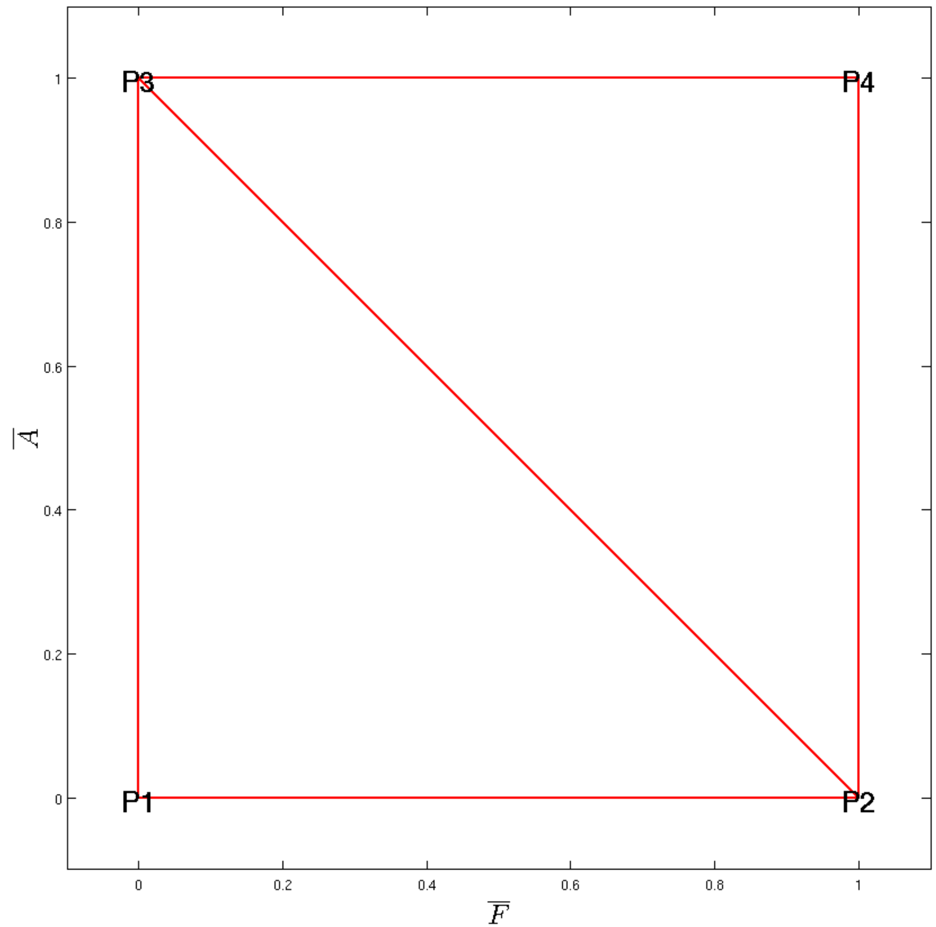
- Iterative method to improve the POD basis
- The error is the mathematical projection error computed using the current POD basis
- "Adaptive mesh refinement" using Delaunay triangulation (dual of voronoi tesselation)

Institut de Mathématiques de Bordeaux

High and low dimensional models

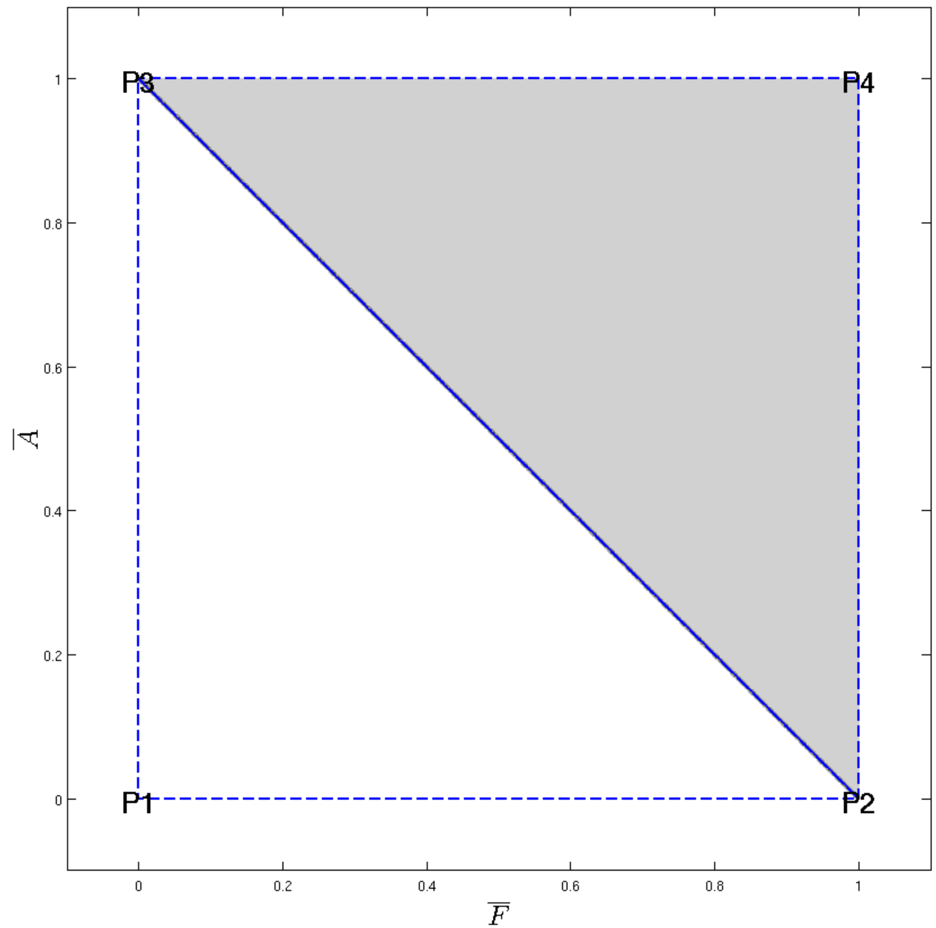
An initial Sampling with non-unique Delaunay Triangulation (4 points)

Pitching airfoil: $F = [30, 70] \mapsto \overline{F} = [0, 1]$ and $A = [1.607, 3.615] \mapsto \overline{A} = [0, 1]$



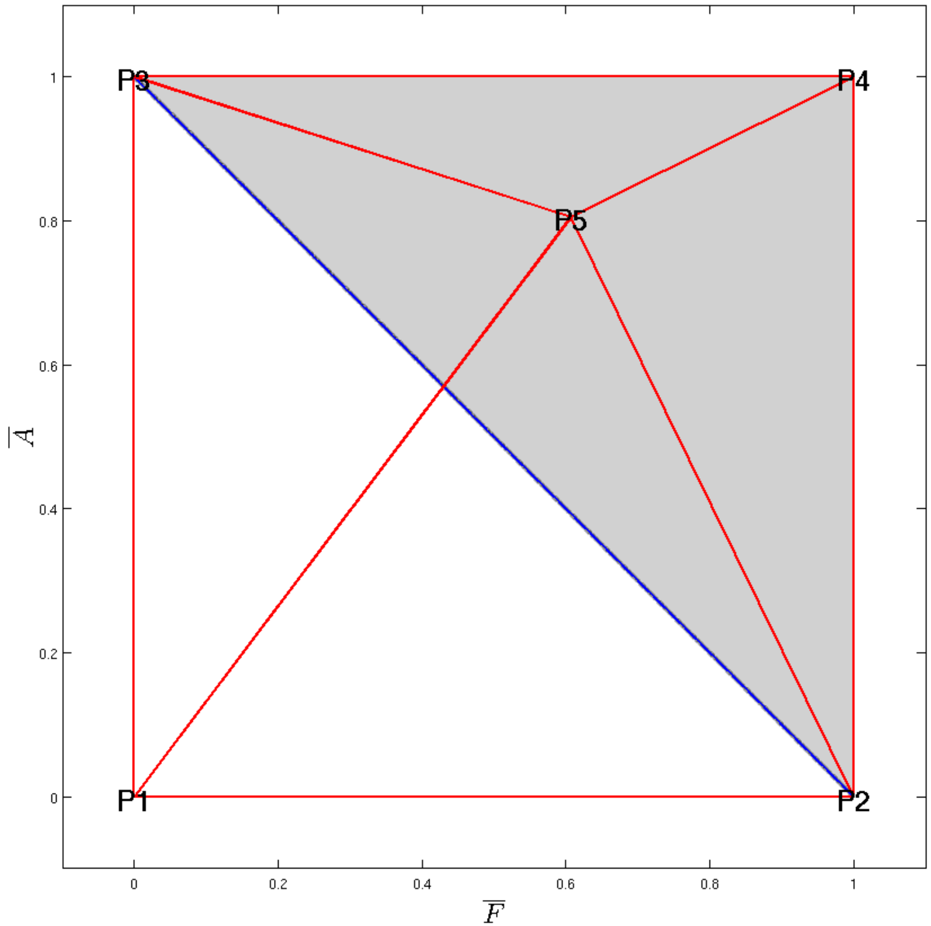
High and low dimensional models

(i) Compute the POD basis (80 snapshots), (ii) compute projection error onto 10 POD modes, (iii) select the triangle with maximal average error



High and low dimensional models

Next Sampling point is the center of mass of that triangle
$A_{new} = 3.21$ and $F_{new} = 54.3$

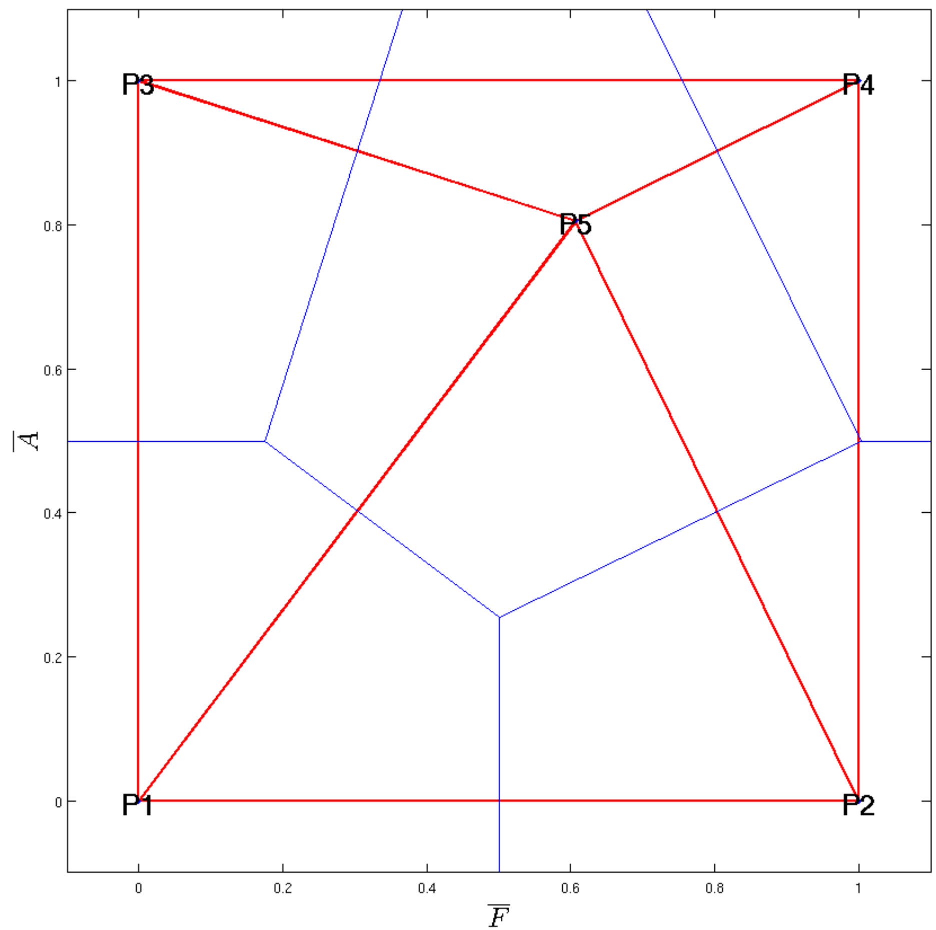


Institut de Mathématiques de Bordeaux

informatics mathematics

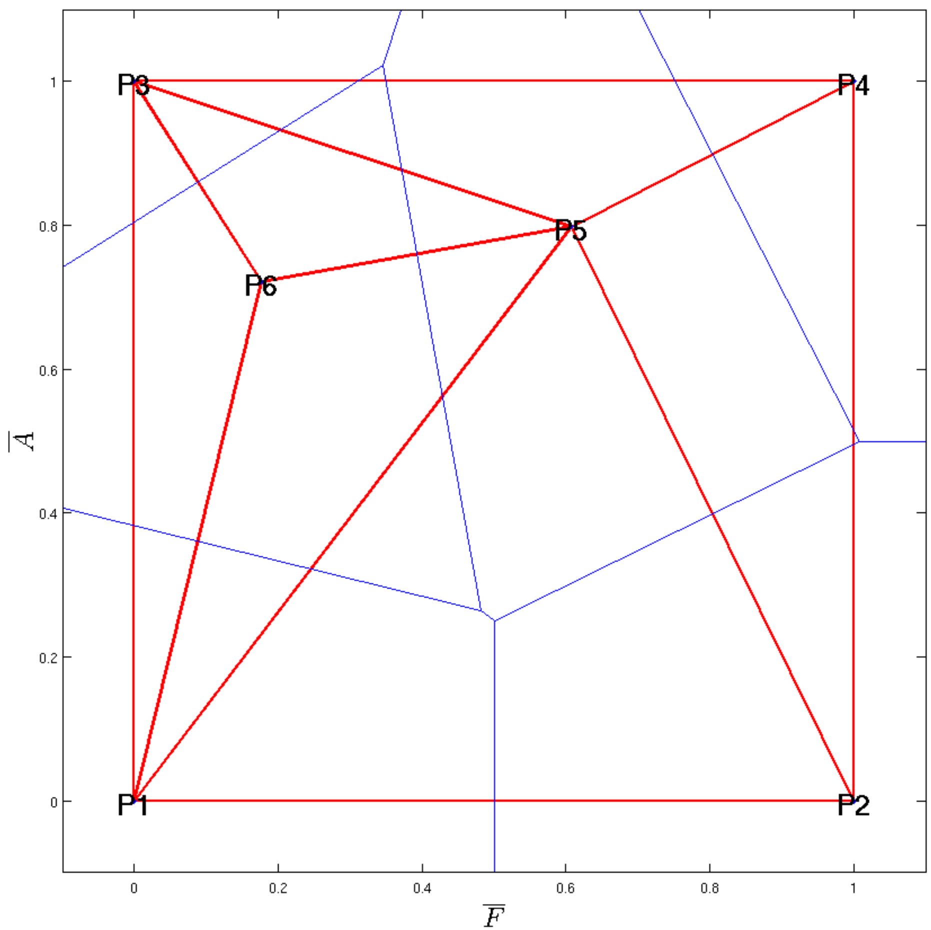
High and low dimensional models

Delaunay sampling with dual Voronoi tesselation
(Voronoi in blue)



High and low dimensional models

Compute new POD basis (5 points, *i.e.* 100 snapshots) and compute the new point $A_{new} = 3.05$ and $F_{new} = 37.1$

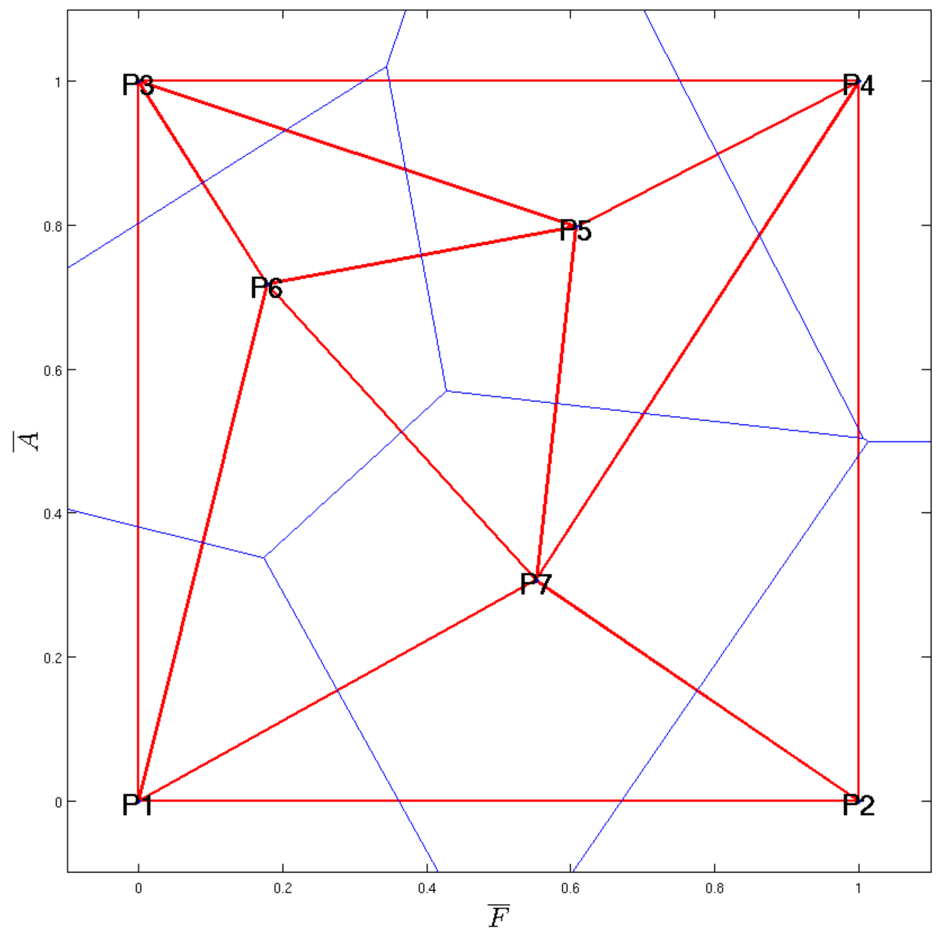


Institut de Mathématiques de Bordeaux

informatics mathematics

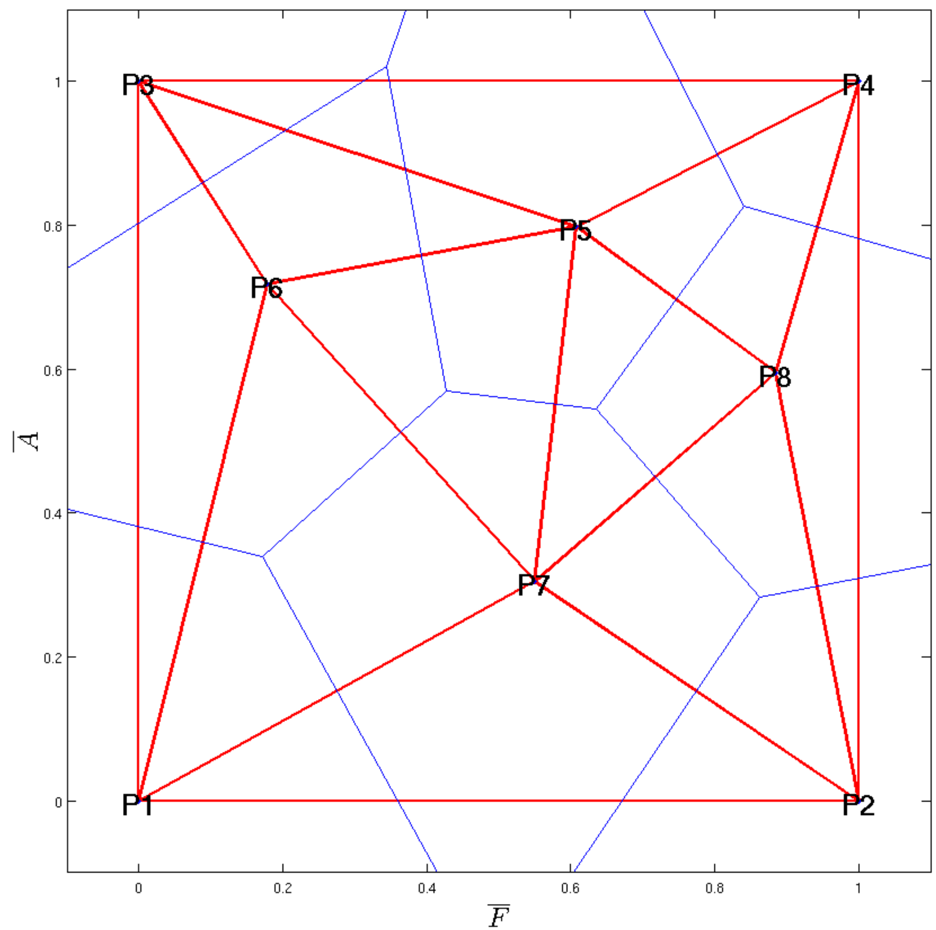
High and low dimensional models

Compute new POD basis (6 points, *i.e.* 120 snapshots) and compute the new point $A_{new} = 2.22$ and $F_{new} = 52.0$



High and low dimensional models

Compute new POD basis (7 points, *i.e.* 140 snapshots) and compute the new point $A_{new} = 2.82$ and $F_{new} = 65.4$



$\hookrightarrow$ **Better than 9 points uniform sampling!**

Institut de Mathématiques de Bordeaux

informatics mathematics

High and low dimensional models

- **How to chose domain Ω_{CFD}?**

$\hookrightarrow$ An error indicator based on leave one out strategy
"sensitivity of the POD basis functions"

– We have M sampling points with $N_s^{(k)}$ snapshots for the k^{th} sampling point

– Alternatively, for each sampling point k:

  – We remove the $N_s^{(k)}$ snapshots and build the POD basis
  – We compute the projection error ($N_s^{(k)}$ onto POD basis)

– We perform an average error over the M sampling points

– We chose a given threshold for the error that is acceptable for POD representation

  – Maximal error is near the obstacle and "far field" ok

– We chose Ω_{CFD} the minimal cartesian box surrounding this error

– We perform high fidelity simulation in that domain.

Institut de Mathématiques de Bordeaux

informatics mathematics

High and low dimensional models

- **Galerkin free reduced order model**

$$\boldsymbol{u}(\boldsymbol{x},\,t) \approx \widetilde{\boldsymbol{u}}(\boldsymbol{x},\,t) = \boldsymbol{u}_g(\boldsymbol{x},\,t) + \sum_{i=1}^{N_R} \hat{u}_i(t)\boldsymbol{\Phi}_i(\boldsymbol{x})$$

$$p(\boldsymbol{x},\,t) \approx \widetilde{p}(\boldsymbol{x},\,t) = p_g(\boldsymbol{x},\,t) \sum_{i=1}^{N_R} \hat{p}_i(t)\Psi_i(\boldsymbol{x})$$

The functions $\boldsymbol{u}_g$ and p_g can snapshots average, zeros, or any desired functions (like gusts)

$\hookrightarrow \{\hat{u}\}_{i=1}^{N_R}$ are obtained minimizing $\|\boldsymbol{u}^* - \widetilde{\boldsymbol{u}}\|_2$ in overlapping domain Ω_o

$\hookrightarrow \{\hat{p}\}_{i=1}^{N_R}$ are obtained minimizing $\|p^* - \widetilde{p}\|_2$ in overlapping domain Ω_o

Institut de Mathématiques de Bordeaux

informatics mathematics Inria

High and low dimensional models

- Example: We are interested in gust effect on 2D airfoil

$\hookrightarrow$ NACA0012 airfoil at $Re = 1000$, $\alpha = 5^\circ$ and chord $c = 1$

$\hookrightarrow$ Interaction with a vortex

$\hookrightarrow$ The flow without gust is steady

$\hookrightarrow$ The computed lift coefficient $C_L = 0.25$ agrees well with the reference results

$\hookrightarrow$ The unsteady vortex shedding at $Re = 1000$ appears for $\alpha \geq 8^\circ$

$\hookrightarrow$ The domain is $[-8\,c,\ 8\,c] \times [-4\,c,\ 4\,c]$. Mesh: $1600 \times 800 \rightarrow 100$ points along the chord

$\hookrightarrow$ The simulation is performed for $0 \leq t \leq 6.5$

Institut de Mathématiques de Bordeaux

informatics mathematics

High and low dimensional models

- Example: We are interested in gust effect on 2D airfoil

High and low dimensional models

- Example: We are interested in gust effect on 2D airfoil

$\hookrightarrow$ The gust is modeled as a vortex (Rotational core + quickly decaying external region)

$$q = U_0 r/R \quad \text{if} \quad r < R$$

$$q = U_0 R^2/r^2 \quad \text{if} \quad r \geq R$$

r: distance from vortex center

R: vortex characteristic radius

U_0: vortex characteristic velocity

q: magnitude of the vortex induced velocity

Institut de Mathématiques de Bordeaux

informatics mathematics Inria

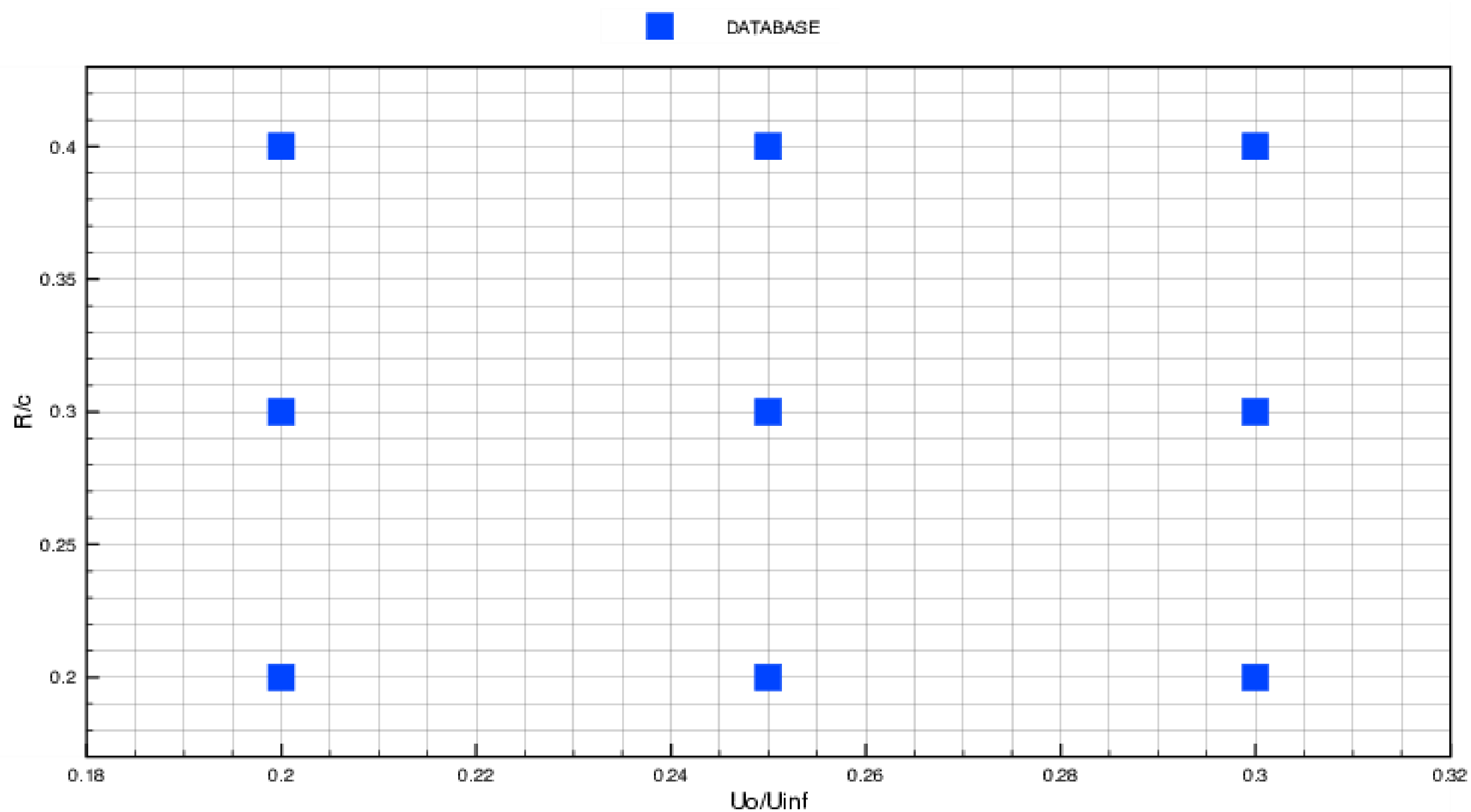
High and low dimensional models

- Definition of the gust functions $\boldsymbol{u}_g$ and p_g: analytical transport $\rightarrow$ reduced CPU costs

High and low dimensional models

▶ Uniform Sampling for robust basis functions over the input parameter subspace

↪ Size and intensity of the vortex

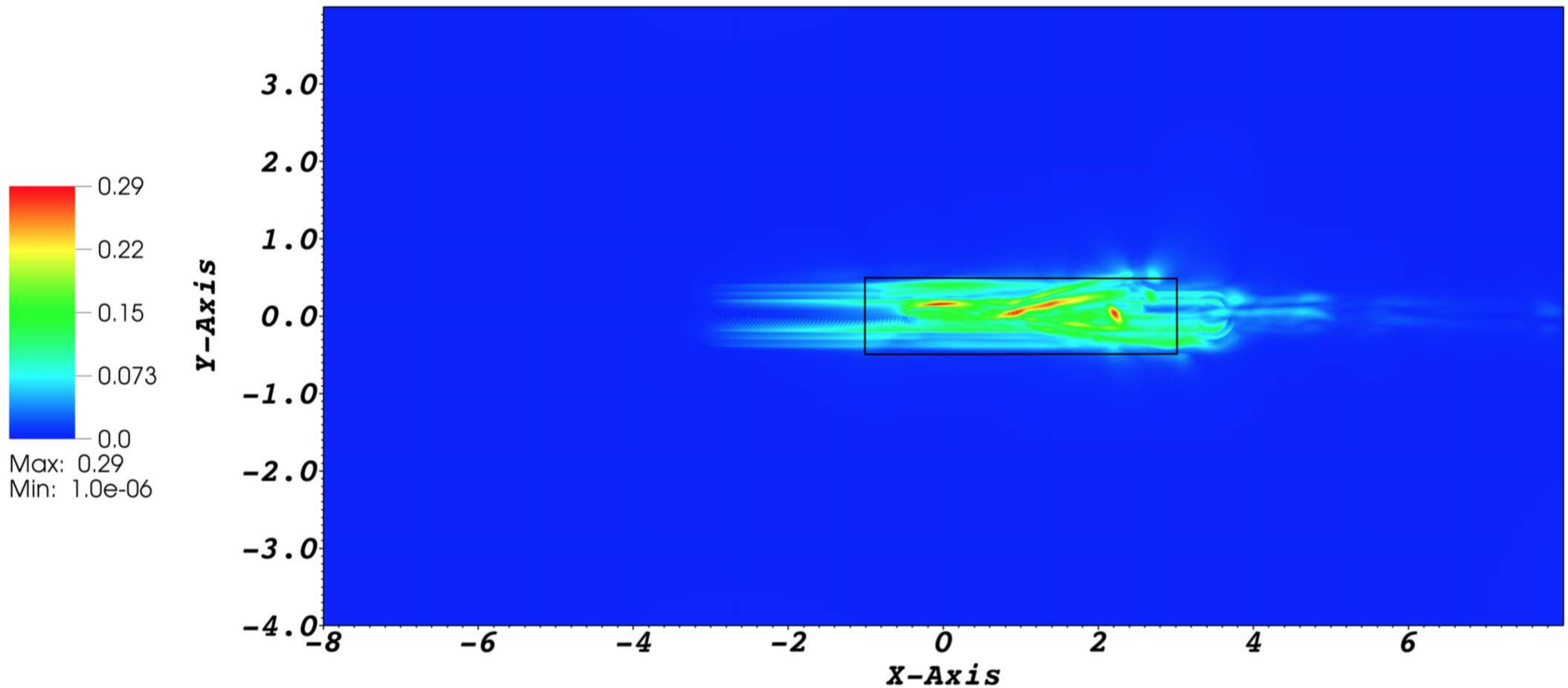


↪ More efficient sampling (CVT) can be used

Institut de Mathématiques de Bordeaux

High and low dimensional models

- Size of the DNS domain: leave one out in the input parameter space

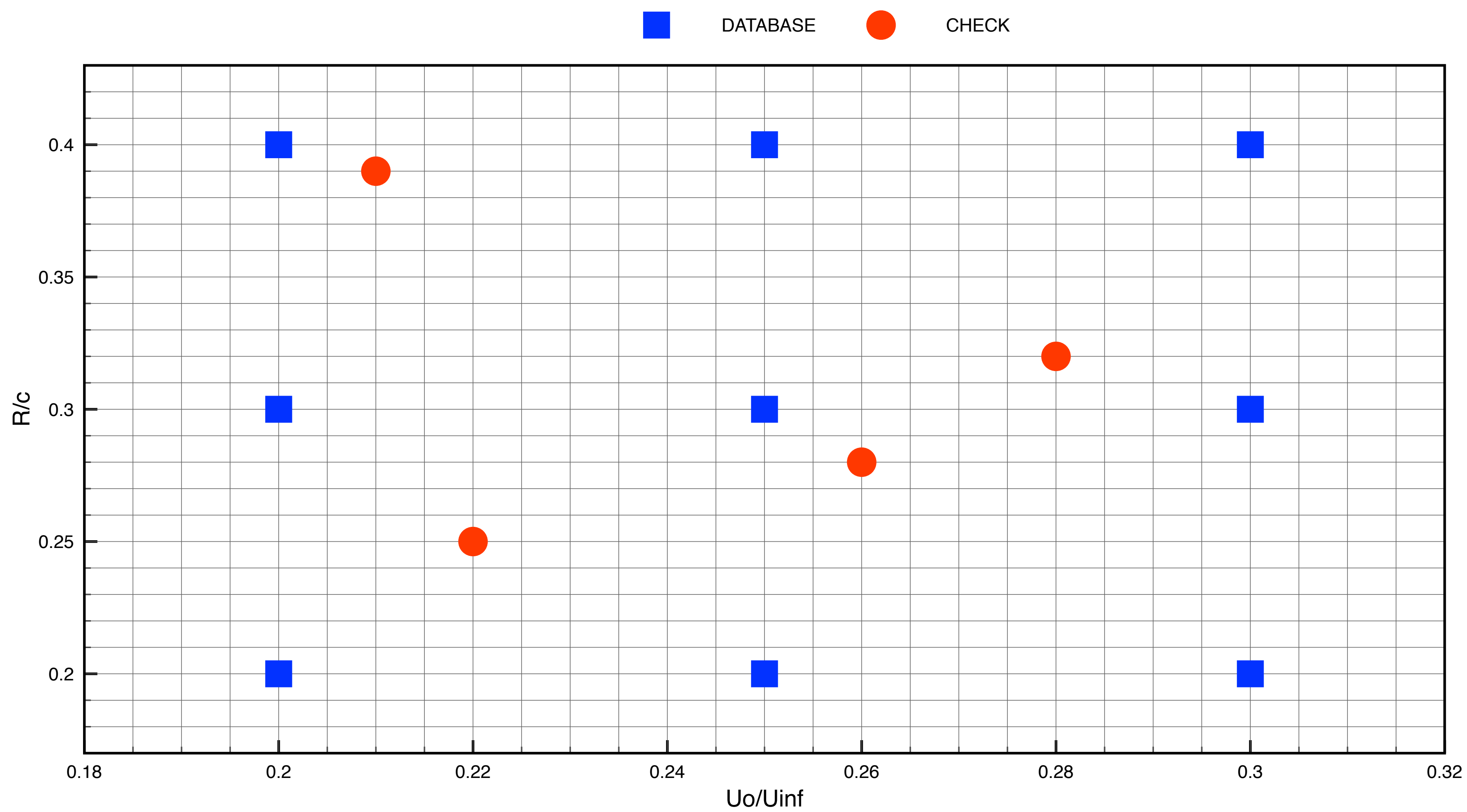


Institut de Mathématiques de Bordeaux

informatics mathematics

High and low dimensional models

- Gust prediction outside the POD database (robustness)



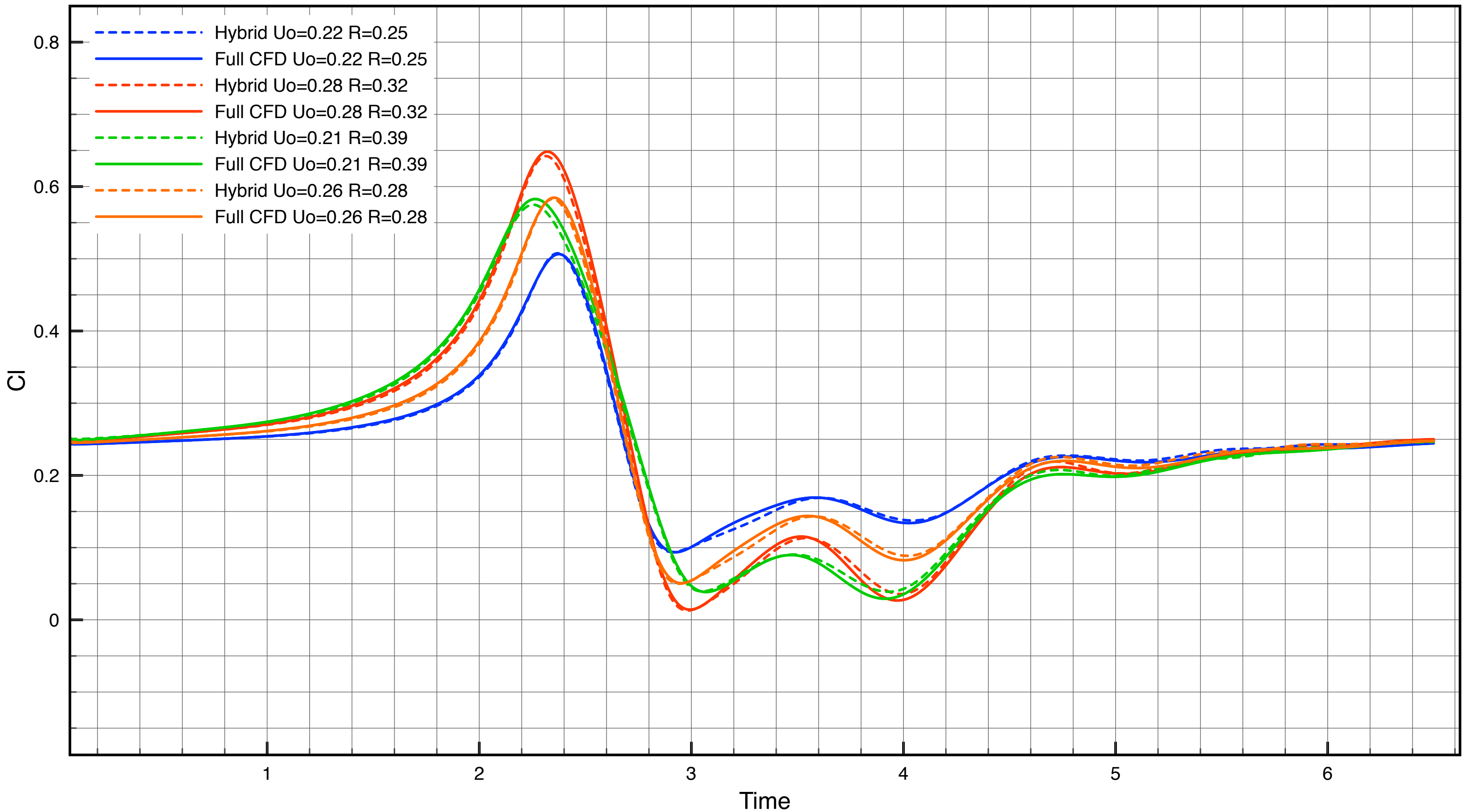
$\hookrightarrow$ test points in red

Institut de Mathématiques de Bordeaux

informatics mathematics

High and low dimensional models

- Test outside the database for gust prediction



Institut de Mathématiques de Bordeaux

$\hookrightarrow$ Good results for Gust prediction, saving a lot of CPU costs

Inria informatics mathematics

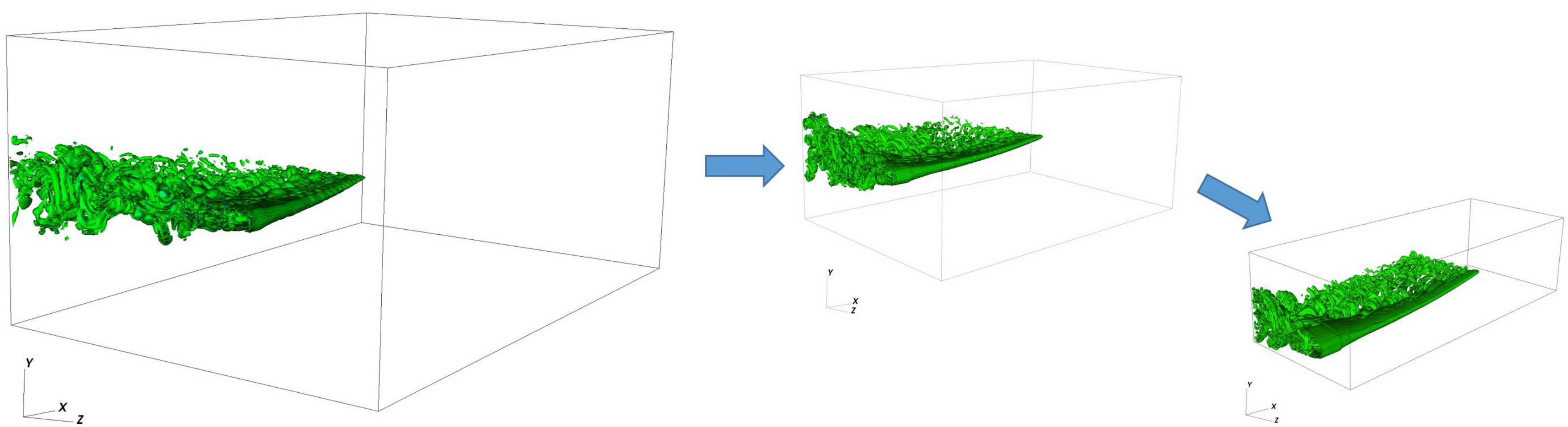
High and low dimensional models

▶ **Example: numerical zoom**

↪ This can be done iteratively

One sampling point

Leave one out based on snapshots clustering (K-means)



Numerical zoom of flow around an APX wind turbine blade (Cantilever beam)

Low Reynolds number (Re=1000), No twist, no rotation

Institut de Mathématiques de Bordeaux

informatics mathematics

Softwave

A versatil numerical code: NaSCar (Navier-Stokes Cartesien)

Librairies for obstacles

- ↪ *Importation and meshing of geometries*
- ↪ *3D synthetic body generation using B-splines (fishes)*

Librairies for fluid/structure interactions

- ↪ *Computation of hydrodynamic effects (forces and torques)*
- ↪ *One-way: imposed deformation and computation of the displacement using Newtons laws*
- ↪ *Two-way: strong implicit coupling, elastic beam model*

Librairies to track interfaces

- ↪ *Level set generation from given obstacle*
- ↪ *Level set transport (WENO5 and RK3 TVD)*
- ↪ *Reinitialization (Godunov type method for upwinding)*
- ↪ *Contour selection for mass conservation*

Librairies for Navier-Stokes on Cartesian mesh

- ↪ *Projection method: 2nd order Chorin-Temam approach*
- ↪ *Volume penalisation*
- ↪ *Immersed Boundaries*
- ↪ *Image Point Correction*
- ↪ *Bi-fluid with surface tension: CSF and GFM methos*
- ↪ *turbulence model: LES Smagorinsky-Lilly*

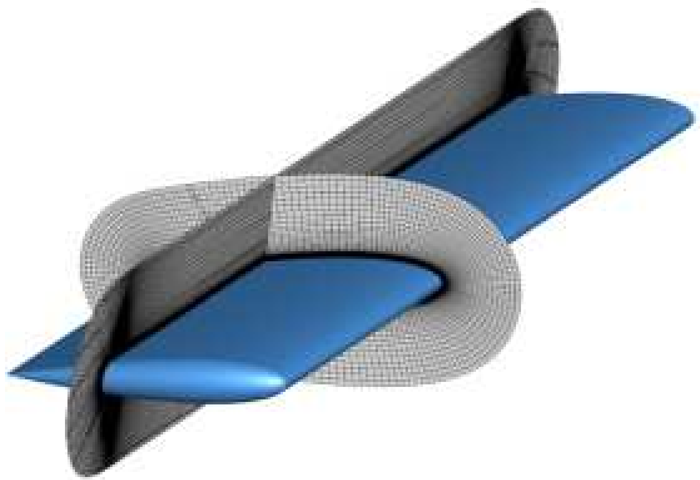
Institut de Mathématiques de Bordeaux

informatics mathematics Inria

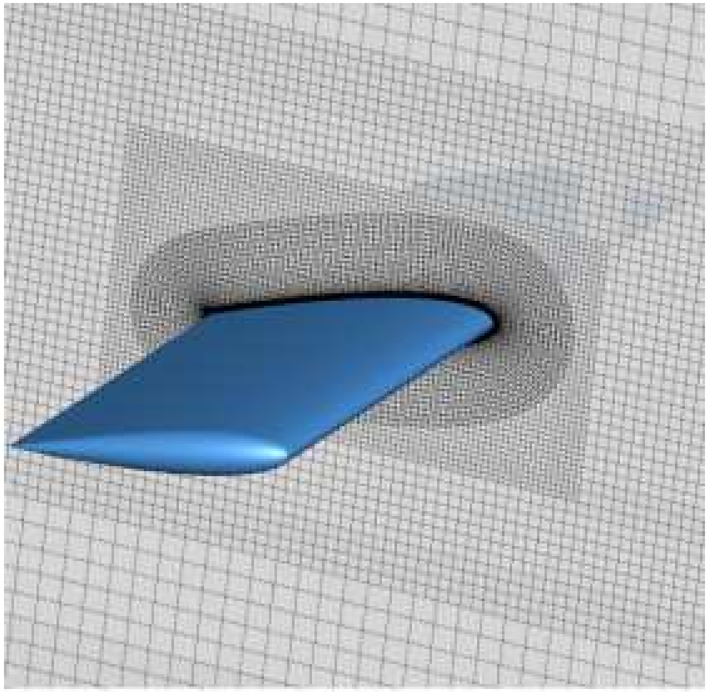
Conclusions

- **Improvement of the code**

  ↪ Overset Chimera meshes (F. Tesser PhD)



(a)



(b)

Stéphanie Péront, Christophe Benoit (ONERA), JCP 232 (1), 2013.

Institut de Mathématiques de Bordeaux

informatics mathematics

Conclusions

- **Improvement of the code**
  - ↪ Overset Chimera meshes (F. Tesser PhD)
  - ↪ Accurate and robust schemes (2nd order AND conservative)
  - ↪ Coupling with other "reduced order models" for BCs (shallow water?)
- **Lot of applications: real data!**
  - ↪ Zebrafish larvae (MRGM)
  - ↪ Flows with many particles (LOMA, CRPP-LOF)
  - ↪ Wind turbines (VALEOL, Cifre PhD)
  - ↪ Pelamis (to test two-way FSI)
  - ↪ ISWEC (W4E)
  - ↪ CorWave (Cifre PhD)
  - ↪ And more ...

Institut de Mathématiques de Bordeaux