

Implementation of a new discrete Immersed Boundary Method in OpenFOAM

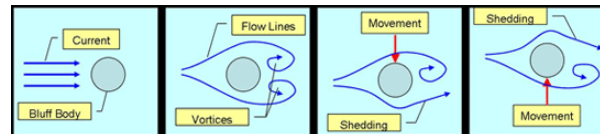
CONSTANT Eddy* , COLIN-BELLOT Clothilde, FAVIER Julien , MELIGA Philippe , SERRE Éric
Aix-Marseille Université, CNRS, Ecole Centrale Marseille, Laboratoire M2P2, UMR 7340, Marseille, France

*PhD student, funded by DGA

Context

Vortex Induced Vibration (VIV) affect

Stealth of military aircrafts and cause fatigue damage on the risers used in offshore petroleum production.



Objectives

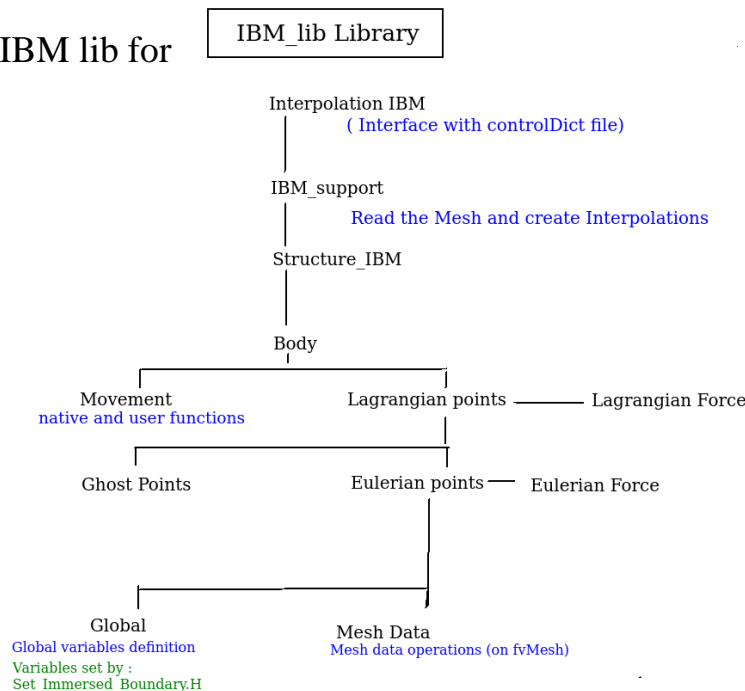
VIV modeling and creation of optimal control laws

Solver

Implementation of a solver able to model **fluid – structure interactions**

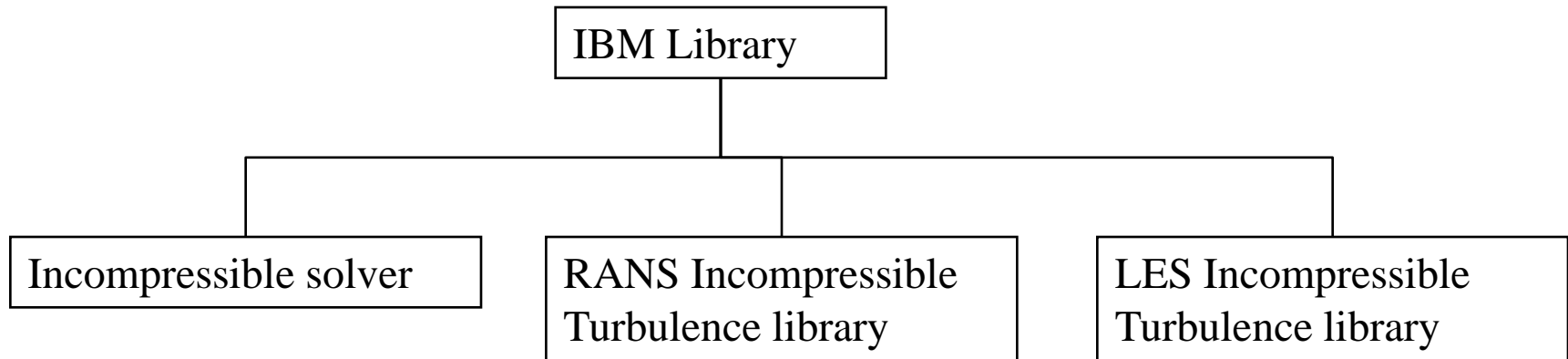
- Implementation of an **object oriented library** « Immersed Boundary Method » (Pinelli & al .JCP 2010) in the **Openfoam** code (tested in version 2.1.1 ; 2.2.1 ;2.2.2; 2.2.X; 3.0.1)
- Able to handle **complex, moving and deformable bodies**

Tree of the IBM Library IBM lib for OpenFOAM.



IBM Library integrated in 3 OpenFoam Libraries

:



- PISO solver

- Spalart Allmaras v 2.X.X
- Spalart Allmaras v 3.X.X

- DDES Spalart Allmaras v 2.X.X
- DDES Spalart Allmaras v 3.X.X

Volume force integrated in N-S equations to model the Geometry :

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} + \mathbf{f}(\mathbf{u})$$

With :

Force term on a *lagrangian grid*

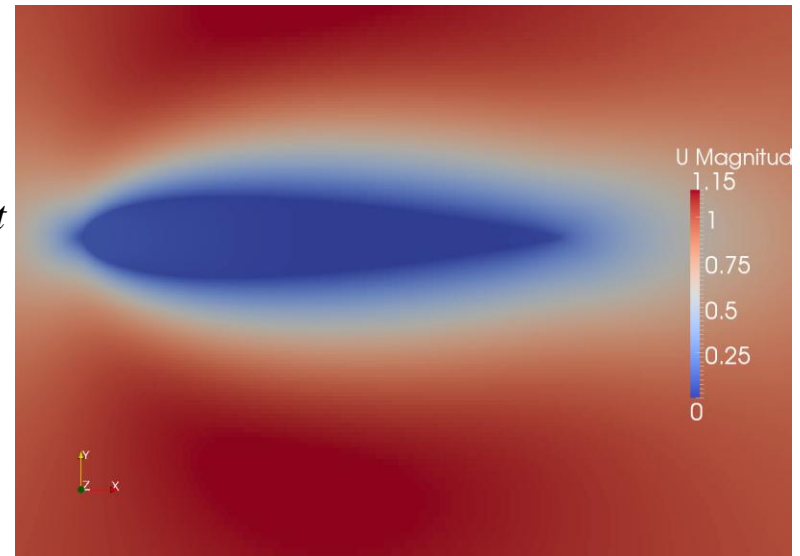
$$\mathbf{F}_s = \frac{\mathbf{U}_s^d - \mathcal{I}[\mathbf{u}]_s}{\Delta t} \quad \text{and} \quad U^d = U \text{ desired on a lagrangian point}$$

Interpolation of the velocity on the lagrangian grid:

$$\mathcal{I}[\mathbf{u}]_s = \sum_{j \in D_s} \mathbf{u}_j \mathbf{W}_d(\mathbf{x}_j - \mathbf{X}_s) \Delta v$$

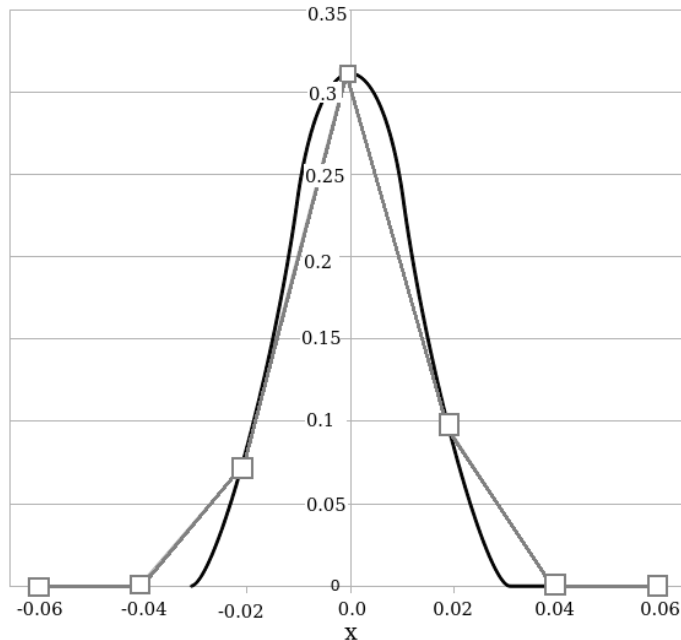
Spreading the force on the eulerian mesh:

$$\mathcal{S}[\mathbf{F}_k] = \mathbf{f}(\mathbf{x}_j) = \sum_{k \in D_j} \mathbf{F}_k \mathbf{W}_d(\mathbf{x}_j - \mathbf{X}_k) \epsilon_k$$

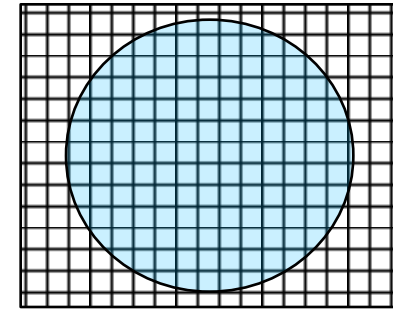


NACA 0016 with Immersed Boundary Method

IBM delta function for an **uniform cartesian mesh**



Eulerian discretization of the IBM force term f around a Lagrangian marker: comparison between 2nd order discretization using a centered scheme and the kernel analytical solution



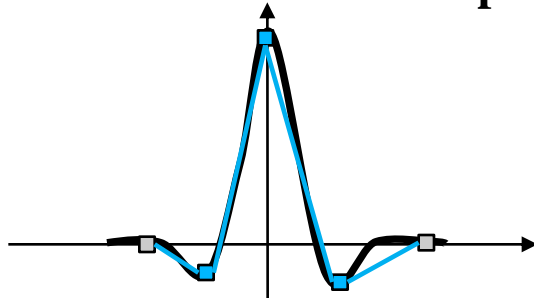
*Immersed boundary on an **uniform mesh***

$$w_d(r) = \begin{cases} \frac{1}{6} \left(5 - 3|r| - \sqrt{-3(1 - |r|)^2 + 1} \right) & 0.5 \leq |r| \leq 1.5 \\ \frac{1}{3} \left(1 + \sqrt{-3r^2 + 1} \right) & |r| \leq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

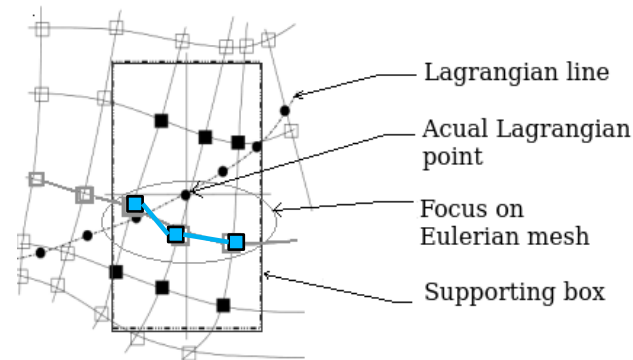
where $r = (x - s)/d$ that satisfies the following properties:

1. $w_d(r)$ is continuous $\forall r \in \mathbb{R}$;
2. $w_d(r) = 0$ if $|r| \geq 1.5$;
3. $\sum_l w_d(r - l) = 1, \forall l \in \mathbb{N}$;
4. $\sum_l (r - l)w_d(r - l) = 0, \forall r, l$;

IBM delta function with **rkpm** method for non-uniform mesh



Eulerian discretization of the IBM force term f around a Lagrangian marker: comparison between 2nd order discretization using a centered scheme and the kernel analytical solution



*Zoom on the mesh discretization of an Immersed boundary on a **non-uniform mesh***

Modification of the delta function following Liu et al. 1995:

$$\tilde{w}_d(x - s) = \sum_{i=0}^n b_i(y, d)(x - s)^i w_d(x - s),$$

where the unknown polynomial coefficients $b_i(y, d)$ are determined by imposing the reproducing conditions

$$\tilde{m}_0(x) = \int_{\Omega} \tilde{w}_d(x - s) ds = 1$$

$$\tilde{m}_i(x) = \int_{\Omega} (x - s)^i \tilde{w}_d(x - s) ds = 0 \quad (i = 1, \dots, N),$$

IBM PisoFoam

Step 1 : Calculation of a velocity predictor for the (IBM), \hat{u} , by solving the momentum equation without the (IBM) force term:

$$\frac{\partial \hat{u}}{\partial t} + \nabla \cdot (\hat{u}\hat{u}) = -\nabla p + \frac{1}{Re} \nabla^2 \hat{u}$$



Step 2: Computation of the (IBM) force term $f^{n+1}(\hat{u}^{n+1})$ from equation (7) on the Eulerian mesh.



Step 3: Compute a first predictor velocity field to initialize the PISO iterative loop, $u^{*,1}$, ($\nabla \cdot u^{*,1} \neq 0$), by solving the discretized momentum equation (1) including the (IBM) force term $f(\hat{u})$ from step 2:

$$\frac{\partial u^{*,1}}{\partial t} + \nabla \cdot (u^{*,1}u^{*,1}) = -\nabla p + \frac{1}{Re} \nabla^2 u^{*,1} + f(\hat{u}) \quad (A.11)$$



Iterative PISO loop for $m = 1, \dots, M$



Step 4: Compute $[H]$ & a_{ii}^{-1}

Step 5: Solve the Poisson problem

$$\nabla \cdot [\nabla p^{*,m+1}] = \nabla \cdot [H] + \nabla \cdot f(\hat{u})$$



Max PISO iterations not reached

Step 6: (Velocity Correction)

$$[u^{*,m+1}] = \{a_{ii}^{-1}\}(-\nabla p^{*,m+1} + f(\hat{u}) + [H])$$

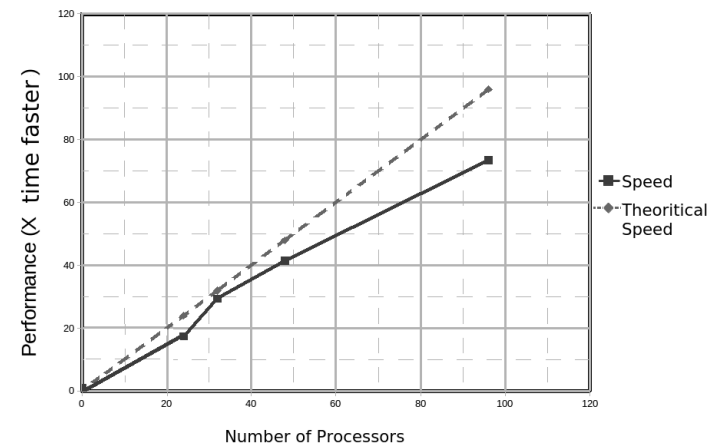
Max PISO iterations reached



$$u^{n+1} = u^{*,M}$$

$$p^{n+1} = p^{*,M}$$

- **Successful modification of the PISO solver** to import the IBM library
- **Submitted paper** in « Computers and fluids »
- **Good scalability** of the solver with the IBM library



Performance (scalability) of the solver as function of the number of processors for a 2D simulation of a flow at $Re = 500$ past a moving cylinder using 10^6 Eulerian points and 312 Lagrangian markers.

Improvement of the near wall accuracy of the method, analytical calculation of the force derivative :

Step 5: *Solve the Poisson problem*

$$\nabla \cdot [\nabla p^{*,m+1}] = \nabla \cdot [H] + \nabla \cdot f(\hat{u})$$

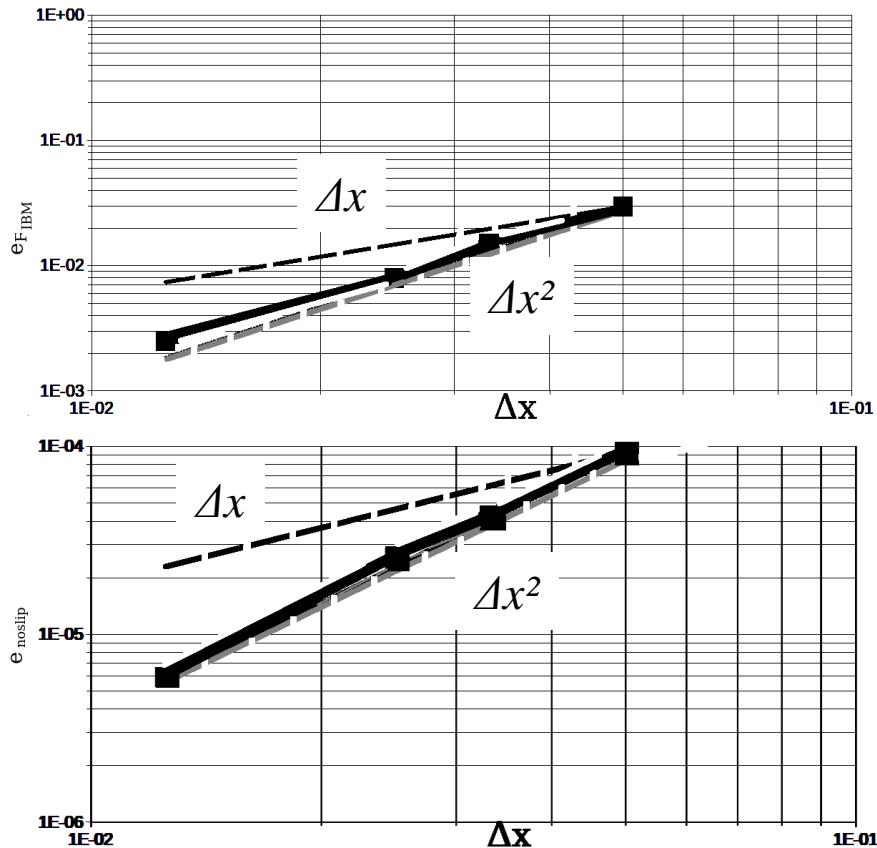
Classical boundary

IBM without correction

IBM with correction



Plots of the velocity divergence around a 2D cylinder at Re = 30



Convergence order for the global force and no slip condition error

Global Force Error :

$$e_{F_{IBM}} = \left| \sum_{k \in D_j} (\mathbf{F}_k - \mathbf{F}_a) \epsilon_k \right|$$

where :

$$\mathbf{F}_a = \frac{\mathbf{U}_k^d - \mathbf{U}_a}{\Delta t}$$

No slip condition Error :

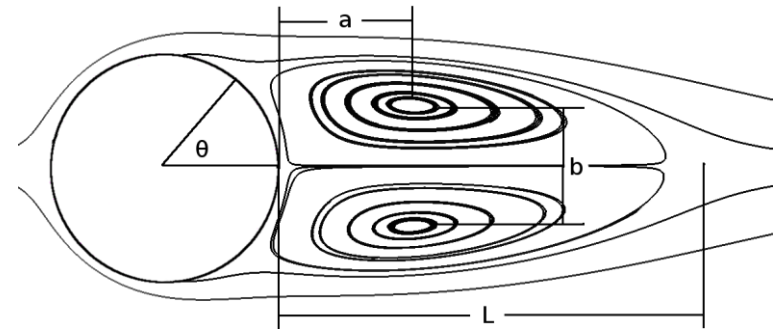
$$e_{noslip} = \| U_s - \mathcal{I}[S[U_k]]_s \|_{\infty}$$

Validation on well-documented test cases

1st test case : 2D fixed cylinder

Re = 30

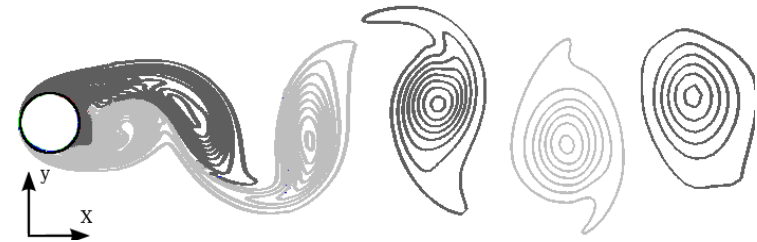
| | a/D | b/D | Cd |
|-----------------------|-------|------|------|
| Non uniform dx 2,5e-2 | 0,6 | 0,52 | 1,84 |
| Non uniform dx 2e-3 | 0,58 | 0,52 | 1,77 |
| Uniform dx 2.10e-2 | 0,556 | 0,53 | 1,78 |
| Uniform dx 10e-2 | 0,55 | 0,53 | 1,77 |
| Pinelli et al. | 0,56 | 0,52 | 1,8 |
| Blackburn & Henderson | | | 1,74 |
| Coutanceau & Bouard | 0,54 | 0,54 | |
| Tritton | | | 1,74 |



Geometrical parameters of the *wake* for a steady flow at $Re = 30$

Re = 185

| | CL rms | Cd mean | St |
|------------------------|--------|---------|-------|
| Non Uniform 2,5.10e-2 | 0,447 | 1,509 | 0,191 |
| Uniform 2.10e-2 | 0,436 | 1,387 | 0,198 |
| Uniform 10e-2 | 0,427 | 1,379 | 0,198 |
| Pinelli et al. 5.10e-3 | 0,423 | 1,43 | 0,196 |
| Pinelli et al. 1.10e-2 | 0,428 | 1,509 | 0,199 |
| Vanella and Balaras | 0,461 | 1,377 | |
| Guilmineau and Queutey | 0,443 | 1,287 | 0,195 |
| Lu and Dalton | 0,422 | 1,31 | 0,195 |
| Williamson | | | 0,193 |

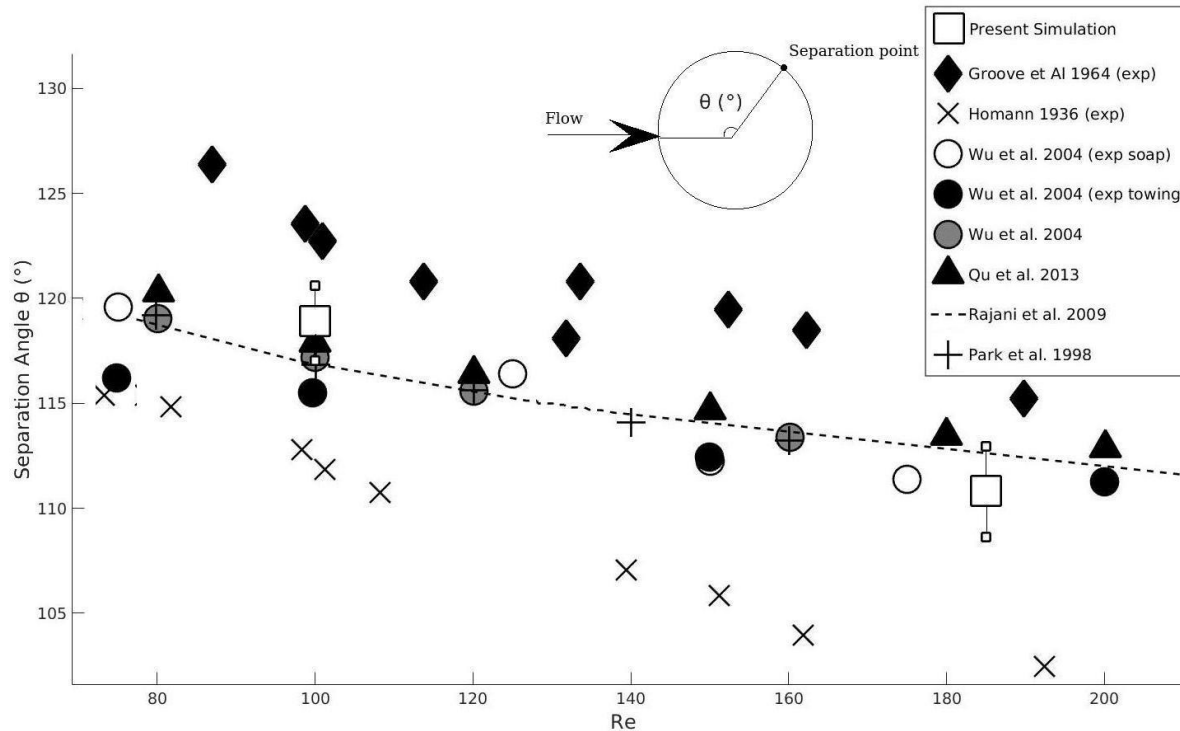


Vorticity contours evidencing the *shedding of large-scale vortices* in 2D flow past a fixed circular cylinder at $Re = 185$

Geometrical parameters of the wake, lift, drag coefficient and strouhal for the configuration of a fixed cylinder. Numerical and experimental data from literature are provided for comparison.

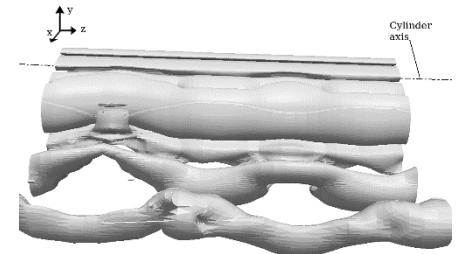
Validation on well-documented test cases

2nd test case : 3D fixed cylinder

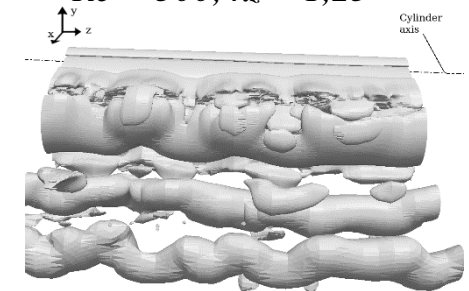


Mean separation angle as a function of the Reynolds number. Error bars corresponding to the min/max values achieved during the duration of the averaging process are shown by the vertical lines on either side of the square symbols.

$Re = 200; \lambda_z = 4,5$



$Re = 300; \lambda_z = 1,25$

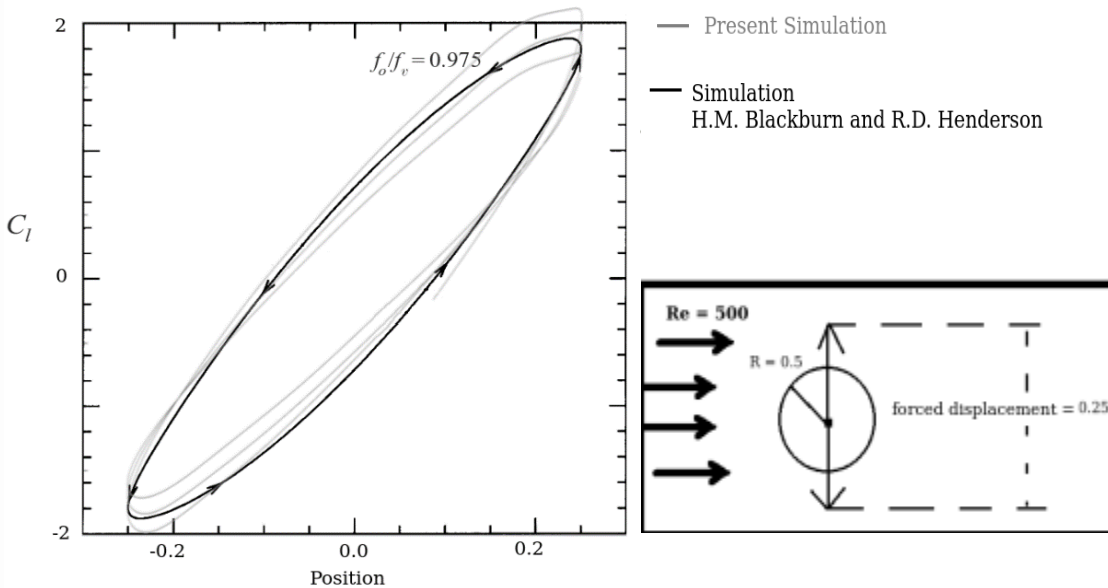


Iso-surfaces of the instantaneous **Q-criterion** ($-0.8 < Q < 0.8$)

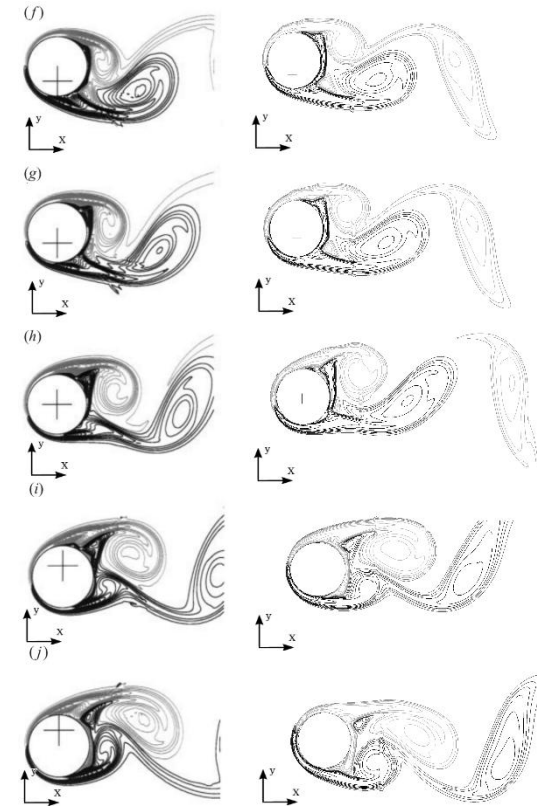
3rd test case : 2D moving cylinder

Re = 500 with **forced oscillations**

$$F_z \text{ ratio} = St / f_z \text{ oscil} = 0,975; A = 0,25 D$$



Lift coefficient C_L as a function of the cylinder displacement for the 2D flow past an oscillating cylinder at $Re=500$: present results (grey line) vs. Results obtained by Blackburn and Henderson 1999.



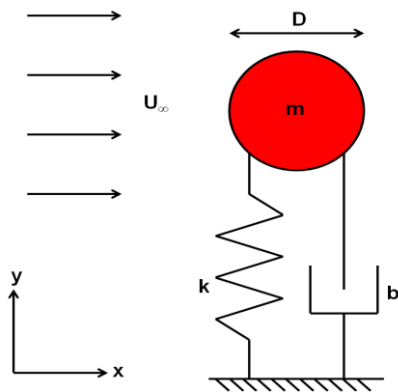
*Comparisons of the instantaneous vorticity contours between the results obtained by H.M.Blackburn and R.D.Henderson (1999) (left column) and present results (right column) on the conguration of **moving cylinder** at $Re = 500$. At 5 different instants (f to j)*

Validation on well documented test cases

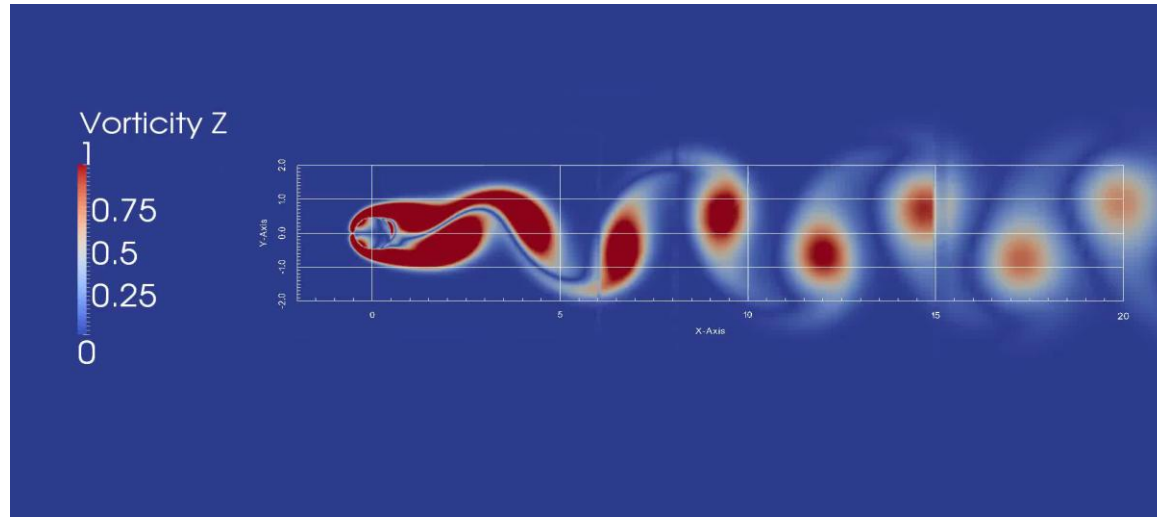
4th test case : 2D moving cylinder

Re = 100 with free oscillations

$$m\ddot{y} + b\dot{y} + ky = F_y$$



Scheme of the *free oscillations model*



Absolute value of *vorticity contours* evidencing the *shedding of large-scale vortices* in 2D flow past a moving circular cylinder with *free oscillations* at Re = 500

| Source | Present | Shiels et al. | Shen et al. | Lee et al. |
|--------------------|---------|---------------|-------------|------------|
| A | 0.539 | 0.58 | 0.57 | 0.64 |
| f | 0.17 | 0.196 | 0.190 | 0.180 |
| C _{D,avg} | 2.0854 | 2.22 | 2.15 | 2.11 |

Amplitude, Frequency and Drag for the configuration of a moving cylinder with *free oscillations*. Numerical and experimental data from literature are provided for comparison.

1st study: turbulent flows

Creation of turbulent dynamics libraries :

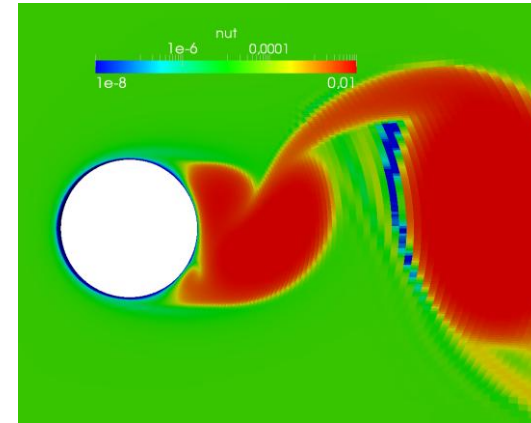
- incompressibleIBMRAS.so
- incompressibleIBMLEs.so

Includes for now:

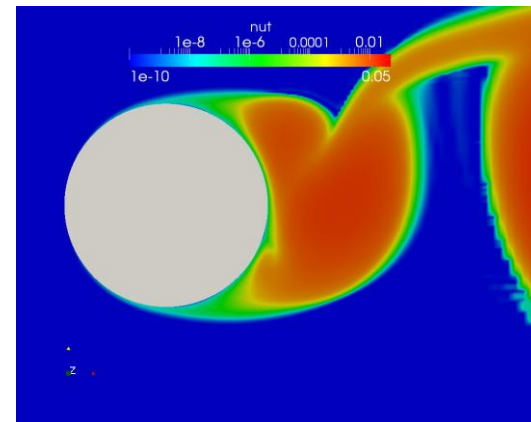
- IBM_Spalart_Allmaras
- IBM_DDES_Spalart_Allmaras

IBM on non-uniform meshes

Classical Boundary : $Re = 3900$



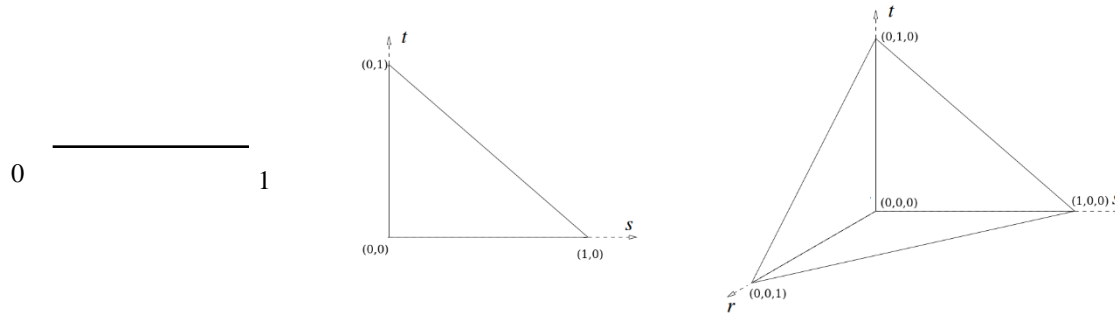
Immersed Boundary : $Re = 3900$



ν_t contours in 2D flow past a fixed circular cylinder

2nd study: deformable bodies

Finite Element Method



P1 Elements in 1,2 and 3D

Small displacement assumption :

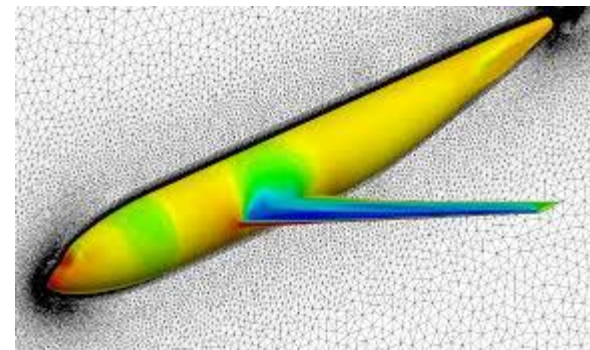
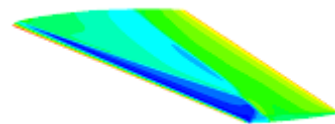
$$\begin{aligned}
 \mathbf{E} &= \frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{1}) \\
 &= \frac{1}{2} (\mathbf{D}_X \boldsymbol{\varphi}^T \mathbf{D}_X \boldsymbol{\varphi} - \mathbf{1}) \\
 &= \frac{1}{2} (\mathbf{D}_X (\mathbf{u} + \mathbf{X})^T \mathbf{D}_X (\mathbf{u} + \mathbf{X}) - \mathbf{1}) \\
 &= \frac{1}{2} (\mathbf{D}_X \mathbf{u}^T \mathbf{D}_X \mathbf{u} + \mathbf{D}_X \mathbf{u} + \mathbf{D}_X \mathbf{u}^T) \\
 &\approx \frac{1}{2} (\mathbf{D}_X \mathbf{u} + \mathbf{D}_X \mathbf{u}^T)
 \end{aligned}$$

Behaviour law : generalized Hook's law for isotropic material

$$\sigma_{ij} = \frac{E}{1 + \nu} \left(\varepsilon_{ij} + \frac{\nu}{1 - 2\nu} \varepsilon_{kk} \delta_{ij} \right)$$

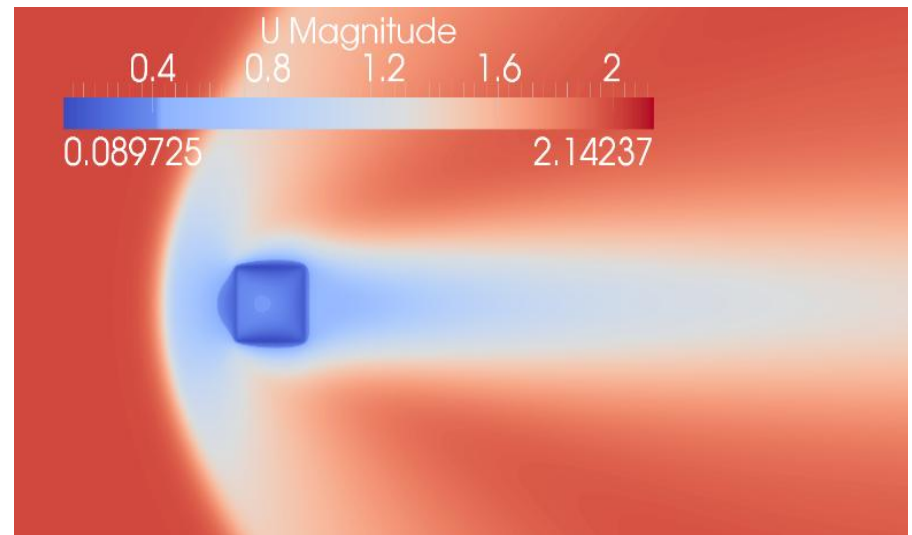
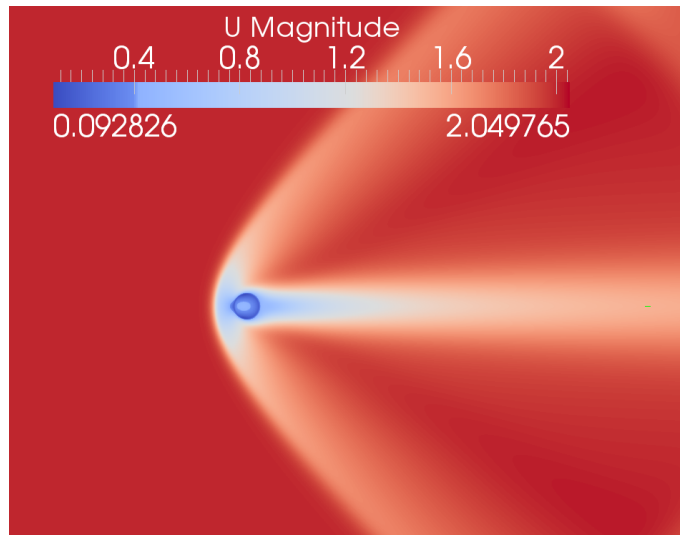
Solver validated on a cantilever beam.

Target geometries :



3rd study: compressible flows

Modification of the solver sonicFoam by the team of Pprime* with the IBM library



U contours calculated by IBM_sonicFoam on a circular and square cylinder

In collaboration with : RIAHI Hamza, MELDI Marcello, GONCALVES DA SILVA Eric

Include IBM in your solver

```
34 #include "fvCFD.H"
35 #include "singlePhaseTransportModel.H"
36 #include "turbulenceModel.H"
37 #include "Interpolation_IBM.C" ← Before Main to declare the IBM structure
38 // * * * * *
39
40 int main(int argc, char *argv[])
41 {
42     #include "setRootCase.H"
43
44     #include "createTime.H"
45     #include "createMesh.H"
46     #include "createFields.H"
47     #include "initContinuityErrs.H"
48     #include "Set Immersed Boundary.H" ← After main to declare the variable
49     // * * * * *
50     and read in the controlDict file
```

Include IBM in your solver

```

50
51 interpolation_ibm.Create_IBM_case( &mesh, runTime,Info);
52
53 Info<< "\nStarting time loop\n" << endl;
54
55 while (runTime.loop())
56 {
57     Info<< "Time = " << runTime.timeName() << nl << endl;
58
59     #include "readPISOControls.H"
60     #include "CourantNo.H"
61
62     // Pressure-velocity PISO corrector
63     {
64         // Momentum predictor
65
66         fvVectorMatrix UEqn
67         (
68             fvm::ddt(U)
69             + fvm::div(phi, U)
70             + turbulence->divDevReff(U)
71         );
72
73         //UEqn.relax();
74
75
76         solve(UEqn == -fvc::grad(p));
77
78
79
80 interpolation_ibm.Force_Velocity_Interpolation( &mesh, runTime,Info, U, p, forceIB,div_kernel_force,d);
81

```

Add command line before the loop of the solver to create the structure IBM

Add command line during the loop to calculate the forceIB (interpolation of the force), div_kernel_force (analytical gradient of the force) and d, the divergence of the field

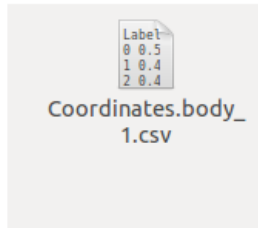
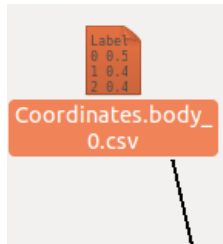
Can be any scalar field to calculate on the wall

Futher efforts

- Implementation of an adjoint solver coupled with the present IBM Method to achieve the control analysis of the VIVs.

Do you have any question ?

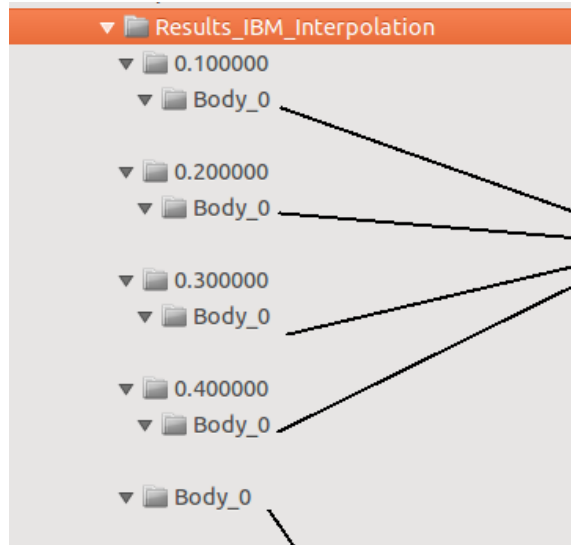
Input Files



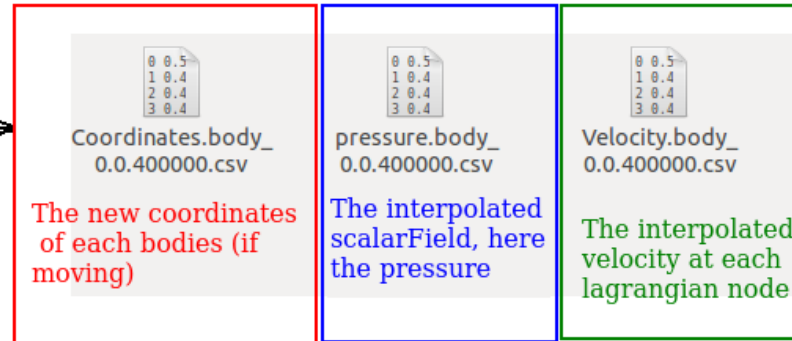
Label of the lagrangian points begin by 0

| | A | B | C | D |
|----|----------------|---|--------------|---|
| 1 | Label X | | Y | Z |
| 2 | 0 0.5 | | | 0 |
| 3 | 1 0.4998986145 | | 0.0100685266 | 0 |
| 4 | 2 0.4995944991 | | 0.0201329701 | 0 |
| 5 | 3 0.4990877771 | | 0.0301892487 | 0 |
| 6 | 4 0.4983786541 | | 0.0402332844 | 0 |
| 7 | 5 0.4974674175 | | 0.0502610037 | 0 |
| 8 | 6 0.496354437 | | 0.0602683401 | 0 |
| 9 | 7 0.495040164 | | 0.0702512352 | 0 |
| 10 | 8 0.4935251313 | | 0.0802056404 | 0 |

Output Files



For each write time :



For each time step :

