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
Object Oriented Library

- 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:

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

With:

**Force term on a lagrangian grid**

\[
\mathbf{F}_s = \left( \frac{U^d_s - \mathcal{I}[\mathbf{u}]_s}{\Delta t} \right)
\]

and

\[U^d = U\text{ desired on a lagrangian point}\]

**Interpolation of the velocity on the lagrangian grid:**

\[
\mathcal{I}[\mathbf{u}]_s = \sum_{j \in D_s} \mathbf{u}_j \mathbf{w}_d(x_j - X_s) \Delta v
\]

**Spreading the force on the eulerian mesh:**

\[
\mathcal{S}[\mathbf{F}_k] = f(x_j) = \sum_{k \in D_j} \mathbf{F}_k \mathbf{w}_d(x_j - X_k) \epsilon_k
\]

NACA 0016 with Immersed Boundary Method
Immersed Boundary Method

IBM delta function for an **uniform cartesian 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

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

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

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

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

\[
\frac{\partial \tilde{u}}{\partial t} + \nabla \cdot (\tilde{u} \tilde{u}) = -\nabla p + \frac{1}{Re} \nabla^2 \tilde{u}
\]

\[
\downarrow
\]

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

\[
\downarrow
\]

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 \( f(\tilde{u}) \) from step 2:

\[
\frac{\partial u^{*,1}}{\partial t} + \nabla \cdot (u^{*,1} u^{*,1}) = -\nabla p + \frac{1}{Re} \nabla^2 u^{*,1} + f(\tilde{u})
\]  

(A.11)

\[
\downarrow
\]

Iterative PISO loop for \( m = 1, ..., M \)

Step 4: Compute \([H]\) & \( a_{n}^{1}\)

Step 5: Solve the Poisson problem

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

\[
\uparrow
\]

Max Piso Iterations not reached

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

Max Piso iterations reached

\[
\downarrow
\]

\[
y^{n+1} = u^{*,M}
\]

\[
p^{n+1} = p^{*,M}
\]

- **Successfull 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 \( 10^8 \) Eulerian points and 312 Lagrangian markers.
IBM PisoFoam

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

max : 0.114  max : 0.632  max : 0.097

Plots of the velocity divergence around a 2D cylinder at Re = 30
Verification: Manufactured solution

**Global Force Error**: 
\[ e_{F_{IBM}} = \left| \sum_{k \in D_j} (F_k - F_a) \epsilon_k \right| \]

where:
\[ F_a = \frac{U_k^d - U_a}{\Delta t} \]

**No slip condition Error**:
\[ e_{\text{noslip}} = \| U_s - \mathcal{I} \{ S[U_k] \}_s \|_\infty \]

*Convergence order* for the global force and no slip condition error.
Validation on well-documented test cases

1st test case: 2D fixed cylinder

Re = 30

<table>
<thead>
<tr>
<th></th>
<th>a/D</th>
<th>b/D</th>
<th>Cd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non uniform dx 2.5e-2</td>
<td>0.6</td>
<td>0.52</td>
<td>1.84</td>
</tr>
<tr>
<td>Non uniform dx 2e-3</td>
<td>0.58</td>
<td>0.52</td>
<td>1.77</td>
</tr>
<tr>
<td>Uniform dx 2.10e-2</td>
<td>0.56</td>
<td>0.53</td>
<td>1.78</td>
</tr>
<tr>
<td>Uniform dx 10e-2</td>
<td>0.55</td>
<td>0.53</td>
<td>1.77</td>
</tr>
<tr>
<td>Pinelli et al.</td>
<td>0.56</td>
<td>0.52</td>
<td>1.6</td>
</tr>
<tr>
<td>Blackburn &amp; Henderson</td>
<td></td>
<td></td>
<td>1.74</td>
</tr>
<tr>
<td>Coutanceau &amp; Bouard</td>
<td>0.54</td>
<td>0.54</td>
<td>1.74</td>
</tr>
<tr>
<td>Tritton</td>
<td></td>
<td></td>
<td>1.74</td>
</tr>
</tbody>
</table>

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.

Re = 185

<table>
<thead>
<tr>
<th></th>
<th>Clrms</th>
<th>Cdmean</th>
<th>St</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non Uniform 2.5.10e-2</td>
<td>0.447</td>
<td>1.509</td>
<td>0.191</td>
</tr>
<tr>
<td>Uniform 2.10e-2</td>
<td>0.436</td>
<td>1.387</td>
<td>0.198</td>
</tr>
<tr>
<td>Uniform 10e-2</td>
<td>0.427</td>
<td>1.379</td>
<td>0.198</td>
</tr>
<tr>
<td>Pinelli et al. 5.10e-3</td>
<td>0.428</td>
<td>1.431</td>
<td>0.196</td>
</tr>
<tr>
<td>Pinelli et al. 1.10e-2</td>
<td>0.428</td>
<td>1.509</td>
<td>0.199</td>
</tr>
<tr>
<td>Vanneste and Belanec</td>
<td>0.461</td>
<td>1.377</td>
<td>0.199</td>
</tr>
<tr>
<td>Guflmnieu and Queutey</td>
<td>0.443</td>
<td>1.287</td>
<td>0.195</td>
</tr>
<tr>
<td>Lu and Dalton</td>
<td>0.422</td>
<td>1.311</td>
<td>0.196</td>
</tr>
<tr>
<td>Williamson</td>
<td></td>
<td>0.193</td>
<td></td>
</tr>
</tbody>
</table>

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

Iso-surfaces of the instantaneous $Q$-criterion ($-0.8 < Q < 0.8$)
Validation on well documented test cases

3rd test case: 2D moving cylinder

$$F_z \text{ ratio } = \frac{St}{f_\text{ oscil}} = 0.975; \ A = 0.25 \ D$$

Re = 500 with **forced oscillations**

---

**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 configuration 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

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

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

<table>
<thead>
<tr>
<th>Source</th>
<th>Present</th>
<th>Shiels et al.</th>
<th>Shen et al.</th>
<th>Lee et al.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.539</td>
<td>0.58</td>
<td>0.57</td>
<td>0.64</td>
</tr>
<tr>
<td>f</td>
<td>0.17</td>
<td>0.196</td>
<td>0.190</td>
<td>0.180</td>
</tr>
<tr>
<td>(C_{D,avg})</td>
<td>2.0854</td>
<td>2.22</td>
<td>2.15</td>
<td>2.11</td>
</tr>
</tbody>
</table>

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

**Vt contours** in 2D flow past a fixed circular cylinder
2nd study: deformable bodies

Finite Element Method

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:

Small displacement assumption:

\[
E = \frac{1}{2} (F^T F - 1) \\
= \frac{1}{2} (D_x \varphi^T D_x \varphi - 1) \\
= \frac{1}{2} \left( D_x (u + X)^T D_x (u + X) - 1 \right) \\
\approx \frac{1}{2} \left( D_x u^T D_x u + D_x u + D_x u^T \right)
\]
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

*-Inst PPrime, UPR 3346, CNRS, Université de Poitiers, ENSMA
Include IBM in your solver

```c
#include "fvCFD.H"
#include "singlePhaseTransportModel.H"
#include "turbulenceModel.H"
#include "Interpolation_IBC.C"

// ***********

int main(int argc, char *argv[])
{
    #include "setRootCase.H"

    #include "createTime.H"
    #include "createMesh.H"
    #include "createFields.H"
    #include "initContinuityErrs.H"
    #include "Set Immersed Boundary.H"

    // ***********
```
Include IBM in your solver

```cpp
Interpolation_ibm.Create_IBM_case(&mesh, runTime, Info);

Info<< "Starting time loop" << endl;

while (runTime.loop())
{
    Info<< "Time = " << runTime.timeName() << nl << endl;

    #include "readPISOC Controls.H"
    #include "CourantNo.H"

    // Pressure-velocity PISO corrector
    // Momentum predictor
    fvVectorMatrix UEqn
    (
        fvm::ddt(U)
        + fvm::div(phi, U)
        + turbulence->divDevReff(U)
    );

    //UEqn.relax();

    solve(UEqn == -fvc::grad(p));

    interpolation_ibm.Force_Velocity_Interpolation( &mesh, runTime, Info, U, p, forceIB, div_kernel_force, d);
```
Further 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

**Coordinates.body_0.csv**

**Coordinates.body_1.csv**

Label of the Lagrangian points begin by 0

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Label</strong></td>
<td><strong>X</strong></td>
<td><strong>Y</strong></td>
<td><strong>Z</strong></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1.04998986145</td>
<td>0.0100685266</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2.04995944991</td>
<td>0.0201329701</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>3.04990877771</td>
<td>0.0301892487</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>4.04983786541</td>
<td>0.0402332844</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>5.04974674175</td>
<td>0.0502610037</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>6.0496354437</td>
<td>0.0602683401</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>7.0495040164</td>
<td>0.0702512352</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>8.04935251313</td>
<td>0.0802056404</td>
<td>0</td>
</tr>
</tbody>
</table>
Output Files

For each write time:

- Coordinates.body_0.0.400000.csv
- pressure.body_0.0.400000.csv
- Velocity.body_0.0.400000.csv

The new coordinates of each bodies (if moving)
The interpolated scalarField, here the pressure
The interpolated velocity at each lagrangian node

For each time step:

- Coordinates.body_0.csv
- displacement.csv
- force.csv
- velocity.csv

If the lagrangian points are sorted write for the 1st it the lagrangian points selected
Displacement of the first lagrangian point (if moving)
Global force of the fluid on each body
Velocity of the first lagrangian point (if moving)