A New Fluid-Structure Interaction Solver in OpenFOAM

Do Quoc Vu, Pham Van Sang*

Hanoi University of Science and Technology - No. 1, Dai Co Viet Str., Hai Ba Trung, Ha Noi, Vietnam

Received: May 28, 2018; Accepted: June 24, 2019

Abstract

In this article, we introduce a new numerical solver for the Fluid-Structure Interaction problem. The solver is developed using Immersed Boundary Method (IBM) integrated into OpenFOAM environment. The velocity-pressure coupling is implemented via modifying the PISO algorithm of OpenFOAM. The solver can solve for the interaction of multiple structures in fluid flow. The collision of structures is simulated using an elastic repulsive force model. A parallel algorithm is developed to make the solver able to run on parallel computer system, structures can move from a partition to another partition. The solver was validated and applied in solving the real problem.

Keywords: Immersed boundary method, Fluid-Structure interaction, IBMFoam, particulated flow

1. Introduction

Over the last few decades, the fluid-structure interaction (FSI) has gained increasing attention in numerical simulation because understanding the relationship between structures and fluid flows is crucial in many real-life applications. Recently, Microfluidics, which refers to different kind of microscale devices used for separating particles can also be simulated using FSI technique. However, current commercial simulation software is not able to effectively handle problems involving moving objects with large trajectories, i.e. the particles in microfluidic devices. The requirement of massive and complex computational mesh to capture the extremely long pathway of moving objects makes such problems very challenging to be solved. In this work, we aim to develop a robust solver that can effectively solve for FSI problems. The new solver, named IBMFoam, has been implemented in the OpenFOAM open source environment, based on the available solver pisoFOAM, using the Immersed Boundary Method (IBM) to simplify the meshing process and to avoid the need of re-meshing at every timestep when simulating moving objects. The solver is also capable of running in parallel computer system in case of dealing with a huge and complex problem. The new solver was validated with a series of well-documented cases. The results are shown to be in good agreement with available data from previous works, which proves the accuracy and efficiency of this new IBMFoam solver.

2. Numerical model

In this section, we briefly introduce the mathematical model for the fluid-structure interaction problem. The Immersed Boundary Method developed for solving the problem was published recently [1], hence the method will not be represented in this article.

Governing equations

We consider the interaction of objects in incompressible fluid flow. The phenomenon is governed by the Navier-Stokes equations, for fluid flow, and Newton equations for 6 DOF motion of mobile objects. The equations are given in the following form.

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u} + f_e \tag{2}$$

$$M \frac{d\mathbf{u}_\mathbf{p}}{dt} = \Delta M \mathbf{g} + \mathbf{F} + \mathbf{F}_i \tag{3}$$

$$\mathbf{I} \frac{d\boldsymbol{\omega}_p}{dt} = T \tag{4}$$

In above equations, $\mathrm{u} = \left(u_x, u_y, u_z\right)$ is fluid velocity vector, p is the static pressure, ν is the kinematic viscosity of the fluid, $\mathrm{u}_p = \left(u_{px}, u_{py}, u_{pz}\right)$ is the translational velocity of the object, $\boldsymbol{\omega}_p = (\omega_{px}, \omega_{py}, \omega_{pz})$ is angular velocity of the object. Other parameters including $M, \mathbf{g}, \mathbf{F}, \mathbf{F}_i, \mathbf{I}, T$ is the mass of the object, gravitational acceleration, the force acting on object's surface by the fluid enclosed in the object volume, collision force, the moment of inertia and torque about the center of mass of the object, respectively. In context of the IBM method applied in this study, the presence of structure in flow field is represented by an additional forcing term f_e in the equation (2), which indicates the mutual interaction

* Corresponding author: Tel.: (+84) 966.633.683
Email: sang.phamvan@hust.edu.vn

between fluid and immersed boundary. The calculating of f_e follows the steps discussed in [1].

Collision model

In the simulations concerning the interaction between many moving objects, a collision model is needed to prevent these bodies from interpenetration each other. In this work, the repulsive model proposed by Wan and Turek [4] has been used to compute the collision force act upon immersed objects during their movement.

Object – object collision: The repulsive force acting upon the i-th object caused by colliding with the j-th object is determined as following, where $R_i, R_j, \mathbf{X}_i, \mathbf{X}_j$ are the radius and the center of mass of the ith and the jth object respectively, $R_{ij} = R_i + R_j$ and $d = |\mathbf{X}_i - \mathbf{X}_j|$ is the distance between their centers of mass, ξ is the range of the repulsive force, ε_p and $\varepsilon_p^{'}$ are small positive stiffness parameters for the collision.

$$\mathbf{F}_{i,j}^{p} = \begin{cases} \frac{1}{\varepsilon_p^{'}}(\mathbf{X}_i - \mathbf{X}_j)(R_{ij} - d_{i,j}), \forall d_{i,j} \le R_{ij} \\ \frac{1}{\varepsilon_p}(\mathbf{X}_i - \mathbf{X}_j)(R_{ij} + \xi - d_{i,j})^2, \forall R_{ij} \le d_{i,j} \le R_{ij} + \xi \\ 0, \forall R_{ij} + \xi \le d_{i,j} \end{cases}$$

Object – wall collision: The repulsive force acting upon the ith object caused by colliding with rigid wall is:

$$\mathbf{F}_{i}^{W} = \begin{cases} \frac{1}{\varepsilon_W^{'}}(\mathbf{X}_i - \mathbf{X}_i^{'})(2R_i - d_i^{'}), \forall d_i^{'} < 2R_i \\ \frac{1}{\varepsilon_W}(\mathbf{X}_i - \mathbf{X}_i^{'})(2R_i + \xi - d_i^{'})^2, \forall 2R_i < d_i^{'} < 2R_i + \xi \\ 0, \forall 2R_i + \xi < d_i^{'} \end{cases}$$

where $\mathbf{X}_i^{'}$ is the coordinate of the nearest imaginary object, ε_w is a small positive stiffness parameter for object-wall collision, usually it can be taken as $\varepsilon_w = \varepsilon_p/2$, and $\varepsilon_w^{'} = \varepsilon_p^{'}/2$.

3. Implementation in OpenFOAM

Simulations of fluid-structure interaction (FSI) in OpenFOAM are normally not an easy task, especially when the moving body has complex geometry. The conventional approaches such as finite volume and finite element methods are computationally expensive in the FSI simulations where the mesh requires re-calculation at every time step. Immersed boundary method is a proper choice for the implementation of an FSI solver in the OpenFOAM environment due to its simple meshing process. In this work, the pisoFOAM, an available transient solver for incompressible fluids, has been chosen for modification since the PISO algorithm employed in pisoFOAM is very effective to solve the equations (1) and (2). The pisoFOAM has been modified by several extra steps (colored red in the Fig.1).

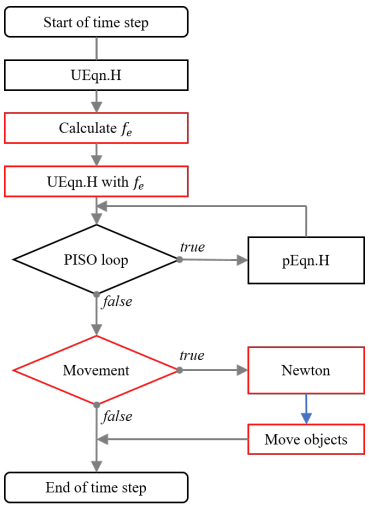

Fig. 1. Flow chart of the IBM - PISO algorithm.

A library named IBMlib has been created to automatically creating the Lagrange points, reading data from external mesh files, calculating f_e, moving objects and writing data for post-process. The core of the IBM solver is the `main` function consisting of the following lines of code:

```
#include "readGeometryInfor.H"
while (runTime.loop())
{
      #include "UEqn.H"
      #include "createForce.H"
      solve(UEqn==-fvc::grad(p)+fe);
      while(piso.correct())
      {
          #include "pEqn.H"
      }
      if(IBM.moving())
      {
          #include "moveObjects.H"
      }
}
```

The predicted velocity is obtained by solving the momentum equation, UEqn.H, which is expressed in OpenFOAM as the following:

```
volVectorField UEqn
{
    fvm::ddt(U)
+   fvm::div(phi, U)
-   fvm::laplacian(nu, U)
}
solve(UEqn == -fvc::grad(p) + pGrad)
```

where `phi` is the flux from previous time step and `nu` is the kinematic viscosity of the fluid. The dictionary `createForce.H` is included to calculate the body force term f_e by calling the function `calcForceEuler()` in the IMBlib library, based on the predicted velocity and desired velocity, while the desired velocity is set to a fixed value or is taken

from the solution of Newton equation. The force term will then be added to the RHS of the Navier-Stokes equation to resolve for velocity. The equations used in the pressure and velocity correction step are also modified accordingly as the body force is included. Afterward, the solver checks whether the problem involves object movement or not before calling the `moveObjects.H` dictionary, which solves for Newton equations to get translational and rotational velocity of structures to move them accordingly.

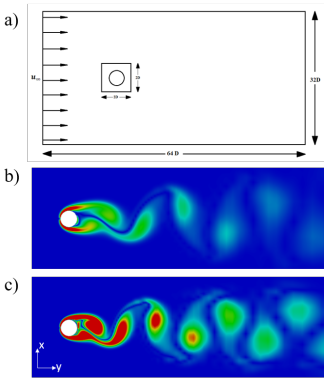

Fig. 2. Computational domain of the cylinder immersed in fluid flow (a) and Von-Karman vortex street at Re= 100 (b) and Re= 185 (c)

4. Results and discussion

4. 1. Validations

Flow over a cylinder

In this problem, we consider a fluid flow over a stationary circular cylinder. The computational domain is sketched in Fig.2. Boundary conditions for the problem are fixed velocity at the inlet, free stream pressure at the outlet and slip condition at the upper and lower wall. The cylinder is discretized into 314 Lagrange points evenly distributed on the surface. Uniform Cartesian mesh is used in the adjacency of the cylinder, i.e. in the region $-D \le x \le D$ and $-D \le y \le D$, outside this region, the mesh size is stretched.

Table 1. Mean drag coefficient and rms lift coefficient at Re= 185

	C_{Dmean}	C_{Lrms}
Constant *et al.* [3]	1.430	0.436
Pinelli *et al.* [4]	1. 387	0.428
Present	1. 430	0.458

The simulation is conducted at two different Reynolds number values: 100 and 185. At both regimes, the Von-Karman vortex shedding is observed as shown in Fig.2b,c. As the flow oscillation, the mean drag coefficient and the root-mean-square (RMS) lift coefficient are calculated and compared with available data from previous works of Pinelli et al. [2] and Constant et al. [3] as depicted in Table. 1. The results are shown to be in good agreement with the published data.

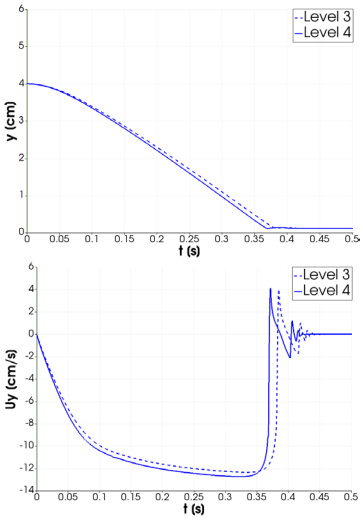

Fig. 3. Time history of the y-coordinate of the particle center (a), v-component of translational velocity (b) in 2 cases: $h = 1/48$ (Level3) and $h = 1/96$ (Level 4).

Sedimentation of a circular cylinder

In this validation, we consider the motion of a circular cylinder sedimenting in a domain of dimension $2 \times 6cm^2$ filling by fluid with density $\rho_f = 1g/cm^3$ and kinematic viscosity $\nu = 0.01cm^2/s$. The cylinder has diameter $d = 0.25cm$, density $\rho_p = 1.5g/cm^3$ and is located at the position $(1,4)cm$ at the beginning. Uniform Cartesian grid is used for the entire domain with different mesh sizes, i.e. $\Delta h = 1/48cm$, $\Delta h = 1/96cm$ and $\Delta h = 1/144cm$.

To validate the simulation result, we compare the maximum Reynolds number during the cylinder sedimentation with previous numerical results under the same condition. The maximal Reynolds number is defined as

$$\mathrm{Re}_{Max} = Max[\mathrm{Re}(t)] = Max[\frac{\rho_p d\sqrt{U_p(t)^2 + V_p(t)^2}}{\mu}]$$

where $\mathbf{U}(t) = (U_p(t), V_p(t))$ is the velocity of the mass center of the cylinder at time t. As can be seen in Table 2, the results of present study are in rough agreement with those of previous works. In addition to the comparison of the maximum Reynolds number, an examination of some other quantities is presented in Fig.3, including time histories of the y-coordinate of the cylinder center, the vertical component of translational velocity of the mass center of the cylinder.

Table 2. Comparison of the maximum Reynolds number during the cylinder sedimentation

	Present		Wang(2008)		Wan,Turek(2005)		Glowinski(2001)		Unlmann(2005)
$\Delta h(cm)$	$\frac{1}{96}$	$\frac{1}{144}$	$\frac{1}{72}$	$\frac{1}{144}$	$\frac{1}{48}$	$\frac{1}{96}$	$\frac{1}{192}$	$\frac{1}{256}$	$\frac{1}{256}$
Re_{Max}	477.75	484.38	502.37	503.26	442.19	465.52	438.6	450.7	495

When the cylinder falls on the bottom of the channel, it suffers from a colliding force and rebounds back. The process of falling and rebound back is repeated as can be seen from the Fig.3b, the sign of the vertical velocity changes alternately along with the decrease in the velocity magnitude, finally the value of vertical velocity reach zero and the cylinder stays on the bottom of the channel

4.2. Applications

Sedimentation of many particles

To examine the capability of the current solver to simulate problems involving motion of many bodies simultaneously, the sedimentation of one hundred and five particles in a closed rectangular domain of $6\times 6cm^2$ is carried out. Each particle has diameter of $d = 0.25cm$,density $\rho_p = 1.5g/cm^3$ and is discretized into 63 points evenly distributed on the surface. The fluid has density: $\rho_f = 1g/cm^3$ and kinematic viscosity $\nu = 0.01cm^2/s$. Uniform Cartesian grid is used for then tire domain, the mesh size is $\Delta h = 1/80\,cm$ results in a total of 2304000 elements. The current solver is programmed to be able to compute in parallel. In this simulation, the mesh is divided into 16 parts managed by corresponding 16 single processors. The average elapsed time for one-time step is about 2.8 second and with the chosen time step $\Delta t = 10^{-5}s$, it takes nearly 148 hours to finish the simulation of 2 seconds long sedimentation.

The temporal evolution of the particle positions within the domain is demonstrated in the Fig.4. The particles are initially arranged into 5 rows and 21 columns, located at the top region of the computational domain, then they fall under the influence of gravity. It is witnessed that during the sedimentation, the particles near two side walls fall quickly while those in the middle of the domain are held by the fluid, this would result in a space area enclosed by particles around. In the end, all the particles settle down and stay on the bottom of the domain. It is worth noticing that, during the whole process, there is no pair of particles having their boundary penetrate each other or into walls, which shows the reliability of current collision model. The success of capturing complex behavior of many bodies in this simulation proves the capability of the IBMFoam solver in applying to other particulate flows and related problems.

Simulation of a vortex-aided sorting device

In this simulation, we consider a design of microfluidic device proposed by X. Wang et al [6]. The original design consists of four major components: a high-aspect-ratio channel for inertial particle ordering; two symmetric chambers for micro vortex formation; two side outlets at the corners of the chambers for extraction of large particles and a main outlet for exit of small particles as illustrated in Fig.5a. The computational domain is sketched in Fig.5b. To reduce the size of computational mesh, the focusing channel will not be modeled and the data of particle sizes and their corresponding focusing positions at the end of the channel will be used as the inputs for current simulation instead. In addition, only one of the two chambers is considered.

The equilibrium positions of moving particles within a channel are the result of the balance between two main forces: the wall-induced lift force $\mathbf{F}_w$ and the shear-gradient induced lift force $\mathbf{F}_s$. When a balance particle moving across the chamber, the formation of the vortex inside the chamber reduces the magnitude of $\mathbf{F}_w$, leading to the particle lateral migration undergoing the $\mathbf{F}_s$. Since the magnitude of the shear-gradient lift force scales as $F_s \propto a^2$, a is the particle's diameter, the larger particles migrate faster across the boundary streamline of main flow thus being captured and isolated from the main flow.

In this simulation, we investigate the behavior of a particle of diameter $d = 0.5mm$ at two different Reynolds numbers. Fig.6 demonstrates the temporal position of the particle at $\mathrm{Re} = 4$ and $\mathrm{Re} = 40$. As can be seen, the Reynolds number is directly related to the size of the vortex formed inside the chamber and consequently affects the trajectory of the particle. At low Reynolds number, the shear-gradient lift force is not enough to push the particle into the chamber, thus the particle remains in the main flow and exit at the main outlet. At $\mathrm{Re} = 40$, the shear-gradient lift force is significant and lateral migration of the particle happens very quickly, leading to the isolation of the particle from the main flow. This size-selectivity phenomenon is like that mentioned in published result of X. Wang et al [6].

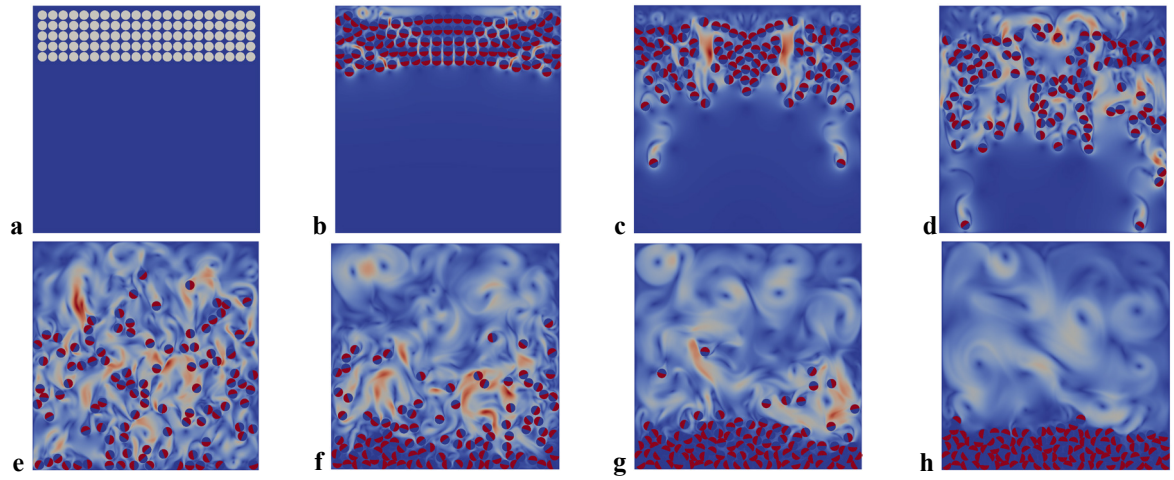

Fig. 4. Temporal motions of 105 particles, images from A to I correspond to the instantaneous position of particles at time $t = 0s$ (a), $0.25s$ (b), $0.5s$ (c), $0.65s$ (d), $0.75s$ (e), $1.25s$ (f), $1.5s$ (g) and $2s$ (h).

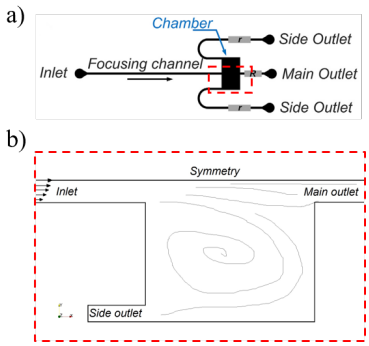

Fig. 5. (a) Schematic of the vortex-aided inertial microfluidic design. (b) The computational.

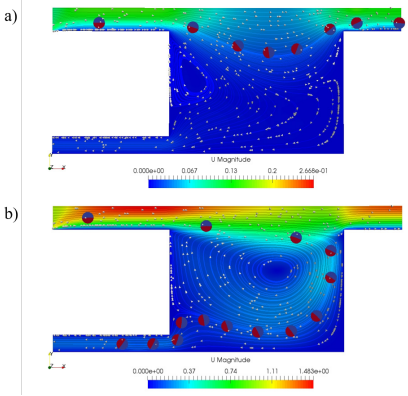

Fig. 6. Particle trajectory at Re = 4 (a) and 40 (b)

5. Conclusion

In this work, a new solver combining the IBM and the PISO algorithm has been successfully implemented in OpenFOAM environment. The solver was validated via two bench-marked problems: flow over a cylinder and sedimentation of a cylinder. The obtained results are in good agreement with the previous study. The solver can compute in parallel, which gives the possibility to solve for big and complex simulation problems such as the sedimentation of hundreds of particles. The new solver is also shown great potential in applications to simulations of microfluidic devices, based on the promising results for the vortex-aided sorting device.

Acknowledgments

This research is funded by the Hanoi University of Science and Technology under project number T2016-PC-026

References

[1] V. S. Pham; An Immersed boundary method for simulation of moving object in Fluid flow; Journal of Science and Technology Technical Universities, Vol. 127, 040-044, 2018.

[2] A. Pinelli, I. Naquavi, U. Piomelli et al; Immersed-boundary methods for general finite-difference and finite-volume Navier-Stokes solvers; Journal of Computational Physics, 2010.

[3] E. Constant, C. Li, J. Favier, M. Meldi, P. Meliga, E. Serre; Implementation of a discrete Immersed Boundary Method in OpenFOAM; Journal of Computer & Fluid, 2016.

[4] D. Wan, S. Turek; An efficient multigrid-FEM method for the simulation of solid-liquid twophase flows; Journal of Computational and Applied Mathematics, 2005.

[5] Z. Wang et al; Combined multi-direct forcing and immersed boundary method for simulating flows with moving particles; International Journal of Multiphase Flow 34, 2008, 283-302.

[6] X. Wang, J. Zhou, I. Papautsky; Vortex-aided inertial microfluidic device for continuous particle separation with high size-selectivity, efficiency and purity, Biomicrofluidics 7, 044119,2013