A 3D unified model to Fluid-Structure Interaction with Block Based Adaptive Mesh Refinement

Thomas Altazin, Frederic Golay, Philippe Fraunié

A 3D unified model to fluid-structure interaction with block-based adaptive mesh refinement

T. Altazin [1,2], **F. Golay** [1], **P. Fraunié** [2]

[1] **Université de Toulon, IMATH, EA 2134, 83957 La Garde, France**

[2] **Université de Toulon, Aix-Marseille Université, CNRS/INSU, IRD, MIO, UM 110, 13288, Marseille, Cedex 09, France.**

Abstract

We propose to bring a better understanding of the interaction between an air-water flow and a floating structure by means of a fast accurate numerical model. Following the 3D low Mach compressible Euler model developed in [14] and previous works from [20], [11] and [13], we are now interested in simulating the motion of a floating structure in an air-water flow. In the context of a fictitious domain, a volumic penalization is applied inside the body to ensure a rigidity constraint through a penalized velocity in order to get the correct motion of the rigid body. The tracking of the solid is insured by the reconstruction of a Heaviside function thanks to a ray-casting algorithm. The validity of our fluid-structure interaction (FSI) procedure is investigated through some examples.

keywords: Euler equations, low Mach, compressible, bi-fluid, mesh refinement, fluid-structure interaction, fictitious domain, penalization, projection method, ray-casting, interface-sharpening.

1 Introduction

Fluid-structure interaction (FSI) still represents a challenge in understanding the complex behaviour of a floating structure subjected to a swell or a complex flow , especially in coastal engineering. The model validated in [14] to simulate such phenomenon is a low Mach compressible model which solves Euler equations in two phases (air/water). The viscous term in the Navier-Stokes equation is indeed neglected given that this study focusses on simulations where the viscosity should not have a dominant influence. The model is refered as an artificial compressibility model since it uses an unphysical speed of sound in water. This allows a less restrictive time-step for faster simulations. In order to get both fast and accurate results, an adaptive mesh refinement has been designed to refine area of interest, especially around a structure embedded in the flow which is the main concern in this work.

FSI problems are encountered in a wide range of problems such as human walking, insect flight or renewable energy [23]. Various methods are available to take into account the presence of an obstacle in a flow. Mittal and Iaccarino [16] give a relevant detailed review of such methods. The first common idea is to use a mesh that fits the boundary of the solid (ie. body-fitted mesh). Thus, nodes are directly located on the interface allowing good accuracy but such meshes can be very tedious to create, especially for complex shapes. A good alternative is the fictitious domain method [12] which allows very simple meshes that don't conform to the geometry of the rigid body. However, special procedures are required to impose correct boundary conditions at the cells in the neighborhood of the interface. The immersed boundary method [17] introduced the idea of modifying the momentum equation to treat the behaviour of a blood flow inside the heart. A lot of studies use derived form of the immersed boundary method to deal with fluid-structure interaction such as the curvilinear immersed boundary (CURVIB) from [4] and [5] or an immersed boundary method with fraction volume smoothing [3]. An other common method to deal with FSI problems is the penalization method. Again, several formulations have been suggested for FSI with various applications : a viscosity penalization [9], a tensorial penalization [19], a sub-mesh penalty method [21],a volume penalization with a Fourier pseudo-spectral discretization [15], [10] or a vortex level method with penalized colliding bodies [7]. For its simplicity, we choose to implement a penalization method.

2 Two-fluid low Mach compressible Euler model

2.1 Governing equations

Let's consider an inviscid isothermal compressible flow ruled by Euler equations with a mixture of two phases, air and water. The mass and momentum conservation equations give :

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{1}$$

and

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + p\bar{\bar{\boldsymbol{I}}}) = \rho \mathbf{g} \ , \tag{2}$$

where ρ denotes the density, p the pressure and $\mathbf{u}$ the velocity. The volume fraction ϕ, sometimes called "color function", is introduced to manage two fluid flows with different density, especially air-water flows. This variable follows an advection equation written in non-conservative form in order to avoid spurious oscillations at the interface air-water [14] :

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0 \ . \tag{3}$$

Equations (1), (2) and (3) form a hyperbolic system from which useful theoretical results can be derived. The system is closed with an artificial pressure law, depending on ϕ, under the assumption of low Mach number ($Ma < 0.3$),

$$p = p_0 + c_0^2 (\rho - (\phi \rho_A + (1 - \phi)\rho_W) \ , \tag{4}$$

where p_0 is the reference pressure, c_0 speed of sound in the mixture and ρ_A and ρ_W the density of air and water respectively. The speed of sound is taken much below the physical expected value, on purpose, to avoid a too restrictive time step due to the explicit scheme in time.

2.2 Numerical implementation

The numerical resolution of the hyperbolic system is conducted by using a Godunov-type finite volume scheme. The time discretization is explicit and of order 2 with Runge-Kutta or Adams-Bashforth integration. The spatial discretization uses the MUSCL reconstruction scheme with Barth limiter to get order 2 as well. The equation (3) is known to be very diffusive. Following [22] , a mass-conserving interface-sharpening procedure has been applied to counteract the effect of the numerical diffusion, which lead us to solve a corrective equation on ϕ by means of a splitting procedure :

$$\frac{\partial \phi}{\partial \tau} = \phi^2 (1 - \phi)^2 (\phi - c) \ , \tag{5}$$

where c is computed numerically to insure mass conservation.
Finally, the code benefits from the MPI library and is parallelized by using domain decomposition.

2.3 Mesh refinement strategy

A dynamic block based adaptive mesh refinement method (BB-AMR) with a local multi-time step algorithm has been developed to balance fast computations and accurate results. The computational domain is divided in multiple blocks and each of them can be meshed according to its refinement level n. Be nx, ny and nz the number of cells in each direction, the number of cells of each block can be computed by :

$$n_{cells} = 8^n \ nx \ ny \ nz$$

For all hyperbolic systems of conservation laws, as for the one we use here, a mathematical entropy can be derived, veryfing the Lax's entropy condition :

$$\frac{\partial s(\mathbf{w})}{\partial t} + \frac{\partial \psi(\mathbf{w})}{\partial x} \leq 0 \ , \tag{6}$$

where the entropy flux $\psi(\mathbf{w})$ satisfy $\nabla_{\mathbf{w}}^T \psi = \nabla_{\mathbf{w}}^T s(\mathbf{w}) f'(\mathbf{w})$. In the present case, for the two-fluid model, the expression of entropy and entropy flux can be :

$$s = \frac{1}{2}\mathbf{u}^2 + c_0^2 \rho ln\rho - c_0^2(\rho_W - \rho_A)\phi \ ,$$

$$\psi = (\frac{1}{2}\rho\mathbf{u}^2 + c_0^2\rho(ln\rho + 1))\mathbf{u} \ .$$

Equation (6) is a conservative expression that can be integrated into the system of conservation laws. Thus it uses the same discretization in space and time from which a numerical production of entropy can be derived. This quantity is used as a refinement criterion for each block dividing the computational domain. Locally, the block is coarser or finer according to the value of its numerical production of entropy relative to the numerical production of entropy of the whole domain. More details can be found in [11] and [13]. The interest of such a criterion is that the entropy acts both as an indicator error as well as a indicator of locations where shocks occur (ie. areas that need to be refined) as mentionned in [18] and [8].

3 Fluid-structure interaction

An improvement of the model described above is suggested here to treat fluid-structure interaction (FSI) problems by introducing a moving obstacle in air-water flows. Among all methods available to take into account an obstacle in a flow, we seek the one which allows us both an easy and time-saving procedure. Following [7], we choose a monolithic approach and consider the fluid-solid system as unified.

3.1 Penalization method

The penalization method leans on a physical basis as it was first introduced in [2] for convection problems in porous media. The idea is to treat the fluid and the solid as an unique fluid system in which the porosity varies. Mathematically, a source term is added in the momentum equation such as:

$$\rho\left(\frac{\partial\mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) = -\nabla p + \rho\mathbf{g} + \frac{\mu}{K}\mathbf{u} \ , \tag{7}$$

where μ is the kinematic viscosity and K the porosity. On one hand, when K tends to infinity, this additional term vanishes and it remains the classical Euler equations, which means that the fluid is utterly permeable. On the other hand, when K tends to zero, this additional term becomes dominant and it remains $\mathbf{u} = \vec{0}$ which means that the fluid is completely impermeable and doesn't move, it can be seen as a solid. Thus, depending on the value of K, this monolithic formulation allows us to treat fluid and solid at the same time.
One should notice that this formulation only allows the introduction of a **fixed** obstacle (u=0). To allow solid motion, we use a slightly different method called volumic penalization [1]. We want the fluid part and the solid part to solve the following equations :

$$\rho\left(\frac{\partial\mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) = -\nabla p + \rho\mathbf{g} \quad \text{in fluid} \ ,$$

$$\mathbf{u} = \overline{\mathbf{u}} \quad \text{in solid} \ .$$

where $\overline{\mathbf{u}}$ is the velocity we want to apply in the solid and will be defined below. These two equations can be rewritten in the following form by introducing the Heaviside function χ and the penalization parameter λ:

$$\rho\left(\frac{\partial\mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) = -\nabla p + \rho\mathbf{g} + \chi\lambda(\mathbf{u} - \overline{\mathbf{u}}) \ . \tag{8}$$

χ equals 1 inside the solid and 0 otherwise. In fluid, the last term of equation 8 vanishes and one recovers Euler equations. Inside the solid, the last term of the equation 8 exists and its contribution

is weighted by λ. λ is chosen very large so that $\lambda(\mathbf{u} - \overline{\mathbf{u}})$ becomes dominant and it remains $\mathbf{u} = \overline{\mathbf{u}}$. Contrary to equation (7), this expression allows to introduce **moving** obstacle (with velocity $\overline{\mathbf{u}}$ which will be calculated accordingly below). Mathematically, we can rely on the convergence results by Angot [1].

3.2   Projection method

Numerically, the implementation of the penalization seems simple as only an additional source term has to be added. One needs to be careful about the value of the penalization parameter since it must be chosen relative to the time step in order to keep stability because of the explicit time scheme.

However, as pointed out in [7], the fluid structure problem can be solved by solving successively the Euler equations and a "correction" equation through time-splitting:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \mathbf{g}$$

$$\text{with} \quad \mathbf{u}(0) = \mathbf{u}_0 \ ,$$

then

$$\frac{\partial \mathbf{u}^*}{\partial \tau} = \chi\lambda(\Pi(\mathbf{u}^*) - \mathbf{u}^*) \ , \tag{9}$$

$$\text{with} \quad \mathbf{u}^*(0) = \mathbf{u}(\delta t) \ ,$$

where $\Pi(\mathbf{u}^*)$ is the projector of the velocity field onto a "rigid" velocity field. With linear algebra, we can prove by choosing appropriate vectorial spaces and by defining adequate projectors that equation (9) after one or two iterations is equivalent to :

$$\mathbf{u}^{n+1} = \Pi(\mathbf{u}^*) = \overline{\mathbf{u}} \ \text{ in solid} \ ,$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* \ \text{ in fluid} \ .$$

This can be seen as a prediction-correction method, very similar to Chorin's [6] for incompressibility constraint. We use this analogy to apply a rigidity constraint inside the solid.

3.3   Penalization velocity

The equation (8) can be applied as it is, if the penalization velocity is known which is not the case because the velocity of the body will depend on the velocity of the flow around it. To find the penalization velocity relative to the velocity of the fluid, the latter is projected onto a "rigid" velocity field, that is a velocity that verifies :

$$\overline{\mathbf{u}} = \mathbf{V_G} + \boldsymbol{\omega} \wedge \mathbf{GM} \ , \tag{10}$$

where $\mathbf{V_G}$ is the translation velocity, $\boldsymbol{\omega}$ the angular velocity and $\mathbf{GM}$ the distance between an arbitrary point M in the solid and G, its center of gravity.

Minimizing the following functional over the solid domain Ω:

$$\min J = \int_{\Omega} (\mathbf{u} - \overline{\mathbf{u}})^2 \ \mathrm{d}\Omega \ ,$$

we get the value of the velocity to be applied inside the solid :

$$\overline{\mathbf{u}} = \frac{1}{|M|} \int_{\Omega} \rho\mathbf{u} \ \mathrm{d}\Omega + \left(\overline{\overline{\mathbf{J}}}^{-1} \int_{\Omega} \mathbf{GM} \wedge \rho\mathbf{u} \ \mathrm{d}\Omega \right) \wedge \mathbf{GM} \ , \tag{11}$$

where $\overline{\overline{\mathbf{J}}}^{-1}$ is the inverse of the inertia matrix of the solid and M its mass.

3.4 Motion of the structure

The correct velocity to be applied inside the body has been computed to simulate a true rigid solid motion. Now, we actually need to move the body onto the computational grid. Several authors used to define the solid boundary by a level-set function which is convected in an eulerian formalism. A drawback is that the level-set function solves the advection equation (3), which is very diffusive. Thus the boundaries may deform through time and the structure's shape does not remain constant. In fact, because of the rigid hypothesis about the structure, we know exactly the position of the solid-fluid boundary.

Let's define a set of markers located on the boundary of the solid, we thus define a surface mesh. Each of these markers can be advected by the equation of rigid motion (10). We need then to retrieve which cells in the computational grid belong to the solid, that is reconstruct the indicator function, or Heaviside function. This can be done by a ray-casting algorithm, as it was previously done for instance in [4]. In short, by counting the number of intersections (even or odd) between a ray and the faces that define the geometry of the solid, we are able to determine which cell is inside (odd) or outside (even) the solid. These results come from the Jordan-curve theorem. This algorithm allows us to include any arbitrary solid with complex shapes in two or three dimensions.
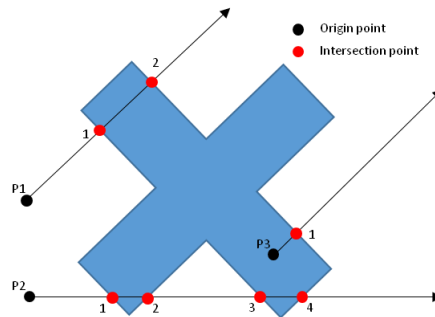

Figure 1: Raycasting algorithm : P1 and P2 are outside the solid because the number of intersections with the solid's contour is even whereas P3 is inside the solid because its number of intersections is odd.

4 Validation

4.1 Water entry of a circular cylinder

We consider the water entry of a circular cylinder (density 0.5, diameter d = 1) in a two-phases flow air/water. The computational domain (10d x 10d) is divided in 400 blocks with nx=ny=nz=5 and the maximum level of refinement is 3. One counts 188 000 cells with a minimum resolution of 0.1d and a maximum resolution of 0.0125d. The boundary conditions are such that the fluid is free to leave the computational domain except for the bottom where a mirror condition is applied. The calculation run for 7 hours on 20 Xeon Phi processors for a real time simulation of 2 seconds. The body starts from its initial position in the air at rest. Its motion is only derived from the penalization velocity.
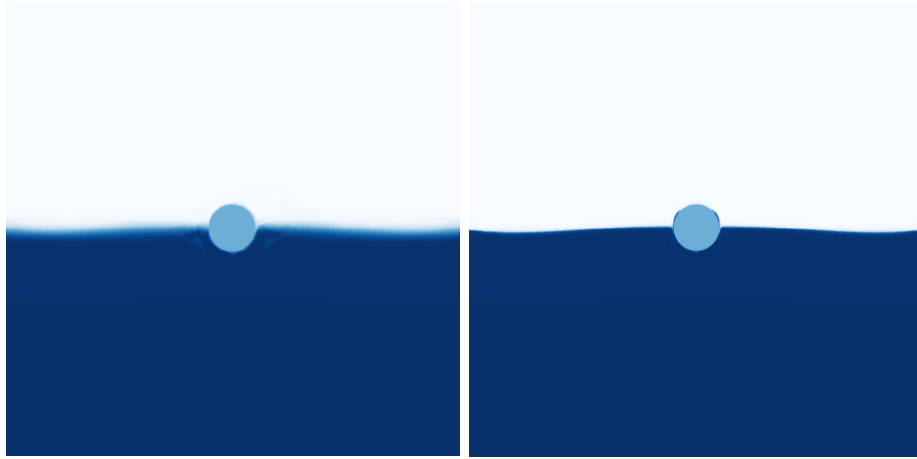
Figure 2: Interface at t=1.5s without (left) and with (right) interface sharpening.

We can notice on the left figure that the interface between air and water becomes blurred. Because of the interface sharpening procedure, we are able to control the diffusion such as the interface's thickness is lower than 8 cells. Without sharpening, the interface spreads over 30 cells.
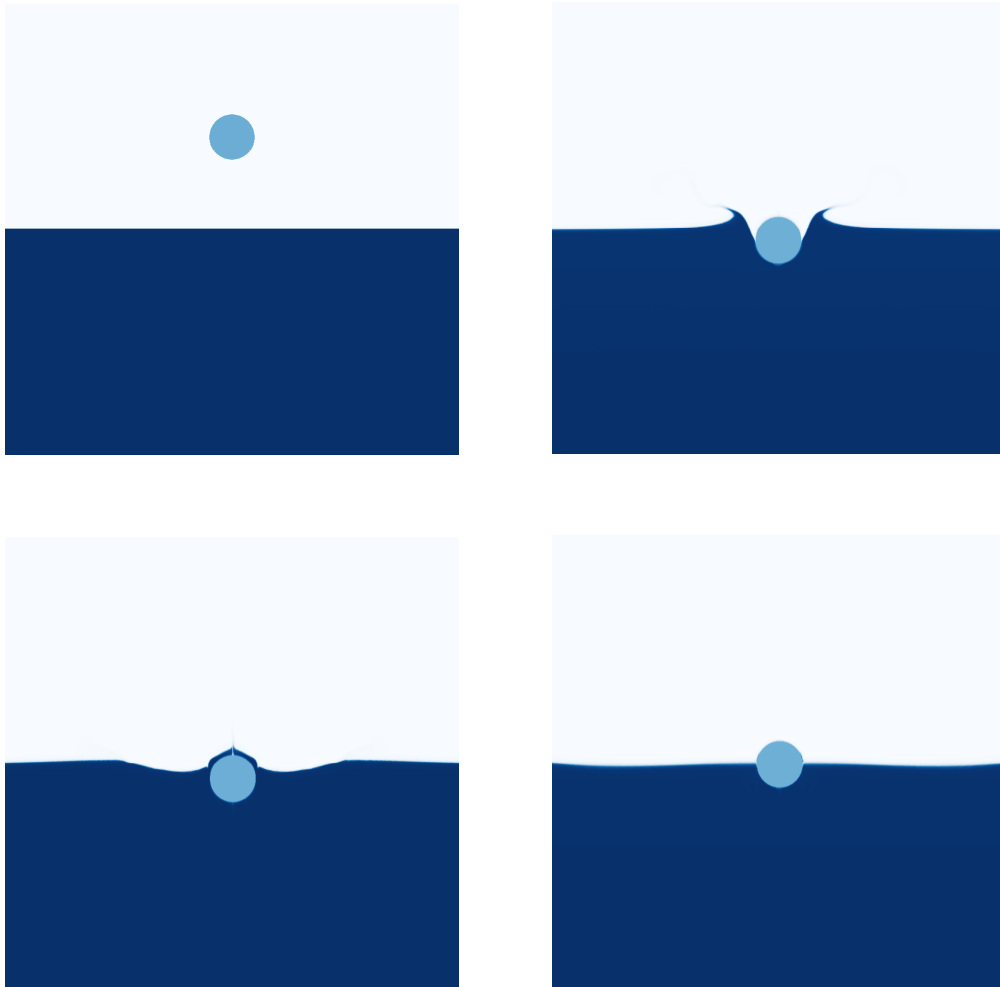

Figure 3: Position at t=0s, t=0.25s, t=0.5s and t=2s

After 2 seconds (see figure 3), a stationnary state is reached and we can notice that half of the body is immersed under the surface as expected according to the density ratio ($\frac{\rho_S}{\rho_W} = 0.5$). We also see a satisfactory thickness of the interface because of the sharpening procedure.

5 Conclusion

The aim of this study was to propose a fast and reliable procedure to treat fluid-structure interaction from an existing model. By considering a fictitious domain approch, we have developed a volumic penalization based on a prediction-correction method. The body's motion is imposed by a penalization velocity which is computed according to the flow's dynamic. A raycasting algorithm allows us to track the position of the fluid-solid boundary in the domain. Primary tests indicate a good physical behaviour of the solid. Further validation tests will be necessary to assess the relevance of our model and experimental confrontations are in progress. Extension to 3D cases will be carried out soon and the fluid-structure solver will greatly benefit from BB-AMR procedure for such applications.

Acknowledgments

References

[1] P. ANGOT, C.-H. BRUNEAU, AND P. FABRIE, *A penalization method to take into account obstacles in incompressible viscous flows*, Numerische Mathematik, 81 (1999), pp. 497–520.

[2] E. ARQUIS AND J. CALTAGIRONE, *Sur les conditions hydrodynamiques au voisinage dune interface milieu fluide-milieu poreux: application à la convection naturelle*, CR Acad. Sci. Paris II, 299 (1984), pp. 1–4.

[3] B. BIGOT, T. BONOMETTI, L. LACAZE, AND O. THUAL, *A simple immersed-boundary method for solid–fluid interaction in constant-and stratified-density flows*, Computers & Fluids, 97 (2014), pp. 126–142.

[4] I. BORAZJANI, L. GE, AND F. SOTIROPOULOS, *Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3d rigid bodies*, J. Comp. Phys., 227 (2008), pp. 7587–7620.

[5] A. CALDERER, S. KANG, AND F. SOTIROPOULOS, *Level set immersed boundary method for coupled simulation of air/water interaction with complex floating structures*, J. Comp. Phys., 277 (2014), pp. 201–227.

[6] A. J. CHORIN, *Numerical solution of the navier-stokes equations*, Mathematics of computation, 22 (1968), pp. 745–762.

[7] M. COQUERELLE AND G.-H. COTTET, *A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies*, J. Comp. Phys., 227 (2008), pp. 9121–9137.

[8] J.-P. CROISILLE, *Contribution à l'étude théorique et à l'approximation par éléments finis du système hyperbolique de la dynamique des gaz multidimensionnelle et multiespèces*, PhD thesis, Paris 6, 1990.

[9] B. DUCASSOU AND S. ABADIE, *Méthode des domaines fictifs appliquée à l'interaction d'un solide mobile avec un écoulement à surface libre en vue de l'application aux systèmes houlomoteur*, in Rencontres Universitaires de Génie Civil, Bayonne, France, May 2015.

[10] T. ENGELS, D. KOLOMENSKIY, K. SCHNEIDER, AND J. SESTERHENN, *Numerical simulation of fluid–structure interaction with the volume penalization method*, J. Comp. Phys., 281 (2015), pp. 96–115.

[11] M. ERSOY, F. GOLAY, AND L. YUSHCHENKO, *Adaptive multiscale scheme based on numerical density of entropy production for conservation laws*, Open Mathematics, 11 (2013), pp. 1392–1415.

[12] R. GLOWINSKI, T.-W. PAN, AND J. PERIAUX, *A fictitious domain method for external incompressible viscous flow modeled by navier-stokes equations*, Computer Methods in Applied Mechanics and Engineering, 112 (1994), pp. 133–148.

[13] F. GOLAY, M. ERSOY, L. YUSHCHENKO, AND D. SOUS, *Block-based adaptive mesh refinement scheme using numerical density of entropy production for three-dimensional two-fluid flows*, International Journal of Computational Fluid Dynamics, 29 (2015), pp. 67–81.

[14] F. GOLAY AND P. HELLUY, *Numerical schemes for low mach wave breaking*, International Journal of Computational Fluid Dynamics, 21 (2007), pp. 69–86.

[15] B. KADOCH, D. KOLOMENSKIY, P. ANGOT, AND K. SCHNEIDER, *A volume penalization method for incompressible flows and scalar advection–diffusion with moving obstacles*, J. Comp. Phys., 231 (2012), pp. 4365–4383.

[16] R. MITTAL AND G. IACCARINO, *Immersed boundary methods*, Annu. Rev. Fluid Mech., 37 (2005), pp. 239–261.

[17] C. S. PESKIN, *Flow patterns around heart valves: a numerical method*, J. Comp. Phys., 10 (1972), pp. 252–271.

[18] G. PUPPO, *Numerical entropy production on shocks and smooth transitions*, Journal of scientific computing, 17 (2002), pp. 263–271.

[19] T. RANDRIANARIVELO, G. PIANET, S. VINCENT, AND J. CALTAGIRONE, *Numerical modelling of solid particle motion using a new penalty method*, International Journal for Numerical Methods in Fluids, 47 (2005), pp. 1245–1251.

[20] A. N. SAMBE, D. SOUS, F. GOLAY, P. FRAUNIÉ, AND R. MARCER, *Numerical wave breaking with macro-roughness*, European Journal of Mechanics-B/Fluids, 30 (2011), pp. 577–588.

[21] A. SARTHOU, S. VINCENT, P. ANGOT, AND J.-P. CALTAGIRONE, *The sub-mesh penalty method*, in 5th International Symposium on Finite Volumes for Complex Applications, 2009, p. 633.

[22] K.-M. SHYUE, *An eulerian interface-sharpening algorithm for compressible gas dynamics*, in Modeling, Simulation and Optimization of Complex Processes-HPSC 2012, Springer, 2014, pp. 221–231.

[23] F. SOTIROPOULOS AND X. YANG, *Immersed boundary methods for simulating fluid–structure interaction*, Progress in Aerospace Sciences, 65 (2014), pp. 1–21.