

AN IMMERSED BOUNDARY APPROACH WITH MULTIGRID ACCELERATION FOR HIGH ORDER DISCONTINUOUS GALERKIN SOLVERS

Stefano Colombo^{1,2*}, Esteban Ferrer^{1,2}, Jiaqing Kou^{1,2} and Eusebio Valero^{1,2}

¹ ETSIAE-UPM - School of Aeronautics, Universidad Politécnica de Madrid. Plaza Cardenal Cisneros 3, E-28040 Madrid, Spain

² Center for Computational Simulation, Universidad Politécnica de Madrid, Campus de Montegancedo, Boadilla del Monte, 28660, Madrid, Spain

* email: stefano.colombo@upm.es

Key words: Immersed boundaries, volume penalisation, p-multigrid, high order discontinuous Galerkin

Abstract. This work describes the implementation of the Immersed Boundary Method (IBM) in a high order discontinuous Galerkin framework for the CFD solver Horses3D [1]. High order schemes are very attractive due to their low numerical dissipation, their capability of providing high-accurate solutions and higher efficiency for a given level of accuracy with respect to low order schemes. However, the generation of high order meshes needed by these schemes is still a bottleneck since it requires a large amount of time. IBM tries to tackle the problem by preserving the high order beneficial properties while avoiding the generation of complex meshes.

1 The Immersed Boundary Method

The idea behind the Immersed Boundary Method (IBM) [2] is to simulate the presence of a body by modifying the equations to be solved. This approach allows to use very simple cartesian meshes. This consideration has attracted a lot of interest in the IBM which has been widely analysed and studied (at least for low order schemes). The effort made by the CFD community has led to different approaches to mimic the body inside the fluid domain like cut-cell [3], ghost nodes[4, 5], direct forcing [6] and volume penalization[10, 11, 9] . The last three types belong to an IBM category where the mesh is fixed and the body is simulated trough the addition of a source term to the equations. In this work, the volume penalization has been chosen due to its simplicity, robustness and because it can be easily extended to moving bodies. From a physical point of view, in the IBM with volume penalization the body is considered as a porous media whose permeability approaches zero. To properly capture the shape of the body the mesh must be refined close to the boundaries which can be done by increasing the the polynomial order near the body. In this work we accelerate the IBM method using a multigrid Full Approximation Scheme (FAS) [12].

1.1 Volume penalization

In the volume penalization approach a source term is applied to the compressible Navier-Stokes equations. A mask function $\chi(\mathbf{x}, t)$ is required to be able to distinguish between the solid region (Ω_s), in which the source term is applied, and the fluid region (Ω_f) where no source is added. The equations become:

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot (\mathbf{F}_{inv} + \mathbf{F}_{vsc}) = \mathbf{S} \quad (1)$$

where \mathbf{Q} is the state defined as $(\rho, \rho u, \rho v, \rho w, E)$ with $E = \frac{P}{\gamma-1} + \frac{1}{2}\rho(u^2 + v^2 + w^2)$ and \mathbf{F}_{inv} , \mathbf{F}_{vsc} are the inviscid and viscous fluxes respectively:

$$\begin{aligned} \mathbf{F}_{x,inv} &= \begin{pmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ \rho uw \\ u(E + P) \end{pmatrix}, \mathbf{F}_{x,vsc} = - \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + q_x \end{pmatrix}, \\ \mathbf{F}_{y,inv} &= \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + P \\ \rho vw \\ v(E + P) \end{pmatrix}, \mathbf{F}_{y,vsc} = - \begin{pmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + q_y \end{pmatrix}, \\ \mathbf{F}_{z,inv} &= \begin{pmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + P \\ w(E + P) \end{pmatrix}, \mathbf{F}_{z,vsc} = - \begin{pmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + q_z \end{pmatrix}. \end{aligned}$$

In the previous equations, the viscous shear stresses are defined as $\tau_{ij} = \mu(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}(\frac{\partial u_k}{\partial x_k}))$ where μ denotes the dynamic viscosity. The heat flux vector $\nabla \mathbf{q}$ is defined as $\frac{\partial \mathbf{q}}{\partial x_i} = \lambda \frac{\partial T}{\partial x_i}$ where λ is the thermal conductivity and T is the static temperature. In eq. 1 the source term is:

$$\mathbf{S} = -\frac{\chi}{\eta} \begin{pmatrix} 0 \\ \rho u - \rho_s u_s \\ \rho v - \rho_s v_s \\ \rho w - \rho_s w_s \\ \rho(u^2 + v^2 + w^2) - \rho_s(u_s^2 + v_s^2 + w_s^2) \end{pmatrix}, \quad (2)$$

where χ is the mask, η is the penalization parameter and the subscript $(\cdot)_s$ stands for the imposed state. In the case of compressible Navier-Stokes equations and solid static bodies (defined through the IBM), the condition $\mathbf{u}_s = (0, 0, 0)$ is applied. The penalization term is set equal to the time step Δt [8] which generates a stiff source term. To avoid the time step limitations due to the presence of the stiff source term, a second order Strang splitting is used to include the source term, allowing the treatment of this source term implicitly[11].

1.2 Mask generation

The generation of the mask is a key point in the IBM. This mask finds the points (Gauss point in high order Discontinuous Galerkin) that are inside and outside the body. The points inside are penalised using the volume penalisation technique. The mask generation can be a costly process in IBM and for this reason we use a robust and efficient algorithm. A ray-tracing technique along with a kd-tree is used. The body is represented by a STL, file which is a three dimensional surface of triangles. The ray-tracing is based on the fact that an odd number of intersections identifies a point inside the body, i.e. $\chi = 1$, while an even number of intersections identifies a point outside the body i.e. $\chi = 0$ (see fig.1). This is sufficient for the definition of $\chi(\mathbf{x}, t)$ on each degree of freedom where the solution is computed. This technique can be

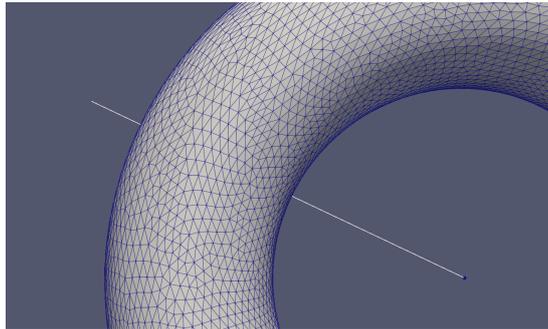


Figure 1: The idea behind the ray-tracing technique: in this picture the number of intersections between the ray and the STL is even, thus the point is outside the body.

further accelerated and automated as follows: generating the minimum bounding parallelepiped enclosing a set of points, also known as Oriented Bounding Box (OBB), embedding the STL file, check if a point is inside or outside the OBB (computationally cheap operation), if the point is inside the OBB a ray-tracing is performed. In this way the number of degrees of freedom is highly reduced. The latter procedure is accelerated thank to the use of a Surface Area Heuristic kd-tree embedding the STL file, which is known to be the best algorithm for ray-tracing [13], [14].

1.3 Surface data reconstruction

In the IBM, the solution is not known on the body surface, but it is required to find integral quantities such as Lift or Drag. In order to compute the aerodynamic coefficients, the value of the state on the surface must be reconstructed. The point on the surface are computed on each triangle of the STL file according to the desired quadrature formula. Once the solution on each triangle's integration points is found, the integral on the surface is computed. In this work, an Inverse Distance Weight (IDW) interpolation is performed [15, 11]: the set of points belonging to the fluid, closest to a surface point (SP) are selected and the solution on the SP is computed

as follows:

$$Q_{SP} = \frac{\sum_{i=1}^{Np} \frac{Q_i}{d_i}}{\sum_{i=1}^{Np} \frac{1}{d_i}}, \quad (3)$$

where Np is the number of interpolation points and d_i is the distance between the i^{th} -node and the surface's normal passing through the surface point SP (cfr. fig.2).

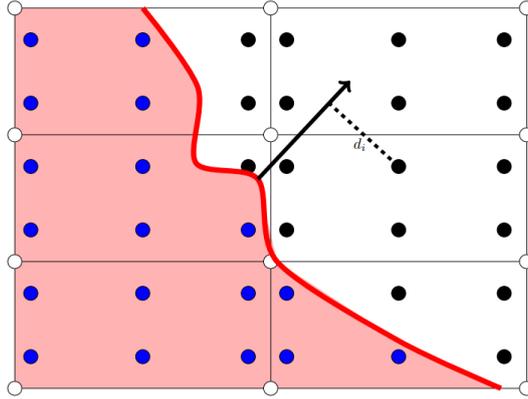


Figure 2: Visualization of the quantity d_i appearing in eq.3. The red region represents the portion of the domain occupied by the body.

2 Numerical results

The IBM previously described has been implemented in the 3D solver horses3D [1]. The tests are performed to validate the implementation and the results are compared with the published results. Moreover the performances of a standard explicit scheme (*i.e.* like Runge Kutta 3) and the FAS multigrid are shown. Both steady and unsteady results are provided. In all the simulations the penalization parameter η is set equal to the time step Δt .

2.1 Steady test case

The first test case is a steady cylinder at Reynolds $Re = 40$, Mach $Ma = 0.2$ and polynomial order 3. The mesh and the results are shown in fig.3. As can be seen in fig.3a, the mesh is a cartesian. Table 3 reports characteristic length scales for this particular case. We observed very good agreement with the literature. In fig.7 the drag coefficient C_d is plotted along with the number of interpolation points and observe good agreement when the number of points is larger than 20-40. The behavior of the residual is reported in fig.6: it is important to highlight that using an explicit scheme, like the standard Runge-Kutta, even with Strang splitting leads to a stagnation in the residual whereas if the FAS multigrid is used, the residual can easily reach the desired threshold in a low number of iterations. The price to pay for using the multigrid is

the increase in the computational time per iteration. The multigrid method is based on solving the system of equations by recursively iterating on solutions of different polynomial orders; as a consequence, a mask for each polynomial order is built.

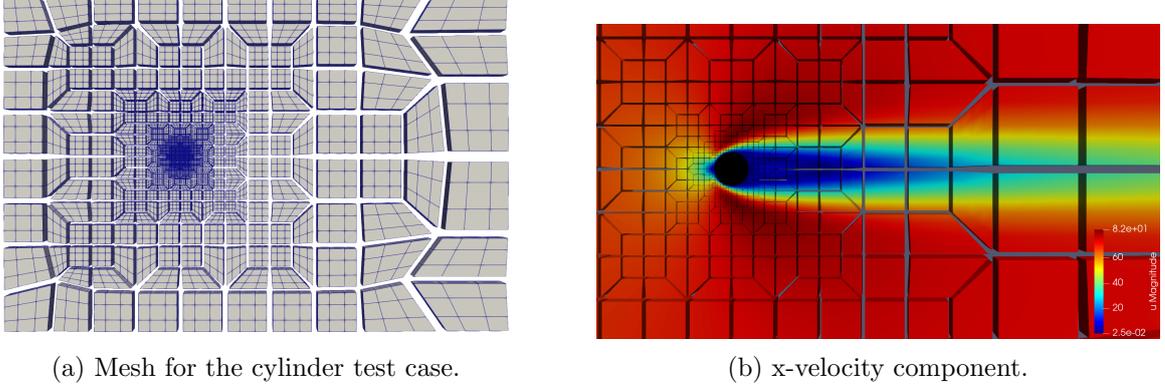


Figure 3: Mesh and simulation result of a cylinder at $Re = 40$ and $Ma = 0.2$.

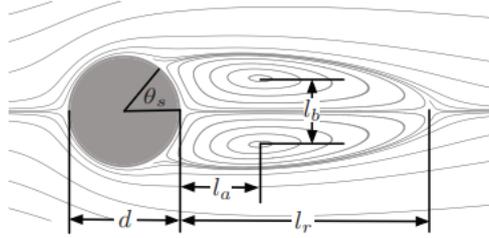


Figure 4: Visualization of the characteristic wake properties. Image from [7].

	$\frac{L_r}{d}$	$\frac{L_a}{d}$	$\frac{L_b}{d}$	θ_r
Linnick & Fasel (2005)	2.28	0.72	0.60	53.6
Taira & Colonius (2007)	2.30	0.73	0.60	53.7
IBM (present work)	2.29	0.72	0.58	53.6

Table 1: Comparison between literature results [22], [23] and the present work. See fig.4 for definitions.

2.2 Unsteady test case

An unsteady $Re = 100$, $Ma = 0.2$ at polynomial order 5 is simulated. In this case the FAS multigrid is used along with a pseudo-time step approach. The Strouhal number is reported and compared in tab.2. It can be seen that the value coming from the simulation is very close to the reference ones.

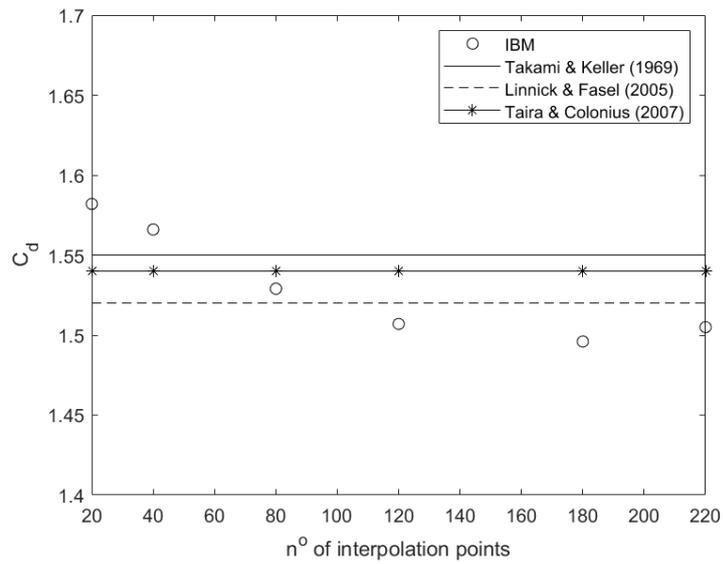


Figure 5: Drag coefficient vs number of interpolation points Np .

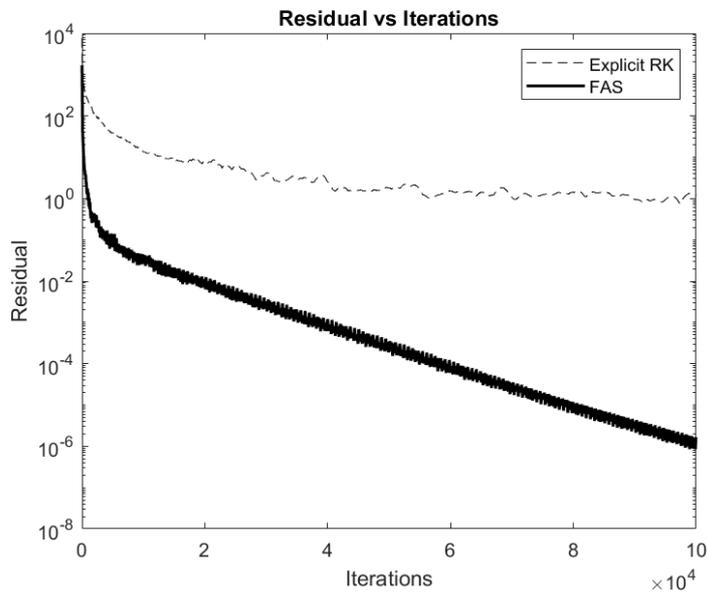


Figure 6: Explicit Runge Kutta 3 vs FAS.

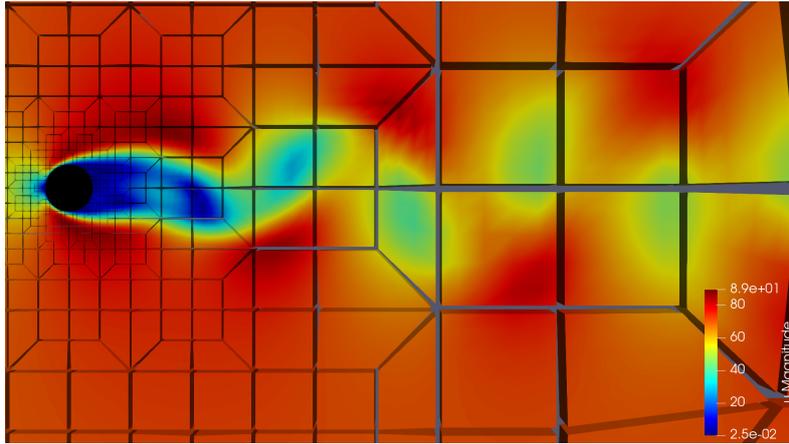


Figure 7: Wake behind a cylinder at $Re = 100$ and $Ma = 0.2$.

	St
Fet <i>et al.</i> (1998) [20]	0.165
Williamson (1992) [19]	0.161
Roshko (1954) [21]	0.167
IBM (present work)	0.160

Table 2: Comparison of the Strouhal number (St) for the $Re = 100$ test case.

3 Conclusions

An Immersed Boundary Method based on volume penalization has been successfully implemented and tested in the high order solver horses3D. To overcome the stiffness of the source term, a semi-implicit Strang splitting has been adopted. The robustness and efficiency of the mask generation have been assessed thanks to the use of SAH kd-tree coupled with ray-tracing. Finally the steady test case shows that coupling the IBM with a multigrid approach, FAS in particular, can highly reduce the number of iterations required to obtain the desired residual threshold.

ACKNOWLEDGEMENTS

This project received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 955923.

REFERENCES

- [1] Ferrer, E. and Rubio, G. and Ntoukas, G. and Laskowski, W. and Mariño, O. A. and Colombo, S. and Mateo-Gabín, A. and de Lara, F. Manrique and Huergo, D. and Manzanero, J. and Rueda-Ramírez, A. M. and Kopriva, D. A. and Valero, E. HORSES3D: a high-order discontinuous Galerkin solver for flow simulations and multi-physics applications. *arXiv* (2022) <https://doi.org/10.48550/arXiv.2206.09733> .

-
- [2] Mittal, R. and Iaccarino G. Immersed Boundary Methods. *Annual Review of Fluid Mechanics* (2005) **37**:239–261.
- [3] Ye, T., Mittal, R., Udaykumar, H.S. and Shyy, W. An accurate Cartesian Grid Method for Viscous Incompressible Flows with Complex Immersed Boundaries. *Journal of Computational Physics* (1999) **156**:209–240.
- [4] Majumdar, S., Iaccarino, G. and Durbin, PA. RANS solver with adaptive structured boundary non-conforming grids. *Annual Research Briefs 2001, Center for Turbulence Research* (2001):353–366.
- [5] Mittal, R., Dong, H., Bozkurttas, M., Najjar, F.M., Vargas, A. and von Loebbecke, A. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *Journal of Computational Physics* (2008) **227**:4825–4852.
- [6] Choi, J.-I., Oberoi, R.C., Edwards, J.R. and Rosati, J.A. An immersed boundary method for complex incompressible flows. *Journal of Computational Physics* (2007) **224**:757–784.
- [7] Canuto, D. Taira, K. (2015). Two-dimensional compressible viscous flow around a circular cylinder. *Journal of Fluid Mechanics*. (2015) **785**. 349-371.
- [8] Engels T., Kolomenskiy D., Schneider K., Sesterhenn J., Numerical simulation of fluid–structure interaction with the volume penalization method. *Journal of Computational Physics* (2015) **281**. 96-115.
- [9] Liu, Qianlong, and Oleg V. Vasilyev. A Brinkman penalization method for compressible flows in complex geometries. *Journal of Computational Physics* (2007) **227.2**: 946-966.
- [10] Kou, J., Hurtado-de-Mendoza, A., Joshi, S., Le Clainche, S., and Ferrer, E. Eigensolution analysis of immersed boundary method based on volume penalization: applications to high-order schemes. *Journal of Computational Physics* (2022) **449**:110817.
- [11] Kou, J., Joshi, S., Hurtado-de-Mendoza, A., Puri, K., Hirsch, C. and Ferrer E. Immersed boundary method for high-order flux reconstruction based on volume penalization. *Journal of Computational Physics* (2022) **448**:110721.
- [12] Rueda-Ramírez, A. Manzanero, J. Ferrer, E. Rubio, G. Valero, E. A p-multigrid strategy with anisotropic p-adaptation based on truncation errors for high-order discontinuous Galerkin methods. *Journal of Computational Physics* (2018) **378**.
- [13] Wald, Ingo Havran, Vlastimil, On Building Fast kd-trees for Ray Tracing, and on Doing that in $O(N \log N)$. *Symposium on Interactive Ray Tracing* (2006). 61-69.
- [14] L. Wu and Z. Chen, An Efficient SAH-Based KD-Tree Construction Algorithm for Ray Tracing *International Conference on Computational Intelligence and Software Engineering* (2009) 1-4.

-
- [15] Jung-Il Choi, Roshan C. Oberoi, Jack R. Edwards, Jacky A. Rosati, An immersed boundary method for complex incompressible flows. *Journal of Computational Physics* (2007) **224.2**: 757-784,
- [16] Mulder, W. and van Leer, B. Experiments with implicit upwind methods for the Euler equations. *Journal of Computational Physics* (1985) **59**:232-246.
- [17] Cuntanceau, M. and Bouard, R. Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow. *Journal of Fluid Mechanics* (1973) **79**:257-272.
- [18] Fornberg, B. A numerical study of steady viscous flow past a circular cylinder. *Journal of Fluid Mechanics* (1980) **98**:819-855.
- [19] Williamson, C.H.K. The natural and forced formation of spot-like' vortex dislocations' in the transition of a wake. *Journal of Fluid Mechanics* (1992) **243**:393-441.
- [20] Fet, U., König, M. and Eckelmann, H. A new Strouhal-Reynolds-number relationship for the circular cylinder in the range $47 < Re < 23 \cdot 10^5$. *Physics of Fluids* (1998) **10**:1547.
- [21] Roshko, A. On the Development of Turbulent Wakes from Vortex Streets. *Washington, DC, NACA Report 1191, National Advisory Committee for Aeronautics* (1954).
- [22] Linnick, M. N. Fasel, H. F. A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. *Journal of Computational Physics* (2005) **204**: 157-192.
- [23] Taira, K. Colonius, T. The immersed boundary method: a projection approach. *Journal of Computational Physics* (2007) **225**: 2118-2137.