

An improved immersed boundary method for turbulent flow simulations on Cartesian grids



Benjamin Constant^a, Stéphanie Pérone^{a,*}, Héloïse Beaugendre^b, Christophe Benoit^a

可压，消除压力和摩擦系数震荡

^a ONERA, Université Paris Saclay, F-92322, Châtillon, France

^b Univ. Bordeaux, INRIA, CNRS, Bordeaux INP, IMB, UMR 5251, F-33400, Talence, France

ARTICLE INFO

Article history:

Available online 26 February 2021

Keywords:

Immersed boundary method
Wall model
Cartesian mesh
Turbulent flows

ABSTRACT

In this paper, we present recent improvements of an Immersed Boundary Method (IBM) for the simulation of turbulent compressible flows on Cartesian grids. The proposed approach enables to remove spurious oscillations at the wall on skin pressure and friction coefficients. Results are compared to a body-fitted approach using the same wall function, showing that the stair-step immersed boundary provides a smooth solution compared to the body-fitted one.

The immersed boundary method has been modified to adapt the location of forced and forcing points involved in the immersed boundary reconstruction to the Reynolds number. This method has been validated either for subsonic and transonic flow regimes, through the simulation of the subsonic turbulent flow around a NACA0012 profile and the transonic flow around a RAE2822 profile and the three-dimensional ONERA M6 wing.

© 2021 Elsevier Inc. All rights reserved.

1. Introduction

Within the past two decades, Computational Fluid Dynamics (CFD) has dramatically risen in the aerospace industry to simulate more and more complex configurations. Reynolds-averaged Navier-Stokes (RANS) simulations on body-fitted or conformal meshes are commonly performed efficiently onto realistic geometries due to the improvements in terms of robustness and accuracy of CFD solvers within the past two decades. Today, industrial configurations take into account complex geometrical details, such as aircraft track fairings or a helicopter rotor head to evaluate their influence on the overall performances of the aircraft or rotorcraft. As a result, the mesh generation has become the bottleneck of the CFD workflow, as it requires manual interaction and expertise, with several weeks of engineer time in comparison with a day for running a RANS simulation.

For this reason, immersed boundary methods (IBM) have become popular within the last decade as the mesh does not need to conform to the obstacles, simplifying significantly the mesh generation. This concept was introduced by Peskin [1,2] many decades ago to simulate fluid-structure interactions in the heart, where a Cartesian mesh was used to simulate blood flow. An immersed boundary condition (IBC) is applied to take into account the obstacles lying in the flow. IBCs can be classified into two main approaches, as referred by Mittal and Iaccarino [3]. The first class of methods consists in introducing a forcing term within the equations, that can represent the exchange of momentum between the fluid and solid through a law based on the theory of elasticity and is well suited for flows with immersed elastic boundaries [1,4], but

* Corresponding author.

E-mail address: stephanie.peron@onera.fr (S. Pérone).

leads to a lack of stability and accuracy for rigid bodies as the problem becomes stiff. Other methods have been developed, among which the penalization method [5] or artificial spring [6]. The flow equations are solved on the whole computational domain, including the solid region, and the forcing is diffused on neighbouring mesh points of the solid interface. IBM can be used on the whole geometry [7,8] or locally [9] to capture the potential effects of geometrical details.

Another approach, similar to IBMs, is the cut-cell approach, which consists in computing directly the intersection between the cells and the obstacles, which has proven efficient for inviscid flow simulations and low Reynolds flows around complex geometries (see Coirier & Powell [10], Berger & Aftosmis [11] and Harada et al. [12]). Contrary to IBM approaches, it has the advantage to be conservative as the flux is directly computed onto the interface. On the other hand, cutting cells can result in badly shaped cells, requiring some additional geometrical treatments (e.g. merging some cells).

The use of Cartesian grids with local grid refinement in combination with embedded obstacles (either with immersed boundary or cut-cell methods) seems to be well-suited for a high-level of automation and computational efficiency [7,13,11]. IBMs on Cartesian grids are either used by Euler or Navier-Stokes solvers or Lattice Boltzmann solvers.

Wall functions are usually required to solve turbulent flows using IBM on adaptive Cartesian grids around arbitrary obstacles to restrict the number of mesh points, as the cell size is equal both in the wall normal and tangential directions [14–17]. Although the use of adaptive Cartesian grids around arbitrary immersed obstacles using wall functions is conceptually attractive, spurious oscillations occur when extracting the skin quantities (pressure and skin friction), which is a strong issue for aerospace applications. Actually, wall-modelled immersed boundary approaches usually use a forcing/reference/image point in the fluid, where the flow field is interpolated [18,19,15,16]. The wall function is applied to compute the friction velocity at this reference point, which is used to reconstruct the tangential velocity at the IBM forced/target point. Consequently, the wall-modelled near-wall resolution is too coarse to capture correctly the nonlinear tangential velocity profile in the wall normal direction. Spurious oscillations appear in the near-wall regions, that can be dissipated far from the wall boundaries provided the numerical scheme is dissipative enough, but are visible on the skin quantities. For that purpose, Capizzano [14] proposed to linearize the velocity profile down to the wall from the reference/image point, while Tamaki et al. [20] proposed an extension of this approach by modifying the eddy viscosity profile too in the Spalart-Allmaras turbulence model, to preserve the balance of the shear stress in the boundary layer.

Here, another approach is presented, based on geometrical considerations and an analogy with wall-modelled body-fitted approaches. Near-wall spacing and location of IBM reference/image points (that will be called *IB image points* in the following) must be consistent with the Reynolds number of the CFD simulation (which can be of ten to one hundred millions for a CFD simulation around an aircraft). The proposed approach relies on the previous work of Péron et al. [16], adapted to determine the location of IB target points, where the solution is reconstructed, in agreement with the near-wall spacing and the Reynolds number of the simulation.

The paper is organized as follows: in section 2, the mathematical models and the flow solver are presented. In section 3, the wall modelling by the immersed boundary method along with the location of image points are detailed. Then the improved immersed boundary treatment is presented in section 4. Simulations of turbulent flows in subsonic and transonic regimes around 2D airfoils and around the ONERA M6 wing are presented to assess the improvements of the proposed method in section 5. Results are compared with results obtained with the former IBM treatments but also with body-fitted wall-modelled solutions and body-fitted solutions where the near-wall discretization enables to capture the boundary layer correctly. Finally, conclusions are drawn in section 6.

2. Description of the flow solver

2.1. Governing equations

The steady-state mean flow is obtained by solving the compressible RANS equations (1):

$$\left\{ \begin{array}{l} \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j} (\rho u_j) = 0 \\ \frac{\partial \rho u_i}{\partial t} + \frac{\partial}{\partial x_j} (\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} (\tau_{ij}) \quad i = 1, 2, 3 \\ \frac{\partial \rho E}{\partial t} + \frac{\partial}{\partial x_j} ((\rho E + p) u_j) = -\frac{\partial}{\partial x_j} (Q_j) + \frac{\partial}{\partial x_j} (\tau_{ij} u_i) \end{array} \right. \quad (1)$$

where ρ , u_i , p , E denote the fluid density, the mean velocity components, the pressure and the total energy per unit mass respectively, Q is the heat flux vector and τ_{ij} the total shear stress tensor, decomposed into the viscous stress and the Reynolds stress modelled here by Spalart-Allmaras model [21], which is chosen for its robustness and its low computational cost.

The Spalart-Allmaras model is a one-equation transport model that consists in introducing a pseudo eddy viscosity $\tilde{\nu}$ as follows:

$$\nu_t = \tilde{\nu} f_{v1}, \text{ where } f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \chi = \frac{\tilde{\nu}}{\nu} \text{ and } c_{v1} = 7.1, \quad (2)$$



Fig. 1. Ghost cell updates on octree-based Cartesian grids.

where ν is the kinematic viscosity of the fluid. The damping function $f_{\nu 1}$ goes to 0 to the wall and 1 in the log layer, such that this pseudo viscosity corresponds to the eddy viscosity ν_t in the log layer.

2.2. Numerical methods

The compressible RANS equations are solved using FAST CFD solver developed at ONERA [22,23], relying on a 2nd-order accurate cell-centered Finite-Volume Method on multiblock structured grids. One major feature of this solver is its efficiency in dealing with unsteady simulations (see [24]) as it enables to update 10 million cells per second per core on a single Intel Broadwell core. It also contains a solver dedicated to Cartesian grids which is 2.5 times more efficient in terms of memory and CPU time than the curvilinear block-structured solver with the same numerical ingredients. For this reason, despite the highest cell count resulting from the generation of a block-structured Cartesian mesh in comparison with a body-fitted approach (structured or unstructured), this extra cost is counterbalanced by the reduced memory requirements (neither the mesh nor the metrics need to be stored, numerical fluxes are simplified...).

FastS solver relies on an hybrid MPI/OpenMP framework, where the memory is distributed (by distributing CFD grids) on the processors at high level, i.e. between nodes, whereas multithreading is managed via OpenMP within a given node. For our purpose, where Cartesian grids are uniform and contain few cells in comparison with grids resolving boundary layers accurately, Cartesian grids are distributed between the cores using OpenMP.

For RANS computations, two second-order accurate spatial schemes are considered, depending on the flow regime: the Roe-MUSCL scheme [25] for transonic flow simulations or an AUSM scheme [26] for subsonic flow simulations, which is based on a modification of the AUSM+(P) scheme (see Edwards & Liou [27]). Jacobian approximations are those proposed by Jameson & Yoon [28] and Coakley [29], whereas the linear system is solved by the LU-SGS method [28].

In our approach, the octree-based Cartesian mesh is defined by a set of overset grids with a minimum overlapping, as described in [30,16]. Consequently, two adjacent grids of same refinement level are extended from two layers of cells on both sides and the solution is directly updated for the two layers of border cells, as depicted in Fig. 1-(a). In the case of two adjacent grids where the left-hand side one is fine and the right-hand side one is coarse, as depicted in Fig. 1-(b), the fine grid is extended from two cells and the coarse one from three cells to ensure that donor cells are always interior points and thus already updated. Interpolation is achieved by a 2nd-order Lagrange interpolation.

In the following, W will denote the flow solution $W = (\rho, \rho u_i, \rho E, \tilde{v})$.

3. Wall modelling by immersed boundaries

3.1. Principles

The immersed boundary method proposed in this paper relies on a ghost-cell direct forcing formulation, described in [16]. This approach consists in imposing the flow variables W at some points close to the obstacles to mimic a boundary condition. These points will be called *IB target* points in the following.

First, solid and fluid regions are identified geometrically, by a hole-cutting algorithm [31]. If a cell centre is found lying inside the solid, the cell is marked as blanked and it will not be updated by the CFD solver during the simulation. If a Cartesian grid is entirely blanked, it is removed during the pre-processing to save memory.

To be consistent with the CFD solver, based on a five-point stencil and thus requiring two layers of ghost cells, two layers of IB target cells must be defined. The previous method described in [16] consists in marking as IB target cells the first two layers of cells at the fringe of blanked cells in the fluid part of the mesh, represented by red dots in Fig. 2. Flow solution W is obtained at *IB target point A* by using an *IB image point* (point B marked by a blue dot in Fig. 2) located within the fluid region along the wall normal direction at point A. As this point B, also known as a reference point [17], is not a discretization point, the flow field at this point is obtained by a 2nd-order interpolation from its surrounding points (also called *donor points*, represented by green dots in Fig. 2).

As described in [16], solution W at image point B is used to reconstruct the flow field W at target point A. Since the pressure p is assumed to be constant in the wall normal direction in the turbulent boundary layer, we then get $p_A = p_B$. In order to take into account the compressibility effects within the boundary layer, the Crocco-Busemann relationship with the adiabatic wall condition provides the temperature at point A T_A as follows:

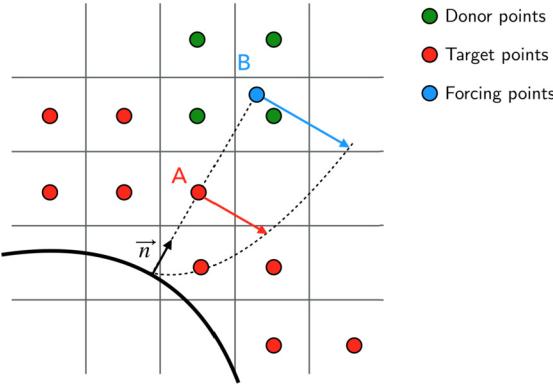


Fig. 2. Direct forcing on an IB target point (A) using its corresponding image point (B). (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

$$T_A = T_B + \frac{\sqrt[3]{Pr}}{2C_P} (u_{t,B}^2 - u_{t,A}^2), \quad (3)$$

where Pr is the Prandtl number and C_P is the specific heat coefficient at constant pressure. This relationship is used to recover the density at point A, that is

$$\rho_A = \frac{p_A}{RT_A}, \text{ where } R \text{ is the perfect gas constant.} \quad (4)$$

For high Reynolds number flows, solving directly the RANS equations on Cartesian grids, where the near-wall spacing is the same in the wall normal and the wall tangential directions, would lead to a mesh that is too prohibitive to capture the velocity gradient accurately. For that purpose, IBM methods on Cartesian grids usually apply wall functions, either based on algebraic models [17,16,32] or TBLE-based (for thin boundary layer equation) models [33,15].

As detailed in [16], an algebraic wall function proposed by Musker [34], that fits either the log, buffer and viscous sublayers, is used to reconstruct the velocity at IB target point A. First, the friction velocity u_τ is evaluated at IB image point B and is used to reconstruct the tangential velocity u_t at target point A as

$$u_{t,A} = u_\tau f(y_A^+), \text{ with } y_A^+ = \frac{y_A u_\tau}{\nu}. \quad (5)$$

The normal velocity $u_{n,A}$ at point A is estimated by a linear reconstruction as

$$u_{n,A} = \frac{y_A}{y_B} u_{n,B}, \quad (6)$$

where y_A and y_B are the distances of points A and B to the wall boundary.

As the expression $v_t = \kappa u_\tau y D$ is valid in the log layer and down to the wall (where u_τ is the friction velocity, y is the wall distance, $\kappa = 0.41$ is the Von Kármán constant, D is the Van Driest damping term, with $D = (1 - e^{-\frac{y^+}{19}})$, with $y^+ = \frac{u_\tau y}{\nu}$), the pseudo-viscosity $\tilde{\nu}$ can be evaluated at target point A solving the nonlinear equation

$$\tilde{\nu}^4 - \kappa u_\tau y D \tilde{\nu}^3 - \kappa u_\tau y D v^3 C_{v1}^3 = 0, \quad (7)$$

derived from equation (2), as described in details in [16].

3.2. Location of image points

In order to reconstruct the solution $W_A^{(n)}$ at iteration n , $W_B^{(n)}$ at image point B must be interpolated from donor points already updated at iteration n , thus from computed points only. This means that the image point B has to be chosen far enough to prevent target points from being donor points. On the other hand, these points must not be too far from the wall to prevent them from being out of the boundary layer. In the literature, most approaches define image points at a given iso-distance to the wall [20,17].

A good compromise is to set that distance to $d_B = \sqrt{2} N_{diag} \times h_{local}$ in 2D and $d_B = \sqrt{3} N_{diag} \times h_{local}$ in 3D, where h_{local} the local spacing of the mesh in the vicinity of target point A and N_{diag} depends on the spatial stencil for the flow solver. Indeed, the worst case is obtained when the normal direction from the wall to target point A is oriented along the diagonal of the cell. For a spatial scheme in the solver that requires one cell per diagonal direction, then N_{diag} is equal to 1.

However, the lack of robustness of this approach can occur for complex geometries, especially at concave corners, where the condition that target or blanked points must not be donor points for the image point is not guaranteed. Moreover,



Fig. 3. Location of IB points for the classical method around the NACA0012 airfoil.

this distance depends strongly on the local spacing near the wall, which may not be constant in the tangential direction. Typically, different components of a complex configuration, such as a fuselage with a wing, may not be defined by the same near-wall spatial resolution to save mesh points and memory.

For that purpose, a method that computes automatically the location of image points that ensures both previous conditions has been developed [16] and is applied here. It consists of determining the fringe of first computed cells surrounding the target points. This fringe is a set of surface grids, represented by stair steps, that define a watertight surface, even if that front intersects Cartesian grids of different levels of refinement. First, target points (displayed in Fig. 3 by red dots) are projected onto that front, resulting in blue dots in Fig. 3, following the wall normal direction \mathbf{n} , obtained by a normalization of the gradient of the distance to the wall at target points. Then, corresponding IB wall points are obtained by the same projection in the opposite direction (represented in black dots in Fig. 3).

3.3. Post-processing in an immersed boundary context

Unlike body-fitted approaches, it is not possible to extract the flow fields directly at wall boundaries. A reconstruction must be performed to obtain some quantities such as skin friction or loads and to visualize them on the obstacles. For that purpose, the Moving Least Squares (or MLS) method is performed. It has been initially built up for the generation of surfaces [35] and has been derived to provide spatial approximations for meshless methods [36,37].

In our approach, the flow quantities (pressure, density, friction velocity) are extracted at IB wall points at a given iteration of the flow solver. As detailed in section 3.1, and similarly to the flow field reconstruction implemented for target points, the wall pressure is extrapolated at each wall point w from its corresponding image point B , which leads to $p_w = p_B$. The wall density is then recovered using the Crocco-Busemann relationship between the image point and the wall, where the tangential velocity is assumed to be zero:

$$T_w = T_B + \frac{\sqrt[3]{Pr}}{2C_p}(u_{t,B}^2), \quad (8)$$

$$\rho_w = \frac{p_w}{RT_w}. \quad (9)$$

These variables are interpolated using a 3rd-order accurate MLS interpolation onto the vertices of a triangular mesh describing the obstacles on which the skin quantities are required. For each vertex VT of the tessellation, a point cloud made by at least 10 IB wall points surrounding the vertex VT is used to project the solution on that vertex using the MLS algorithm. Consequently, in order not to introduce interpolation errors due to strong discrepancies between the distribution of IB wall points and the tessellation (e.g. a MLS stencil 10 times larger than the characteristics length of the target triangle), the tessellation should be consistent with the Cartesian mesh discretization in the vicinity of the obstacles.

After extraction, the skin pressure and friction coefficients can be computed on the surface:

$$C_p = \frac{p_w - p_\infty}{\frac{1}{2} \rho_\infty U_\infty^2} \quad (10)$$

$$C_f = \frac{\tau_\omega}{\frac{1}{2} \rho_\infty U_\infty^2} = \frac{\rho_w u_\tau^2}{\frac{1}{2} \rho_\infty U_\infty^2} \quad (11)$$

4. Improved immersed boundary treatment

4.1. Motivation

The aforementioned IBM treatment leads to spurious oscillations on the skin pressure and friction coefficients. This has also been observed by Capizzano [14], Tamaki et al. [20] and Wilhelm et al. [17]. Actually, the numerical scheme in combination with the near-wall uniform spacing is not capable of capturing the strong variations of the tangential velocity

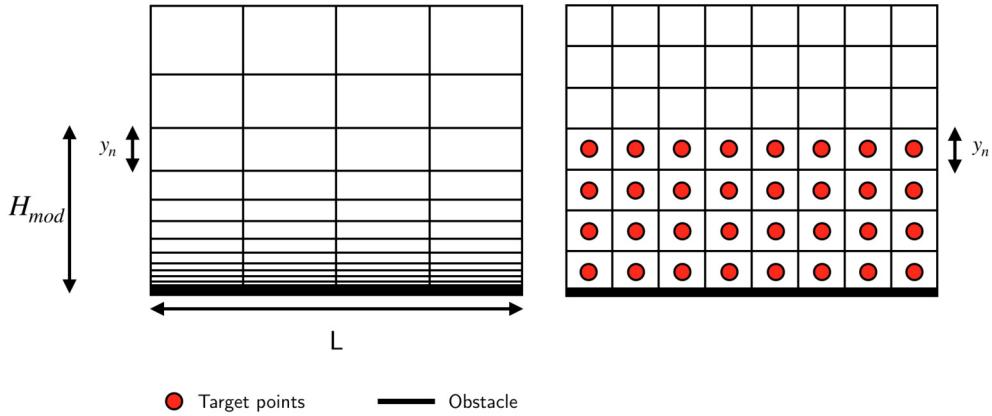


Fig. 4. Comparison between Cartesian and body-fitted meshes for a flat plate of length L .

in the wall normal direction. In addition, the stair-step distribution of IB target and image points can lead to neighbouring points that do not lie within the same sublayer of the turbulent boundary layer, e.g. the first one being in the viscous sublayer and the second one in the log layer, leading to oscillations in the near-wall region of the solution, as their velocity gradients are different although they represent two boundary points in the stencil of the first computed point above.

Capizzano [14] and Tamaki et al. [20] proposed a linearization of the velocity profile between the wall and the image point location and a modification of the eddy-viscosity profile to maintain the balance of the shear stress.

In this paper, another approach is proposed, based on geometrical considerations about the location of IB target and image points. The novel treatment consists in applying the immersed boundary condition at a fixed y^+ , such that IB target and image points are located in the log layer, for cell count concerns. The objective is to mimic the wall-modelled body-fitted approach using the wall-modelled IBM treatment on Cartesian grids. It can be noted that this method has the advantage of being independent from the chosen wall model and the turbulence model.

4.2. Estimation of IB target and image point location

Contrary to the improvements of the IBM treatment found in the literature [14,20], for which the turbulence model and the numerical scheme are modified, the present approach is based upon geometrical aspects that concern the location of target and image points with respect to the wall. It relies on the flat-plate boundary layer theory that enables to compute the height of the first cell off the wall for a desired value of y^+ .

Let us consider the case where the boundary layer is fully resolved by the mesh (Fig. 4-(left)). Then consider a Cartesian mesh of uniform near-wall spacing y_n , as depicted in Fig. 4-(right). In order to be consistent with a mesh where cells stretch off the wall, the wall-modelled IBM treatment on the Cartesian mesh of uniform cell spacing y_n implies to locate the IB image points at distance H_{mod} from the wall, hereafter called *modelling height*. This implies that the near-wall spacing alone is not sufficient to model the boundary layer correctly, since it must also be consistent with the wall distance at which the wall model is applied (at IB image points). This is why no mesh convergence was possible using the former treatment [38], as the first computed points still fall in a region where the nonlinearity of the velocity cannot be well represented by the uniform spacing.

In practice, it is proposed to provide a value of y^+ for the first computed points just above the last layer of IB target cells, called y_{target}^+ . As the IB image points are located on the front of first computed cells in our approach, this implies that the value of y^+ at image points is nearly y_{target}^+ . In our framework, the near-wall spacing y_n is imposed first, as it can depend on the geometrical features that must be represented by the mesh, such as a variation of the curvature or gaps between very close bodies. Then, a flat plate approximation is performed to evaluate the distance to the wall of the first computed cells H_{mod} , given the values of the Reynolds number, the characteristic length L of the geometry and the flow conditions. For a fixed y_{target}^+ , this resulting height H_{mod} provides a first approximation of the threshold between computed points and modelled IB target points where the solution is reconstructed. In order to apply the wall model within the log sublayer, the y_{target}^+ should be set between 50 and 300. Moreover, the choice of a specific y_{target}^+ value in this region should be made so as to ensure that the cell spacing of the first calculated points inside the Cartesian mesh is consistent with the cell spacing required for a body-conformal mesh at the same distance to the wall. The algorithm is described as follows:

Let denote $y^+ = \frac{u_\tau y}{v}$, where $u_\tau = \sqrt{\frac{\tau_\omega}{\rho}}$, with τ_ω the wall shear stress and $C_f = \frac{\tau_\omega}{\frac{1}{2} \rho_\infty U_\infty^2} = f(Re)$, the modelling height

H_{mod} is then defined by:

$$H_{mod} = \sqrt{2} \frac{y_{target}^+ L}{Re \sqrt{f(Re)}}, \quad (12)$$

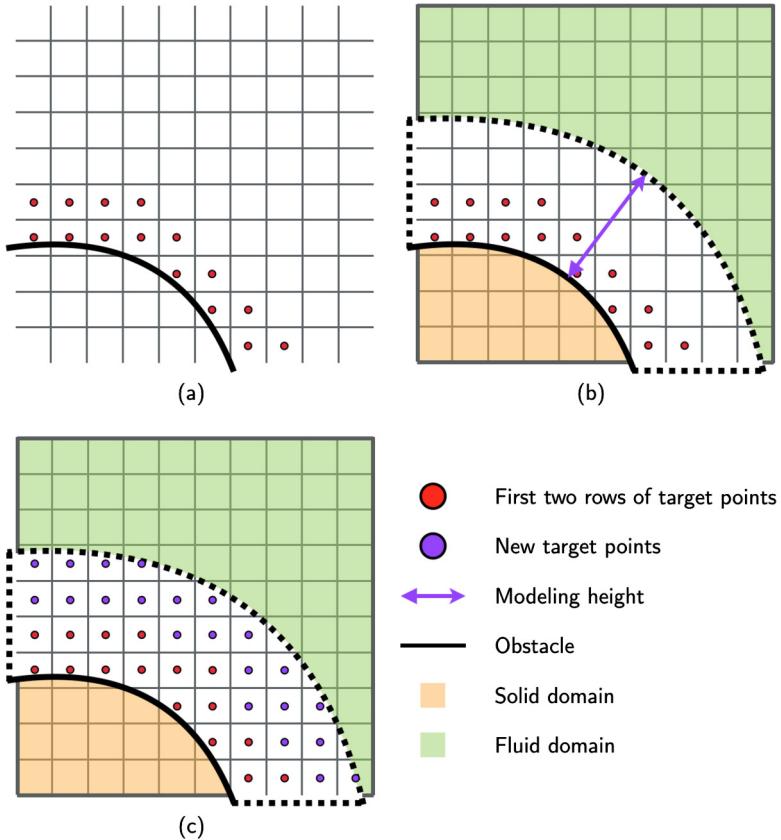


Fig. 5. IB target point: former location (a) and new positioning (c) according to the threshold distance between the modelled and computed regions (b).

where the function $f(Re) = 0.058Re^{-0.2}$ is a flat plate approximation for the turbulent boundary layer.

Conversely, the optimal y_{target}^+ for a given near-wall spacing y_n can be obtained by using the same flat plate approximation. Considering that the spacing of the body-fitted mesh displayed in Fig. 4-(left) is following a geometric progression with common ratio $q = 1.2$, and given that y_n also corresponds to its n -th cell, we can deduce that $y_n = y_0 q^n$, where y_0 is the height of the first cell. Since this first cell is generally chosen to impose $y^+ = 1$ at the wall and considering previous calculations, we may write:

$$y_0 = \sqrt{2} \frac{L}{Re \sqrt{f(Re)}}. \quad (13)$$

$H_{mod} = \sum_{i=1}^n y_i$ can also be rewritten:

$$H_{mod} = y_0 \frac{1 - q^{n+1}}{1 - q} = \frac{y_0 - q y_n}{1 - q}. \quad (14)$$

Finally, using equations (12), (13) and (14), the optimal y_{target}^+ is given by:

$$y_{target}^+ = H_{mod}/y_0 \quad (15)$$

Fig. 5 describes the modified IBM preprocessing.

- The former method consists in marking the two first layers of cells at the fringe of solid cells as IB target points (Fig. 5-(a)).
- The threshold distance to wall H_{mod} is estimated using the flat plate approximation (Fig. 5-(b)).
- IB target points are the two layers of cells just below the surface of distance H_{mod} (Fig. 5-(c)).

For three-dimensional test-cases, a change in the near-wall spacing is often required in the tangential direction to the wall to prevent a huge amount of mesh points. As the Cartesian mesh is octree-based, one refinement level can be crossed between neighbouring grids. Thus, in the vicinity of the border between a fine and a twice as coarse grid near the wall,

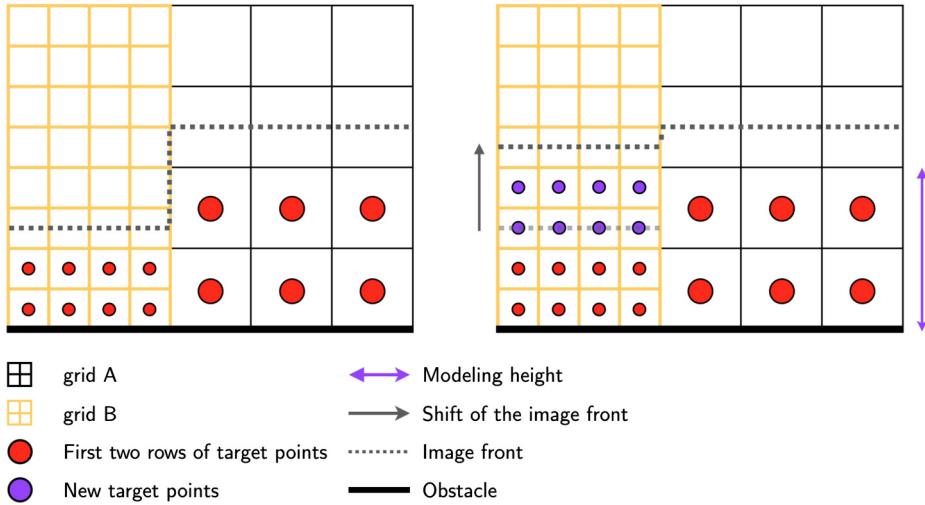


Fig. 6. IBM location: impact of the change of refinement level in the tangential direction to the wall.

Table 1
Classification of algorithms for IB target and image point location.

-	Layers of target points	Front type of image points
F0	2	fixed distance to the wall (smooth front)
F1	2	stair-step front
F2	3	stair-step front
F42	adaptive	stair-step front

the IBM reconstruction is achieved at IB target points using image points that might be located in different regions of the boundary layer, as depicted in Fig. 6-(left). This leads to a strong discrepancy in the estimation of the velocity at target points that are neighbours in the tangential direction but whose refinement level is not the same. This can be observed on the skin friction coefficient, where a spurious oscillation occurs in the vicinity of the change of the mesh refinement level near the wall. This will be discussed in section 5.3 on the ONERA M6 wing configuration, where the leading edge is refined to take into account the variation of the curvature while the other near-wall regions are twice as coarse to avoid the cell count to be too high.

A consequence of the novel treatment is the location of IB image points at roughly the same distance to the wall, as depicted in Fig. 6-(right). The location of the new target points is also modified, resulting in nearly the same threshold height between the modelled and computed regions, independently from the refinement level of the near-wall grid. In section 5.3, the removal of the spurious oscillations on the skin friction in the vicinity of the change of refinement level will be demonstrated.

4.3. Summary of different IBM treatments

As several IBM treatments will be compared in the following, Table 1 summarizes their characteristics regarding the location of IB target and image points:

An illustration of the location of IB target and image points is provided in Fig. 7 for the four IBM treatments in the case where a change in the refinement level occurs tangentially to the wall.

5. Numerical results

In order to demonstrate that the modified near-wall geometrical treatment of the proposed immersed boundary method enables to improve significantly the distribution of the skin pressure and friction, bidimensional and three-dimensional test-cases are considered in subsonic and transonic regimes. The influence of the location of IB target points and image points is highlighted by a comparison between the different approaches for determining their positions. Results are compared with a body-fitted approach where the boundary layer is resolved by the mesh. In addition, a comparison between the IBM results and a body-fitted approach with a wall function applied at same y^+ assesses the influence of the wall function onto the accuracy of the solution. This study demonstrates that the improved IBM approach provides similar results to wall-law body-fitted simulation, despite the stair-step fashion of IB target and image points.

Steady RANS simulations are performed for IBM on Cartesian grids and body-fitted mesh using FAST solve, with the same numerical ingredients, in particular the same spatial schemes and turbulence model. Musker's wall function is applied

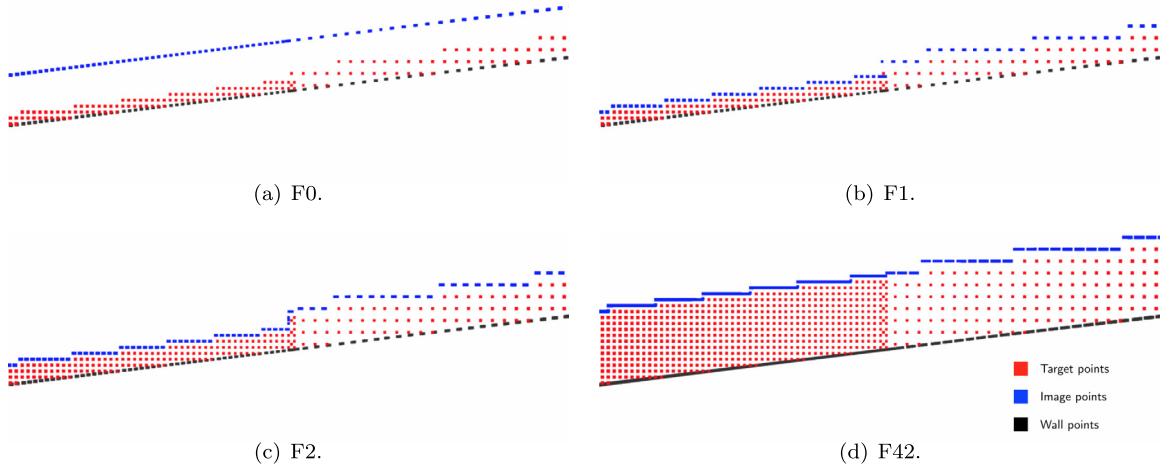


Fig. 7. Comparison of the location of IB points for the four treatments near a wall boundary (in black), with a change of refinement level along the wall.

Table 2
Cartesian mesh characteristics for the subsonic NACA0012 airfoil test-case.

-	IBM5	IBM6	IBM7	IBM8	BF
Minimum spacing h_{min}/c	0.0002	0.0001	0.00005	0.00002	10^{-6}
Number of pts (millions)	4.1	7.8	16.4	34.9	16.1

either within the body-fitted approach for the wall-law simulations and within the IBM approach on Cartesian grids to reconstruct the velocity and friction at IBM points.

In the following, most of the IBM simulations using the improved algorithm are performed with $y^+ = 100$ for the image points along the geometry.

Imposing $y^+ = 100$ enables to apply the wall model in the log sub-layer and at the same time to guarantee that the cell spacing of the first calculated points above target points is in agreement to the cell spacing required for a body-conformal mesh at the same distance to the wall.

Two bidimensional turbulent flow simulations around airfoils in the subsonic and transonic regimes are considered to assess the improvements in terms of skin pressure and friction of the proposed method. Results are compared to a solution obtained on a wall-resolved conformal mesh and to experiments. The modified immersed boundary treatment is compared with previous methods. The influence of the near-wall spacing and the angle of attack is also considered. A comparison between a body-fitted wall-modelled solution and an IBM solution is also achieved to make a distinction between the influence of the wall model and of the stair-step IBM. Finally, a three-dimensional turbulent flow simulation around the ONERA M6 wing in the transonic regime is performed. Results are compared with a body-fitted solution and experimental data.

5.1. Simulation of the subsonic turbulent flow around a NACA0012 profile

The first test-case is the subsonic turbulent flow around a NACA0012 airfoil under the following flow conditions: freestream Mach number $M_\infty = 0.15$, a zero angle of attack and Reynolds number based on the chord is $Re = 6$ millions. The Cartesian mesh is automatically generated according to the mesh resolution imposed around the airfoil. The extent of the computational domain corresponds to $40c$, where c is the chord length and is equal to 1. Farfield boundary conditions are defined at the external boundaries and an immersed boundary condition is applied on the airfoil (see Fig. 8).

The steady RANS equations are discretized with the 2nd-order accurate AUSM spatial scheme described in paragraph 2.2. An implicit backward Euler scheme is performed to solve the pseudo-time integration.

5.1.1. Influence of the location of IBM target and image points

This paragraph aims at evaluating how the location of target and image points is crucial to capture accurately skin quantities. The mesh settings for this test-case are gathered in Table 2.

Fig. 9 displays the isocontours of the Mach number for a body-fitted simulation where the boundary layer is solved by the mesh and an IBM wall-modelled simulation on Cartesian mesh IBM7, showing a good agreement between both solutions.

Fig. 10 compares the different IBM treatments for the same near wall spacing (10^{-4}), which corresponds to the IBM6 Cartesian mesh defined in Table 2. FAST IBM6 F0 and FAST IBM6 F1 results present strong oscillations along the chord

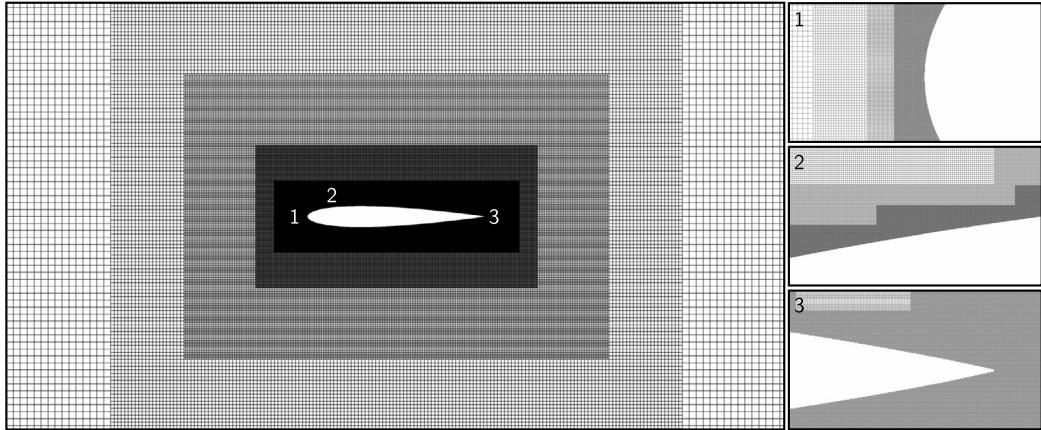
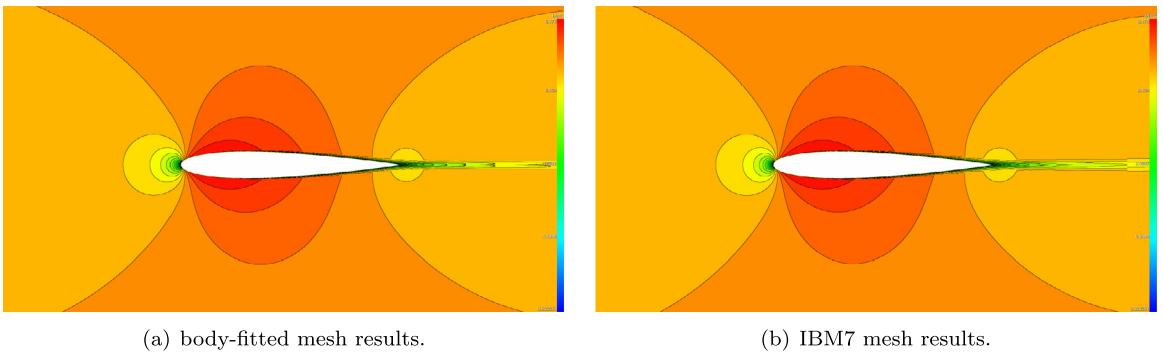


Fig. 8. Views of the IBM Cartesian mesh around the NACA0012 airfoil.



(a) body-fitted mesh results.

(b) IBM7 mesh results.

Fig. 9. NACA0012 airfoil, $M_\infty = 0.15$, $Re = 6$ millions: Mach number flowfield.

Table 3

NACA0012 airfoil, $M_\infty = 0.15$, $Re = 6$ millions: mean values of y^+ for image points with respect to the location of image and target points.

-	IBM6 F0	IBM6 F1	IBM6 F2	IBM6 F42	BF-WM
y^+	82	54	66	123	124

for both skin coefficients, especially near the leading edge. Despite the oscillations, the trend of the skin pressure distribution compares fairly to the body-fitted solution FAST BF. But the skin friction coefficient is over-estimated and deviates significantly from the reference solution near the leading edge.

The F2 algorithm appears as a better option as it suppresses all the oscillations on FAST IBM6 F2 for both coefficients for the chosen near-wall resolution. The skin pressure distribution matches the reference solution, whereas it still results in an over-estimation of the skin friction near the leading edge, at a lower level than F0 and F1 algorithms. The improvement is probably due to a better agreement with the location of IB image points and the chosen near-wall resolution. We observed that the spurious oscillations eventually appeared when the near-wall resolution was increased.

Finally, the improved method is the only treatment among the four treatments considered here that is capable of both removing spurious oscillations and matching the body-fitted reference, as the skin friction coefficient is no longer over-estimated near the leading edge for FAST IBM6 F42.

Furthermore, Table 3 confirms that ensuring that both target and image points lay in the same sub-layer is indeed important to remove spurious oscillations, especially in regions where the wall presents a high curvature. The average y^+ obtained with F0 might appear sufficient to ensure such a requirement, but one should keep in mind that the IB image points are projected at a fixed distance to the wall during preprocessing, as explained in Table 1, whereas the IB target points can lie in the log layer but also in the viscous sublayer due to their stair-step distribution near curved boundaries.

5.1.2. Influence of the near-wall resolution

In this study, the improved treatment (F42) is chosen. Several uniform near-wall resolutions are evaluated, from coarse (IBM5) to ultra-fine (IBM8), summarized in Table 2. In [38], Renaud et al. observed no mesh convergence for the same test-case using the F0 IBM treatment. They observed a good agreement of the solution in terms of pressure coefficient,

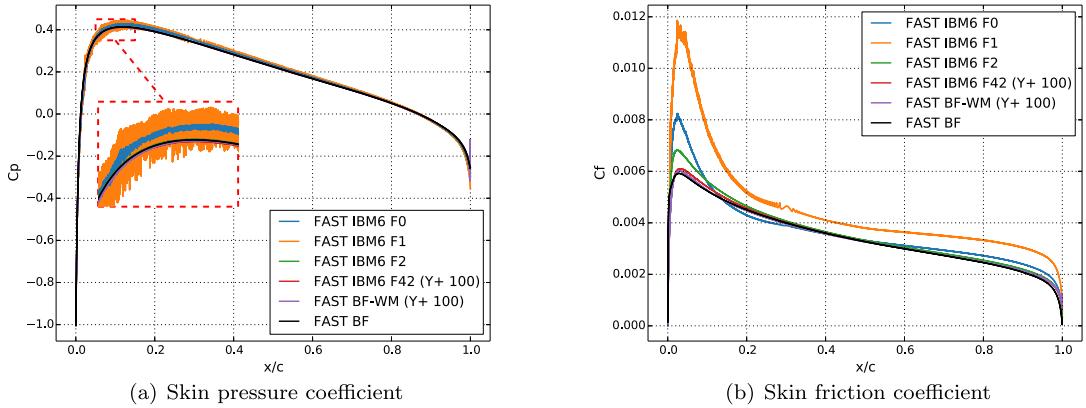


Fig. 10. NACA0012 airfoil, $M_\infty = 0.15$, $Re = 6$ millions: influence of the location of target and image points.

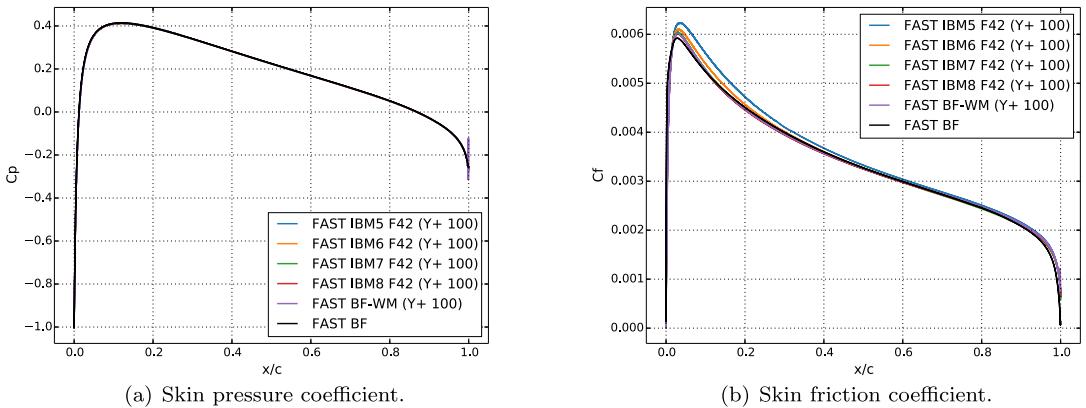


Fig. 11. NACA0012 airfoil, $M_\infty = 0.15$, $Re = 6$ millions: influence of the near-wall resolution.

Table 4

NACA0012 airfoil, $M_\infty = 0.15$, $Re = 6$ millions: mean values of y^+ for image points with respect to the near-wall resolution.

-	IBM5 F42	IBM6 F42	IBM7 F42	IBM8 F42	BF-WM
y^+	134	123	117	115	124

but the friction coefficient was under-predicted for coarser meshes and became over-predicted for finer meshes. Spurious oscillations appeared, similar to what was observed previously for F0 treatment. The same general behaviour has also been reproduced ever since with the F1 and F2 treatments.

Here, whatever the mesh refinement, FAST IBM F42 correctly reproduces the pressure coefficient and the friction coefficient distributions, as displayed in Fig. 11. The obtained solutions are in agreement with the reference FAST BF solution. The wall model slightly influences the friction coefficient, in particular near the leading edge.

Table 4 also proves that the new algorithm enables to locate image points according to the desired $y_{target}^+ = 100$.

Fig. 12 displays longitudinal velocity profiles at several positions along the chord for FAST IBM7 F42 simulation, showing that these velocity profiles obtained with the new IBM treatment are in good agreement with the FAST BF-WM solution, where the body-fitted mesh is used in combination with the same wall model as for the IBM approach. Yet, variations between these two wall-modelled solutions and the FAST BF reference appear at the beginning of the airfoil, which could explain the small discrepancies observed on the skin friction coefficient previously.

5.1.3. Influence of the wall model

Following the previous observations, Fig. 13 compiles the information on the impact of the wall model on the skin friction prediction. To do so, various y_{target}^+ are studied with the body-fitted mesh, first for different boundary layer subregions (Fig. 13-(a)), then inside the log sub-layer (Fig. 13-(b)).

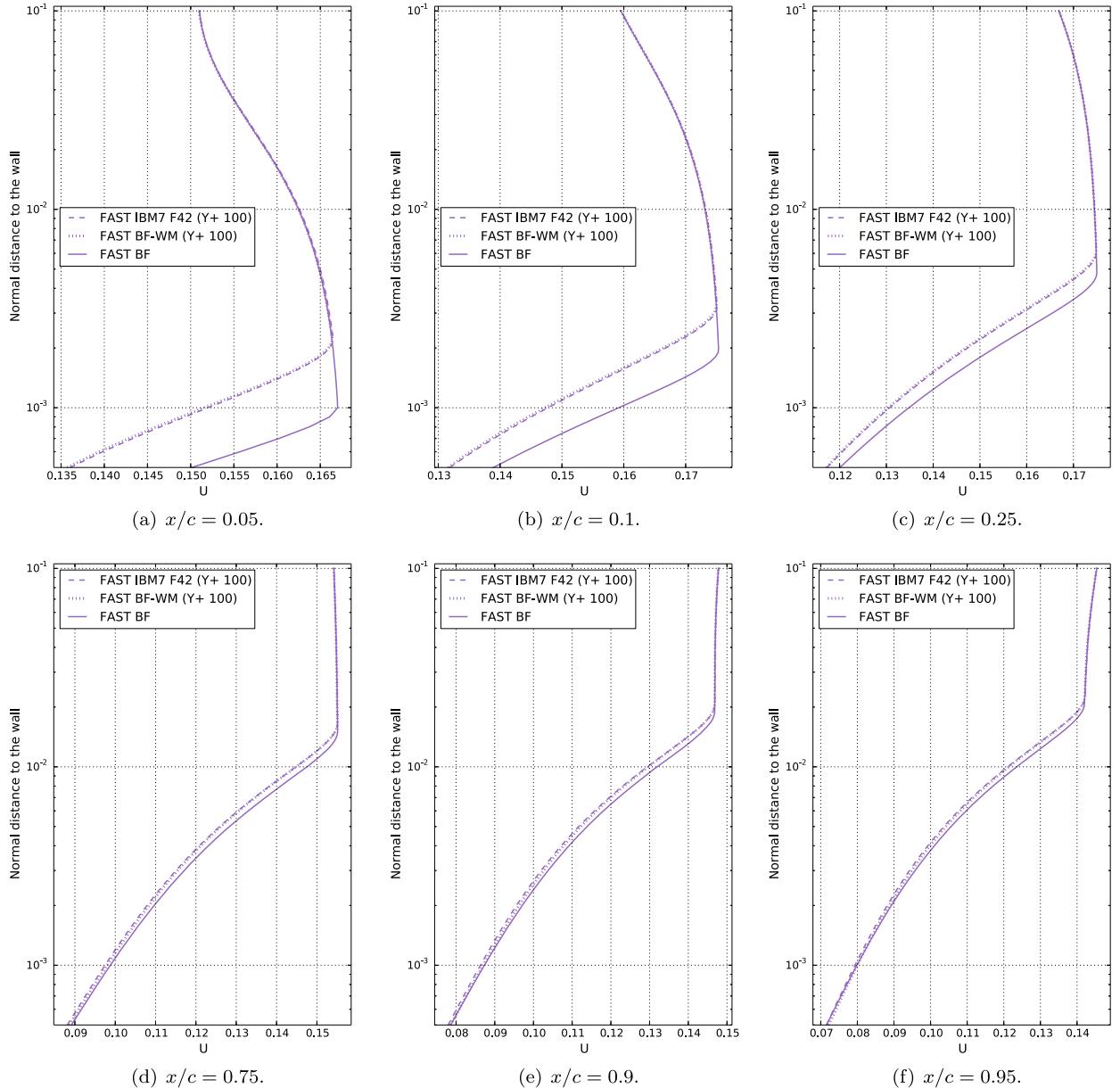


Fig. 12. NACA0012 airfoil, $M_\infty = 0.15$, $Re = 6$ millions: x-velocity profiles.

It can be observed that the skin friction distribution depends on the sublayer where the IB image points fall. This confirms that the IB target and image points should both lay in the same given sublayer and can explain the spurious oscillations previously encountered.

First, in the inner subregion ($y^+ = 1$), that is unattainable for a Cartesian mesh, the wall model is capable of accurately predict the close wall physics. Eventually, bringing the IB target points closer to the wall increases the quality of the solution that correctly converges toward the FAST BF reference.

Then, in the buffer zone ($y^+ = 10$), the forcing points fall in an area that is not well defined by the wall model and the solution eventually differs from the reference.

Finally, in the log sub-layer ($y^+ \in [50, 300]$), the solution is again well predicted by the wall model. However, as noticed before, solutions slightly vary near the leading edge from one another depending on the y_{target}^+ . According to the Fig. 12, a reasonable hypothesis would be that the boundary layer has yet to be fully turbulent and remains laminar near the leading edge. Hence, a new wall model might be investigated in the future to obtain the correct modelling in this area.

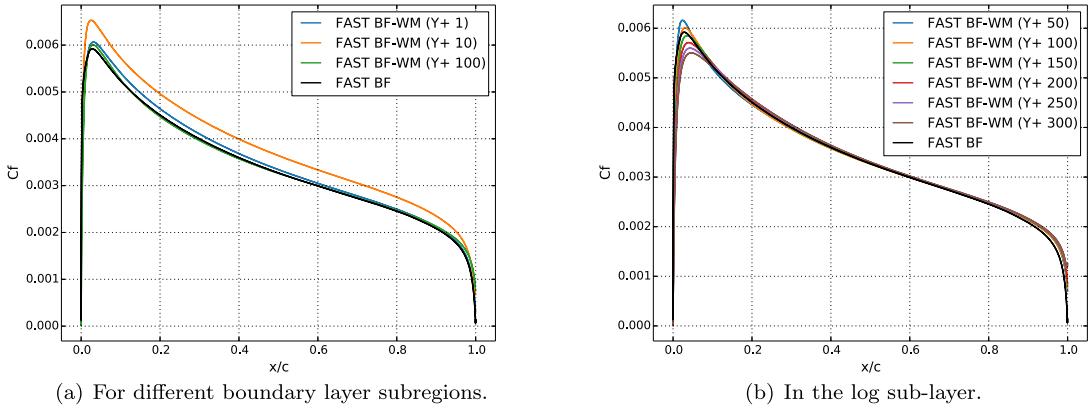


Fig. 13. NACA0012 airfoil, $M_\infty = 0.15$, $Re = 6$ millions: effect of the wall model on the skin friction distribution on a body-fitted mesh for different modelling heights.

Table 5
Cartesian mesh characteristics for the RAE2822 airfoil test-case.

-	IBM6	IBM7	IBM8	BF (body-fitted)
Minimum spacing h_{min}/c	0.0001	0.00005	0.00002	10^{-6}
Number of points (millions)	7.2	15.9	36.9	0.137

5.1.4. Influence of the angle of attack

Figs. 14 and 15 summarize the behaviour of the improved IBM treatment when increasing the angle of attack, the other flow and solver parameters being unchanged. Without any additional treatment, solution FAST IBM7 F42 demonstrates the robustness of the method as the solution is still accurate as the angle of attack increases. Yet, as already highlighted, a slight difference is observed near the leading edge for the friction coefficient. However, since FAST IBM7 F42 and FAST BF-WM match for all the angles of attacks that are considered here, it can be assessed that this is due to the wall model, as mentioned earlier, rather than to the Cartesian mesh or the stair-step IBM.

5.2. Simulation of the transonic turbulent flow around a RAE2822 airfoil

The second test-case is the simulation of the transonic turbulent flow around a RAE2822 airfoil at freestream Mach number $M_\infty = 0.729$, an angle of attack α of 2.79° and a Reynolds number based on the chord c equal to $Re_c = 6.5$ millions. This test-case aims at evaluating the accuracy of the method in the case of an interaction between the boundary layer and a shock, that is located at mid-span of the airfoil on the suction side.

The Cartesian mesh is automatically generated with an extent of $20c$ and a uniform near-wall spacing. Fig. 16 presents a global view around the airfoil and close-up views in the vicinity of the leading edge, the suction side and the trailing edge, showing more clearly the distribution of the refinement levels in the near-wall region. The method is evaluated for different near-wall spacings, for which the mesh characteristics are summarized in Table 5.

Similarly to the previous validation case, the steady-state RANS equations are solved using FASTS solver. A backward Euler implicit scheme is applied in combination with the Roe-MUSCL scheme for the spatial discretization. The standard Spalart-Allmaras turbulence model is used and Musker's wall model is applied for the reconstruction of the velocity at IB target points.

Fig. 17 compares the iso-contours of the Mach number for the IBM wall-modelled simulation and the wall-resolved body-fitted solution, showing a good agreement between both results, even if the shock appears thinner for the IBM simulation, as in its vicinity, the mesh is finer than the body-fitted mesh.

Fig. 18 demonstrates the mesh convergence of the method to the reference body-fitted solution. The shock location is better predicted as the near-wall spacing increases. At convergence, the skin pressure distribution is identical to that of a body-fitted solution where the boundary layer is modelled by a wall function. The skin friction distribution, depicted in Fig. 18-(b), is overall underestimated compared to the body-fitted solution where the near-wall region is resolved by the mesh. As the skin friction is identical to that of a body-fitted wall-modelled solution, these discrepancies are probably linked to the wall model, since it does not take into account the pressure gradient and convection in the vicinity of the shock, as discussed by Capizzano [14].

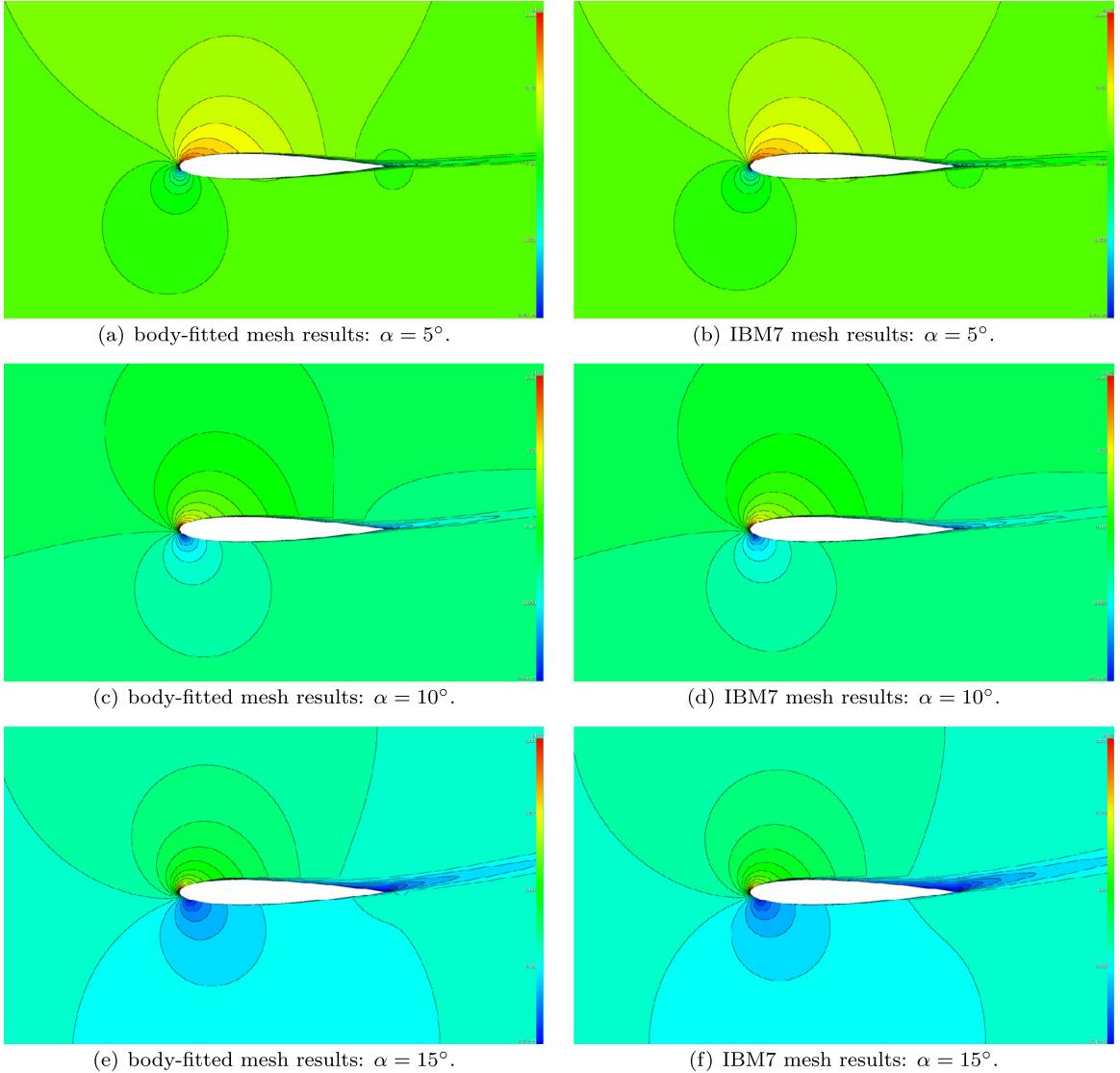


Fig. 14. NACA0012 airfoil, $M_\infty = 0.15$, $Re = 6$ millions: influence of the angle of attack α on the Mach flowfield.

5.3. Simulation of the transonic turbulent flow around the ONERA M6 wing

The final test is the attached transonic turbulent flow around a ONERA M6 wing at freestream Mach number $M_\infty = 0.839$, angle of attack $\alpha = 3.06^\circ$ and Reynolds number $Re_c = 11.72$ millions. This test case has been originally described in [39] by Schmitt and Charpin in 1979 and is still widely popular in the CFD community to evaluate turbulent models in CFD solvers [40].

The Cartesian mesh is automatically generated with an extent of $15c$. It is composed of 1.7 billion points, with a near-wall spacing of $h_w = 4 \times 10^{-4}c$ around the wing and a local refinement of $h_{LE} = 2 \times 10^{-4}c$ in the vicinity of the leading edge. A slice at spanwise section $y/b = 0.55$ of the final mesh can be found in Fig. 19, where the change of refinement level along the chord can be seen.

The regular surface mesh of the ONERA M6 wing is also modified with INRIA surface remeshing tool MMGS [41] such that the discretization of the surface is compliant with the near-wall spacing of the Cartesian mesh, which is usually necessary to prevent from solution faceting due to the projection of IB target points onto the triangular mesh which can be observed if the triangles are roughly ten times larger and more than the near-wall cells.

Here, IBM simulations using the improved method are performed with a target $y_{target}^+ = 1000$ for the image points. Actually, the near-wall cell spacing h_{min} must be in agreement with y_{target}^+ , such that the first calculated points above IB target points are compliant with the cell spacing required for a body-conformal mesh at the same distance y_{target}^+ . In order to prevent a huge number of mesh points, the near-wall spacing around the wing is set to $h_w = 4 \times 10^{-4}c$, corresponding

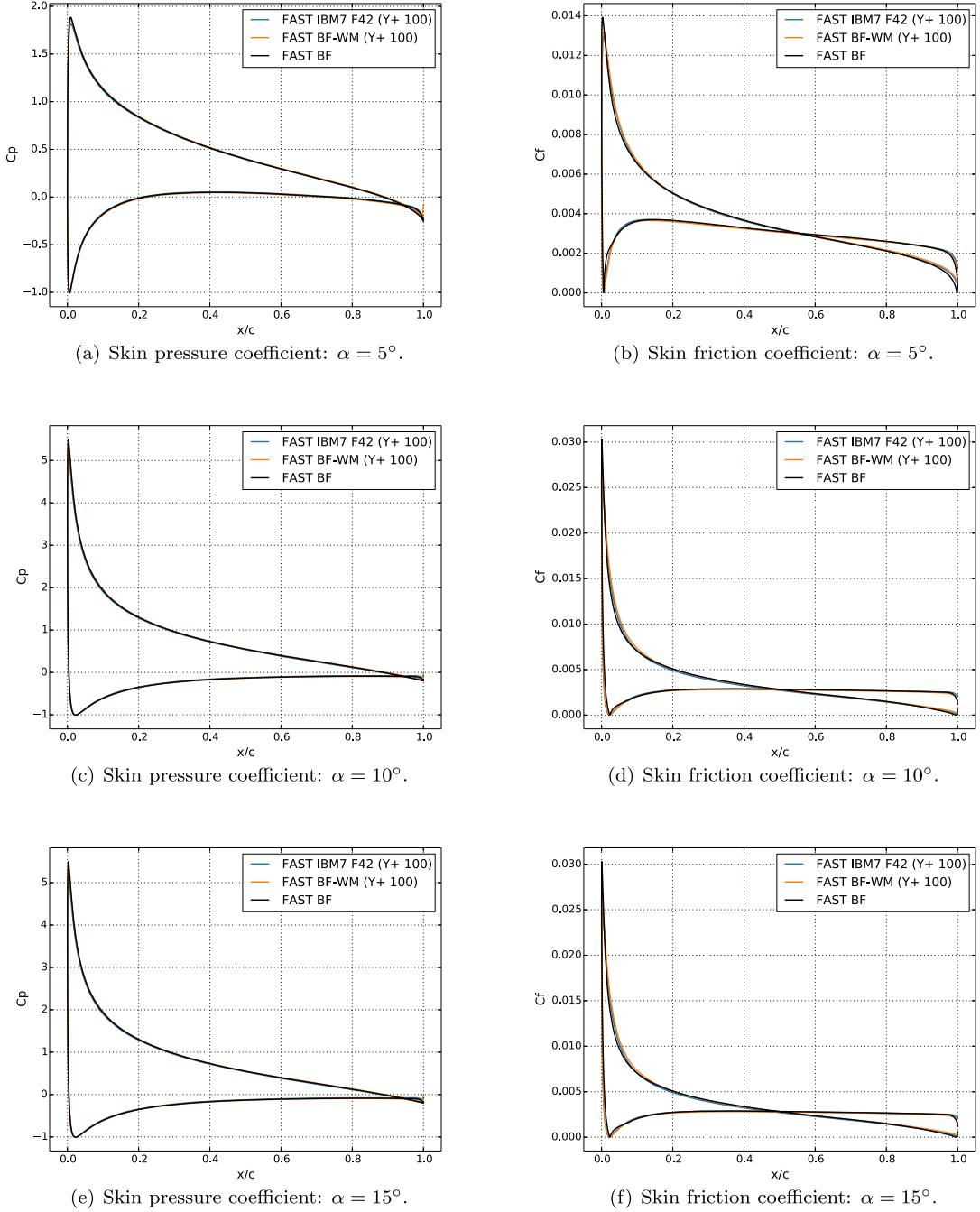


Fig. 15. NACA0012 airfoil, $M_\infty = 0.15$, $Re = 6$ millions: influence of the angle of attack α .

to $y_{target}^+ = 1000$. Note that the ideal y_{target}^+ is imposed by the coarsest near-wall spacing h_w and not h_{LE} in the vicinity of the trailing edge. As already observed on bidimensional test-cases, this value of y_{target}^+ should have been decreased for a better prediction of the skin friction in this region. But imposing a value $y_{target}^+ = 500$ would eventually lead to spurious oscillations in the near-field around the rest of the wing as the discretization is not sufficient to compute accurately the gradients given that distance y_{target}^+ .

The simulations have been performed with the same solver parameters as for the RAE2822 test case, on 20×28 Intel cores of the ONERA SATOR cluster.

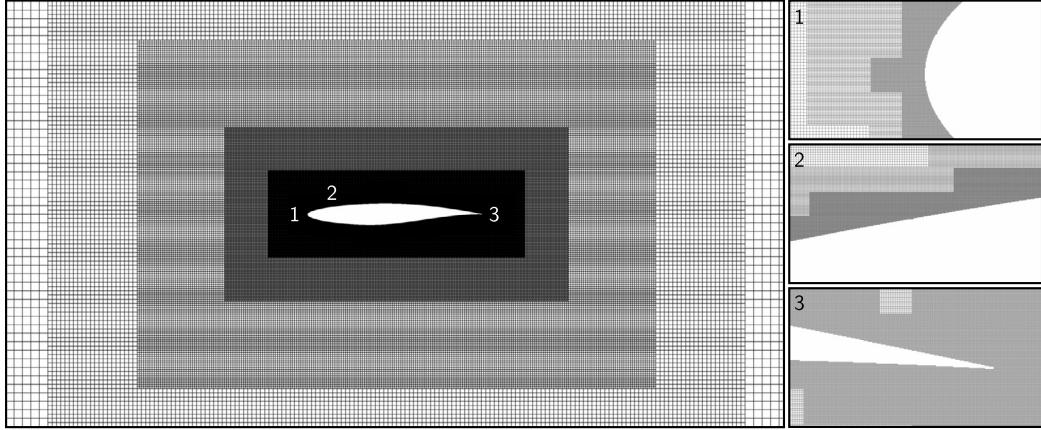
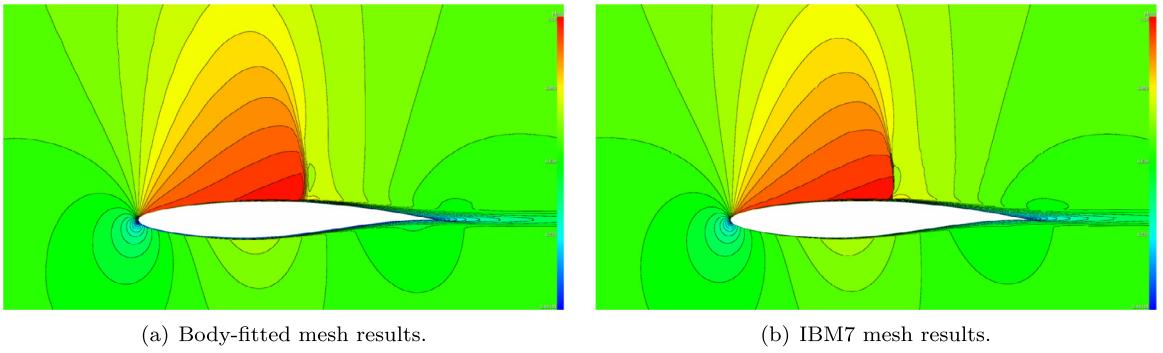


Fig. 16. Views of the IBM Cartesian mesh around the RAE2822 airfoil.



(a) Body-fitted mesh results.

(b) IBM7 mesh results.

Fig. 17. RAE2822 airfoil: comparison of iso-contours of the Mach number between wall-modelled IBM and wall-resolved body-fitted solutions.

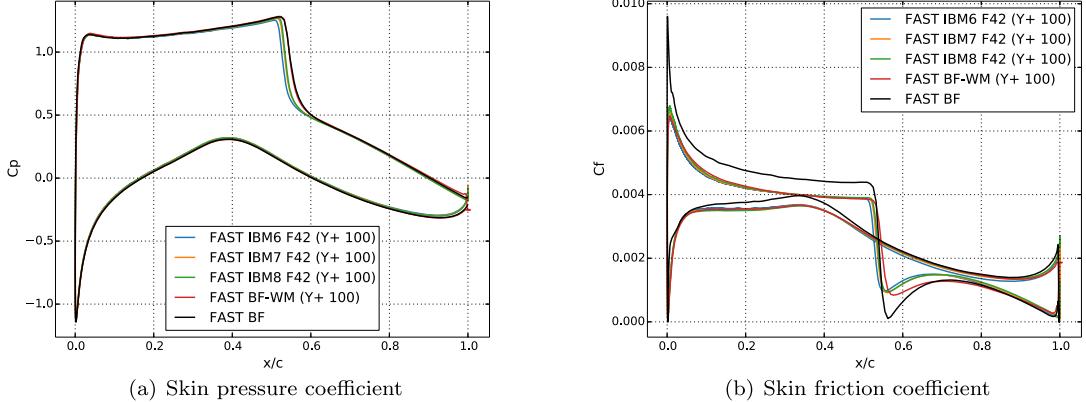


Fig. 18. RAE2822 airfoil: influence of the near-wall resolution.

Final IBM results are compared with experimental data from [39] and a solution obtained on a fine wall-resolved conformal mesh proposed on the NASA Turbulence Model Resources website using the CFL3D solver and the standard Spalart-Allmaras turbulence model.

Similarly to the previous bidimensional cases, no extra cost was observed during the pre-processing when using the improved method. However, for the present tridimensional case, a slight increase of 2% of the CPU time was observed with the F42 algorithm in comparison with the F1 algorithm.

Finally, it has to be noted that the equation (12), used to assess the modelling height with the F42 algorithm, still applies in a satisfactory manner for tridimensional cases. Hence, no further developments were undertaken to extend the validity of our method from 2D to 3D simulations.

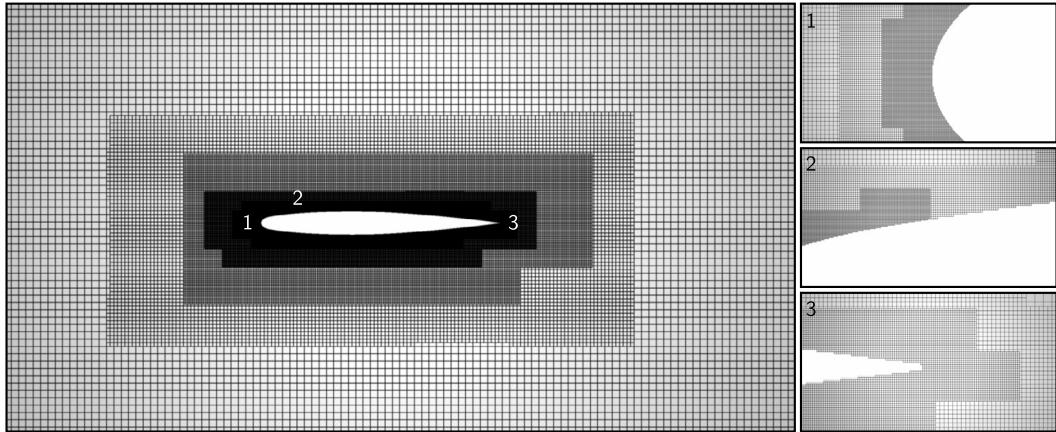


Fig. 19. Views of the IBM Cartesian mesh around the ONERA M6 wing at $y/b = 0.55$.

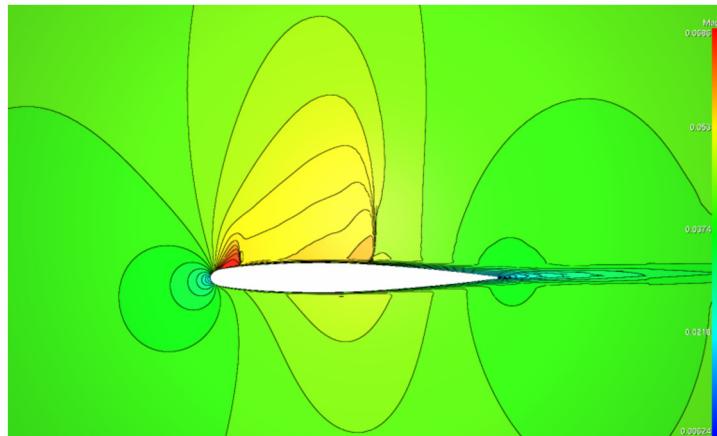


Fig. 20. ONERA M6 wing, $M_\infty = 0.839$, $Re = 11.72$ millions: Mach number flowfield at $y/b = 0.55$.

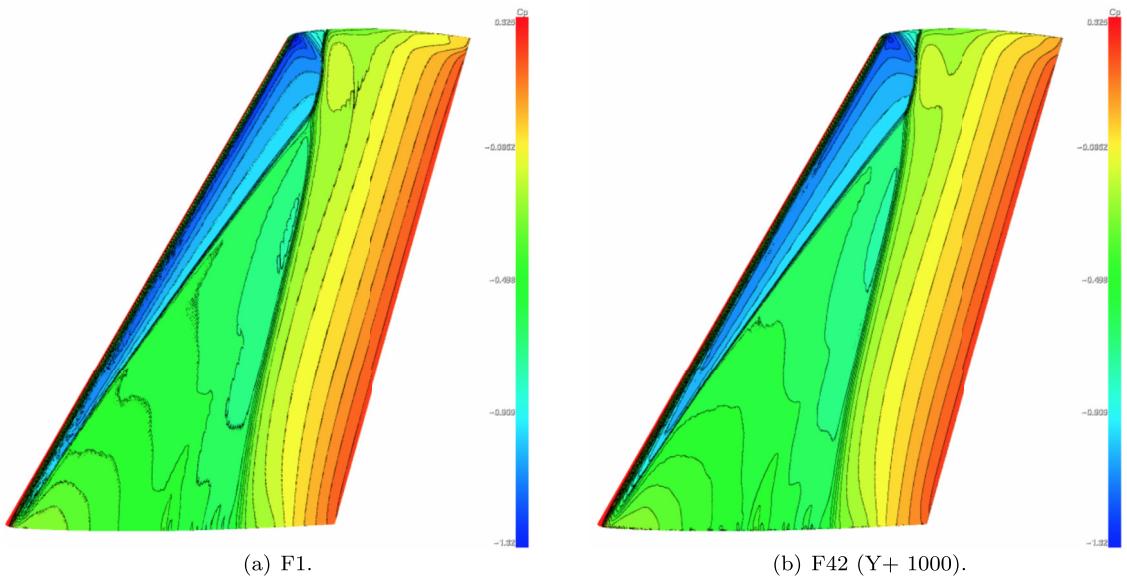


Fig. 21. ONERA M6 wing, $M_\infty = 0.839$, $Re = 11.72$ millions: comparison of C_p distributions between two wall-modelled IBM.

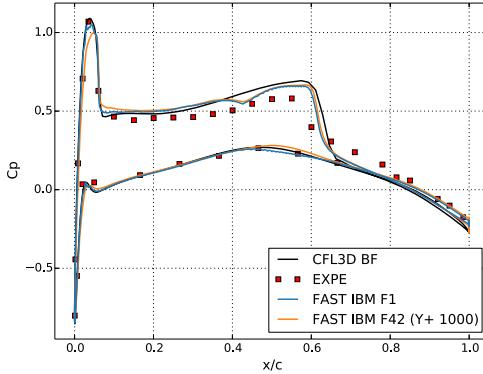
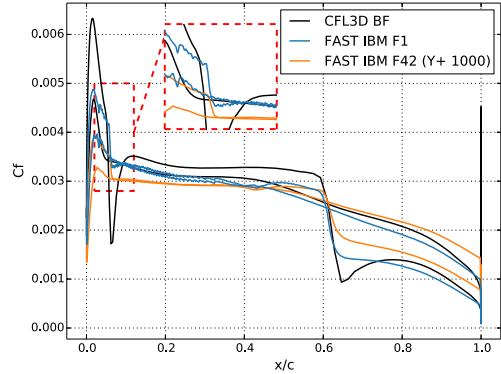
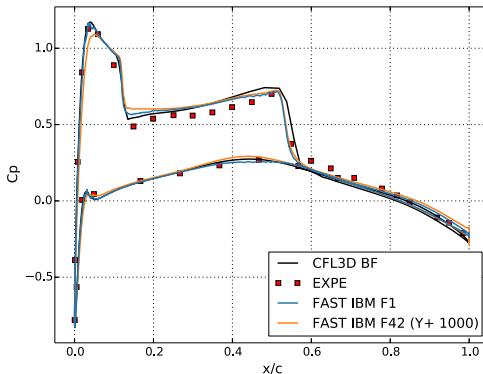
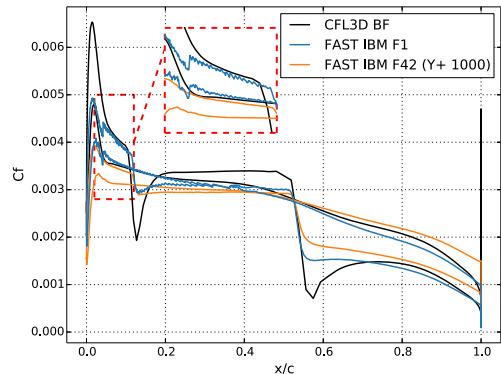
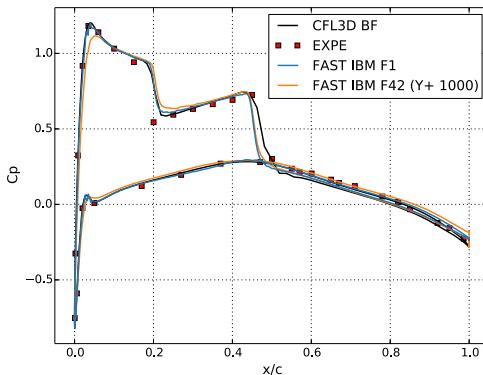
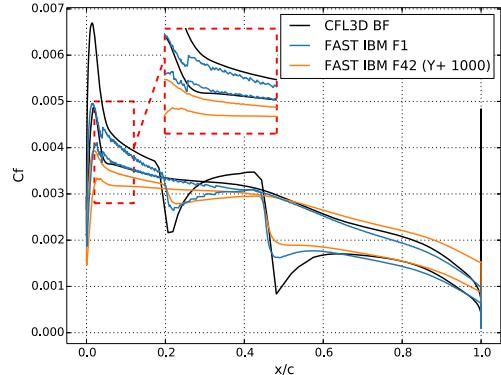
(a) Skin pressure coefficient: $y/b = 0.2$.(b) Skin friction coefficient: $y/b = 0.2$.(c) Skin pressure coefficient: $y/b = 0.44$.(d) Skin friction coefficient: $y/b = 0.44$.(e) Skin pressure coefficient: $y/b = 0.65$.(f) Skin friction coefficient: $y/b = 0.65$.

Fig. 22. ONERA M6 Wing, $M_\infty = 0.84$, $Re = 11.72$ millions: comparison of skin pressure and skin friction distributions at different spans on the wing (improved treatment versus former treatment and body-fitted solution).

Fig. 20 displays the isocontours of the Mach number at a constant spanwise section $y/b = 0.55$, showing no oscillations near the obstacles.

Fig. 21 compares the skin pressure coefficient for the F1 algorithm (Fig. 21-(a)) and the F42 algorithm (Fig. 21-(b)), showing that the improved IBM treatment captures well the λ -shaped shock on the suction side of the wing without the spurious oscillations observed with former F1 algorithm.

Figs. 22 and 23 display the skin pressure and skin friction distributions for different sections of the ONERA M6 wing.

The FAST IBM F42 and FAST IBM F1 solutions show similar trends for the skin pressure distribution and compare well with both numerical and experimental reference solutions. The shock location is also well predicted for all the sections.

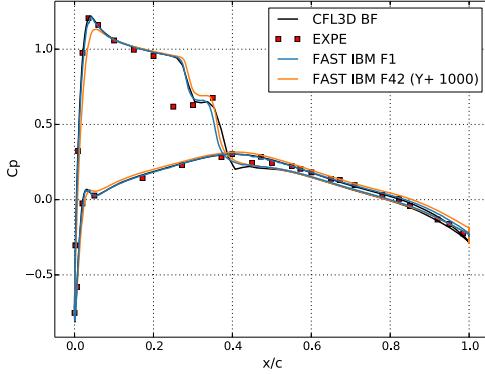
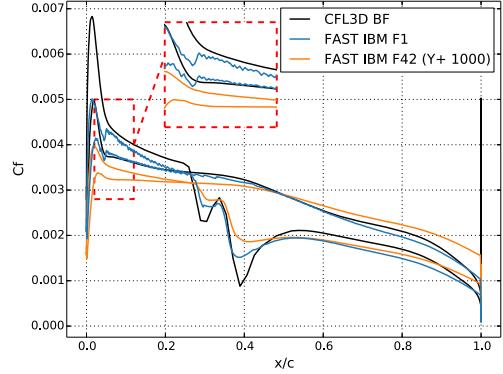
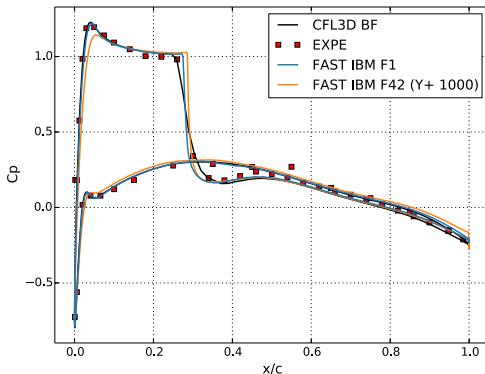
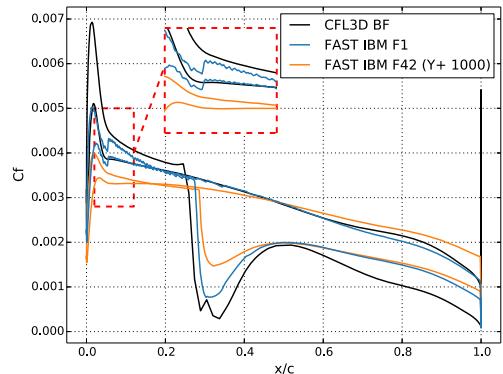
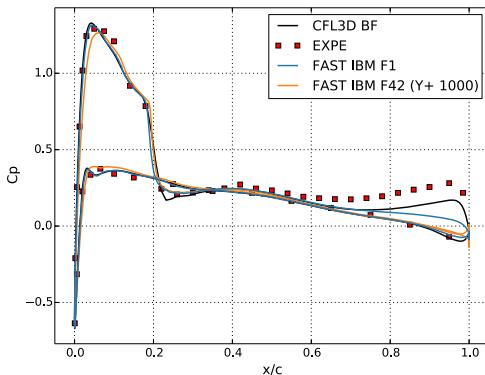
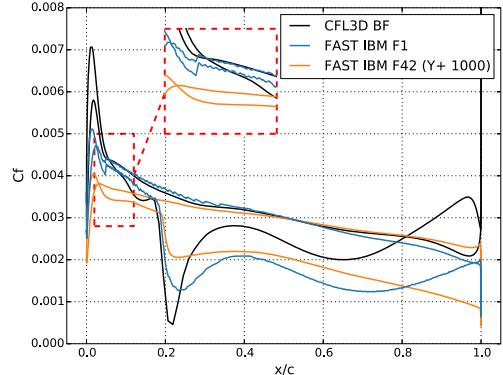
(a) Skin pressure coefficient: $y/b = 0.8$.(b) Skin friction coefficient: $y/b = 0.8$.(c) Skin pressure coefficient: $y/b = 0.9$.(d) Skin friction coefficient: $y/b = 0.9$.(e) Skin pressure coefficient: $y/b = 0.99$.(f) Skin friction coefficient: $y/b = 0.99$.

Fig. 23. ONERA M6 Wing, $M_\infty = 0.84$, $Re = 11.72$ millions: comparison of skin pressure and skin friction distributions at different spans on the wing (improved treatment versus former treatment and body-fitted solution).

Besides, the improved method enables to remove the oscillations appearing on the sections close to the wing root for the former method, as displayed in Figs. 22-(a,c,e).

As previously observed on the RAE2822 test case, the skin friction distribution is still underpredicted for the IBM solutions compared to the CFL3D BF reference solution, especially on the suction side and near the leading edge. Once again, this underestimation is likely due to the limitations of our wall modelling for transonic flows, as stated in section 5.2.

A slightly better prediction of the peaks near the leading edge and the shock is observed in favour of the F1 algorithm, as its actual y_{target}^+ is lower than 1000, but the improved method provides smoother results, as it suppresses all spurious oscillations as well as the deviations and perturbations observed on the FAST IBM F1 solutions in the vicinity of the change of refinement level.

6. Conclusions

An improved immersed boundary method on Cartesian grids for RANS simulations of turbulent compressible flows has been proposed. Smooth pressure and skin friction coefficients have been obtained and compared with body-fitted approaches. In the subsonic regime, results on the NACA0012 airfoil demonstrate that the grid convergence to the body-fitted solution is reached for this flow regime and that the method is not sensitive to the angle of attack. In the transonic flow regime, skin pressure distribution is well captured and compares well to the body-fitted solution. The skin friction distribution is under-estimated in comparison with the wall-resolved body-fitted solution but compares to the wall-modelled body-fitted solution, showing that the stair-step distribution is no longer responsible for the discrepancies but the wall model. The ONERA M6 wing test-case highlights the extension to three-dimensional cases of the present approach. It can be noticed that the present approach is neither dependent from the wall model nor from the turbulence model. Future work will consist in improving the wall model to take into account for laminar regions and pressure gradients within the IBM treatment.

CRediT authorship contribution statement

Benjamin Constant: Conceptualization, Software, Validation, Writing – original draft. **Séphanie Péron:** Conceptualization, Writing – original draft. **Héloïse Beaugendre:** Supervision, Writing – review & editing. **Christophe Benoit:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors would like to acknowledge Marc Terracol for our fruitful discussions, Ivan Mary and Thomas Renaud from ONERA for their helpful remarks.

References

- [1] C. Peskin, Flow patterns around heart valves: a numerical method, *J. Comput. Phys.* 10 (2) (1972) 252–271.
- [2] C. Peskin, The immersed boundary method, *Acta Numer.* 11 (2002) 479–517.
- [3] R. Mittal, G. Iaccarino, Immersed boundary methods, *Annu. Rev. Fluid Mech.* 37 (2005) 239–261.
- [4] R. Beyer, R. LeVeque, Analysis of a one-dimensional model for the immersed boundary method, *SIAM J. Numer. Anal.* 29 (2) (1992) 332–364.
- [5] P. Angot, C.-H. Bruneau, P. Fabrie, A penalization method to take into account obstacles in incompressible viscous flows, *Numer. Math.* 81 (1999) 497–520.
- [6] M.-C. Lai, C.S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, *J. Comput. Phys.* 160 (2) (2000) 705–719.
- [7] K. Nakahashi, Immersed boundary method for compressible Euler equations in the Building-Cube Method, *AIAA paper* 2011-3386, 2011.
- [8] Y.-H. Tseng, J. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, *J. Comput. Phys.* 192 (2) (2003) 593–623.
- [9] W. Zhu, T. Behrens, W. Shen, J. Sørensen, Hybrid immersed boundary method for airfoils with a trailing-edge flap, *AIAA J.* 51 (1) (2012) 30–41.
- [10] W. Coirier, K. Powell, Solution-adaptive Cartesian cell approach for viscous and inviscid flows, *AIAA J.* 34 (5) (1996) 938–945.
- [11] M. Berger, M. Aftosmis, Progress towards a Cartesian cut-cell method for viscous compressible flow, *AIAA paper* 2012-1301, 2012.
- [12] M. Harada, Y. Tamaki, Y. Takahashi, T. Imamura, Simple and robust cut-cell method for high-Reynolds-number-flow simulation on Cartesian grids, *AIAA J.* 55 (8) (2017) 2833–2841.
- [13] C. Brehm, M. Barad, C. Kiris, Open rotor computational aeroacoustic analysis with an immersed boundary method, in: *54th AIAA Aerospace Sciences Meeting*, 2016, p. 0815.
- [14] F. Capizzano, Turbulent wall model for immersed boundary methods, *AIAA J.* 49 (11) (2011) 2367–2381.
- [15] M. Berger, M. Aftosmis, An ODE-based wall model for turbulent flow simulations, *AIAA J.* (2017) 1–15.
- [16] S. Péron, C. Benoit, T. Renaud, I. Mary, An immersed boundary method on Cartesian adaptive grids for the simulation of compressible flows around arbitrary geometries, *Eng. Comput.* (2020) 1–19.
- [17] S. Wilhelm, J. Jacob, P. Sagaut, An explicit power-law-based wall model for lattice Boltzmann method–Reynolds-averaged numerical simulations of the flow around airfoils, *Phys. Fluids* 30 (6) (2018) 065111.
- [18] G. Kalitzin, G. Iaccarino, Turbulence modeling in an immersed-boundary RANS method, *CTR Annu. Briefs* (2002) 415–426.
- [19] T. Pu, C. Zhou, An immersed boundary/wall modeling method for rans simulation of compressible turbulent flows, *Int. J. Numer. Methods Fluids* 87 (5) (2018) 217–238.
- [20] Y. Tamaki, M. Harada, T. Imamura, Near-wall modification of Spalart–Allmaras turbulence model for immersed boundary method, *AIAA J.* (2017) 3027–3039.
- [21] P.R. Spalart, S.R. Allmaras, A one-equation turbulence model for aerodynamic flows, *AIAA J.* 94 (1992).
- [22] I. Mary, Flexible Aerodynamic Solver Technology in an HPC Environment, Maison de la Simulation Seminars, 2016, [http://www.maisondelasimulation.fr/seminar/data/\\$201611\\$_slides_\\$1\\$.ppt](http://www.maisondelasimulation.fr/seminar/data/201611_slides_1.ppt).
- [23] <https://w3.onera.fr/FAST>.
- [24] J. Dandois, I. Mary, V. Brion, Large-eddy simulation of laminar transonic buffet, *J. Fluid Mech.* 850 (2018) 156–178.
- [25] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, *J. Comput. Phys.* 43 (2) (1981) 357–372.
- [26] I. Mary, P. Sagaut, Large Eddy simulation of flow around an airfoil near stall, *AIAA J.* 40 (6) (2002) 1139–1145.

- [27] J. Edwards, M.-S. Liou, Low-diffusion flux-splitting methods for flows at all speeds, *AIAA J.* 36 (9) (1998) 1610–1617.
- [28] A. Jameson, S. Yoon, Lower-upper implicit schemes with multiple grids for the Euler equations, *AIAA J.* 25 (7) (1987) 929–935.
- [29] T. Coakley, Implicit upwind methods for the compressible Navier-Stokes equations, *AIAA J.* 23 (3) (1985) 374–380.
- [30] S. Péron, C. Benoit, Automatic off-body overset adaptive Cartesian mesh method based on an octree approach, *J. Comput. Phys.* 232 (1) (2013) 153–173.
- [31] C. Benoit, S. Péron, S. Landier, Cassiopee: a CFD pre- and post-processing tool, *Aerosp. Sci. Technol.* 45 (2015) 272–283.
- [32] S. Péron, T. Renaud, M. Terracol, C. Benoit, I. Mary, An immersed boundary method for preliminary design aerodynamic studies of complex configurations, in: *23rd AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2017-3623, 2017.
- [33] F. Capizzano, Coupling a wall diffusion model with an immersed boundary technique, *AIAA J.* 54 (2) (2016) 728–734.
- [34] A. Musker, Explicit expression for the smooth wall velocity distribution in a turbulent boundary layer, *AIAA J.* 17 (6) (1979) 655–657.
- [35] P. Lancaster, K. Salkauskas, Surfaces generated by moving least squares methods, *Math. Comput.* 37 (155) (1981) 141–158.
- [36] A. Nealen, An as-short-as-possible introduction to the least squares, weighted least squares and moving least squares methods for scattered data approximation and interpolation, <http://www.nealen.com/projects>, 2004.
- [37] Z.-Q. Cheng, Y.-Z. Wang, B. Li, K. Xu, G. Dang, S.-Y. Jin, A survey of methods for moving least squares surfaces, in: *Proceedings of the Fifth Eurographics/IEEE VGTC Conference on Point-Based Graphics*, 2008, pp. 9–23.
- [38] T. Renaud, C. Benoit, S. Péron, I. Mary, N. Alferez, Validation of an immersed boundary method for compressible flows, in: *AIAA Scitech 2019 Forum*, AIAA paper 2019-2179.
- [39] V. Schmitt, F. Charpin, Pressure distribution on the ONERA-M6 wing at transonic Mach number, in: *AGARD Report AR 138, Experimental Data Base for Computer Program Assessment*, 1979.
- [40] J. Mayeur, A. Dumont, D. Destarac, V. Gleize, Reynolds-averaged Navier-Stokes simulations on NACA0012 and ONERA-M6 wing with the ONERA elsa solver, *AIAA J.* 54 (9) (2016) 2671–2687.
- [41] C. Dapogny, C. Dobrzynski, P. Frey, Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems, *J. Comput. Phys.* 262 (2014) 358–378.