

On the suppression of spurious pressure oscillations in immersed boundary methods with unstructured grids

Diogo Matias Coimbra Martins
diogomatiasmartins@ist.utl.pt

Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

November 2016

Abstract

In this work a new immersed boundary (IB) method is presented, which features a conservative fluid domain characterization together with a flexible least-squares flow-reconstruction to emulate the body presence. The method decreases spurious oscillations in the pressure field and consequently in the body force calculations, which are a common issue in several immersed boundary methods, by applying a continuity constraint in the least-squares interpolation and guaranteeing that adjacent interpolation polynomials are continuous between each other. This approach strictly enforces continuity in the flow reconstruction domain, reducing time discontinuities caused by the boundary conditions applied at the immersed boundary.

Additionally, the advantages an unstructured IB method can offer are evident when considering how the high flexibility of unstructured grids can be used with adaptive algorithms or to generate precisely locally refined grids and how the meshing of complex fixed domains in conjunction with immersed moving complex bodies is extremely useful. The proposed method is therefore applied to unstructured grids and the effects of grid type are evaluated.

Several test cases are shown to validate the method and demonstrate its capabilities. The method is shown to be second order accurate independently of grid type and to not introduce significant error. The SFO suppression properties of the method are also shown to be superior to those of available corrections and it is shown that polyhedral grids produce smaller oscillations than Cartesian or triangular ones.

Keywords: immersed boundary IB, spurious force oscillations SFO, unstructured grids, least-squares

1. Introduction

Immersed boundary (IB) methods have, since the pioneering work of [1], been used for a multitude of flow problems and demonstrated great capabilities and flexibility.

When solving flow problems numerically the meshing process can sometimes be as computationally expensive as the fluid simulation itself and IB methods greatly simplify this process because they do not require body-conformal grids. This is particularly useful for moving or deforming bodies in a flow, where IB methods don't require remeshing of the domain at each time step.

A problem that has been often reported in IB simulations of moving boundaries is the existence of oscillations in pressure, see e.g. [2–6], that result in spurious force oscillations, or SFO's. In continuous forcing approaches this effect can be significantly reduced by simply increasing the size of the forcing stencil [7], which smoothens discontinuities over the computational grid. However no remedy has been found to eliminate them when using discrete forcing

methods.

The origin of these SFOs is associated with a spatial discontinuity in the pressure field and a temporal discontinuity in the velocity field. This occurs when cells in the fluid domain are moved into the solid domain (fresh cells) or vice-versa (dead cells) between two time steps [5]. SFOs decrease when the grid size is decreased or the time step is increased. A method developed by Kim and Choi [8] consisting of mass source/sinks enforcing continuity has been successively implemented to dampen pressure oscillations. Seo and Mittal [4] suppressed SFOs with a hybrid ghost/cut-cell method for the continuity equation, which more strictly enforced continuity. Luo et al. [6] were capable of suppressing them by smoothing the treatment of the immersed boundary over a few grid cells.

The great majority of IB simulations use regular Cartesian grids, which are extremely popular for being very simple to generate and implement in parallel and because together with an IB method can be used for complex body geometries. However several

arguments can be made in favor of using IB methods in curvilinear and unstructured grids [9–13]. In problems where a fixed complex geometry contains moving bodies, being able to use a body conformal grid for the fixed domain, together with an IB method for the moving body (thereby removing the need for continuous remeshing) is extremely advantageous. Sotiropoulos has simulated heart flow problems [9, 14, 15] using a fixed curvilinear grid together with an immersed boundary method for moving bodies to great success. Furthermore when using immersed boundary methods together with precise local refinement the use of non-Cartesian grids offers a lot of advantages.

2. Numerical Methods

The numerical method for the bulk flow used in the present work has been previously developed in the LASEF department and has been thoroughly verified in several flow conditions [16, 17]. A detailed description of the numerical method can be found in the referenced works, and both thesis from Albuquerque [18] and Magalhães [19].

The finite volume method solves the fluid equations implicitly in a collocated grid arrangement with second order accuracy. Both the SIMPLE [20] and PISO [21] algorithms have been used to solve the pressure coupling. The PISO algorithm offered faster convergence because it does not require under-relaxation for the pressure correction, drastically reducing the number of iterations for each time step, however some numerical stability problems exist, and the SIMPLE algorithm, being more robust, was used in all simulations with unstructured grids. Both algorithms have been explained in countless works since the original publications, so no explanation is offered in the current work.

2.1. Governing Equations

All fluid simulations in this work share the hypothesis of incompressible, isothermal and Newtonian fluid, in bi-dimensional flow conditions. When these hypothesis are taken into account, the simplified version of the Navier-Stokes equations for fluid flow reads as:

$$\nabla \cdot \mathbf{u} = 0 \quad (1)$$

$$\frac{\partial(\mathbf{u})}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = -\frac{1}{\rho} \nabla(p) + \nabla \cdot (\nu \nabla \mathbf{u}) \quad (2)$$

2.2. Unstructured Grids

In the current work several unstructured grids were used for the different test cases. The process of generating unstructured grids with good grid quality is a whole field of study in itself, which deserves its own investigation. Grid quality relates to concepts such as warp angle, also called non-orthogonality, skewness and volume ratio which were described by Juretic [22].

To generate the unstructured grids with the best possible grid-quality the distMesh [23] software was used. This open-source software generates triangular grids with good grid-quality. A dual-meshing program was developed to transform triangular to polyhedral grids. First a triangular grid with roughly half the desired cell-size and the desired local refinement is generated. Then polyhedrals are created by uniting each triangle’s centroids, producing a polyhedral grid with good grid quality.

In the current work the cell reference length h is used to compare cells of different topologies.

$$h = \sqrt{Cell_{area}} \quad (3)$$

2.3. Discretization Schemes

Due to the unstructured nature of the solver, special discretization schemes are used for the convective and diffusive terms, as well as for the cell centered gradient computations. The used schemes are well suited for unstructured grids and allow the method to achieve second order accuracy for different grid types over a large range of grid quality values.

The used convective scheme uses information from the two cells sharing the current face, namely the cell value of ϕ and also the cell centered gradient, together with local geometric information.

$$\phi_f = \eta_{TRI} \phi_{P_0} + (1 - \eta_{TRI}) \phi_{P_1} + \underbrace{(\mathbf{f} - \mathbf{P}_0 - \eta_{TRI} \mathbf{d}) \cdot (\overline{\nabla \phi})_f}_{\text{skewness correction}} \quad (4)$$

$$\eta_{TRI} = \frac{(\mathbf{f} - \mathbf{P}_0) \cdot \mathbf{d}}{\mathbf{d} \cdot \mathbf{d}} \quad (5)$$

where ϕ_{P_0} and ϕ_{P_1} are the cell values of ϕ at the two cells, η_{TRI} is a blending factor, \mathbf{P}_0 and \mathbf{f} are the cell P_0 ’s and face’s centroids respectively, and \mathbf{d} the distance vector between the two cell centroids. The overlined term $(\overline{\nabla \phi})_f$ is computed by averaging the two cell’s cell-centered gradients. The blending factor η_{TRI} corresponds to a linear interpolation to the point in the line between \mathbf{P}_0 and \mathbf{P}_1 that’s geometrically closer to \mathbf{f} . The term identified as skewness correction was introduced by Juretic [22] and is required to provide robustness in unstructured grids.

The cell centered gradients are calculated with the Gauss theorem making use of the mentioned convective scheme. Because the convective and gradient schemes depend on each other, the gradient scheme is computed in three iterations and in the first one the convective scheme is used without the skewness correction term, since the correction term depends on the cell-centered gradient.

The used diffusive scheme was first proposed by Jasak [24] and it has a correction term for the warp-angle based on a tangential correction. For the temporal term an Implicit-Euler scheme was used.

3. Immersed Boundary Treatment

The first step of the IB method consists of isolating the fluid domain consisting of all the cells completely immersed in the fluid region. A boundary is created at the interface of the fluid domain, separating it from the remaining computational domain. At this boundary a Dirichlet boundary condition for velocity is imposed and the usual discretization schemes used in Dirichlet boundaries are applied. The imposed velocity at the interface is computed in a way that the body influence is accurately represented in the overall flow. This is achieved using a least-squares interpolation for each immersed boundary face.

3.1. IB Categorization

For notation purposes cells which contain all vertices outside the solid body are labeled as *fluid cells*. Cells with vertices both inside and outside the solid body are labeled as *immersed boundary cells* (or *ib cells* for short), and any face these cells have that is shared with a *fluid cell* is labeled as an *ib face*. The remaining cells are labeled as *solid cells* (fig. 1). An advantage of this kind of categorization is that it does not create grid quality problems as those reported in the cut-cell methods.

The solid body's surface is described by discrete points, with known velocity and position at each time instant (prescribed motion). Only the closest solid point to each *ib face* is used, which leads to a linearization of the solid boundary.

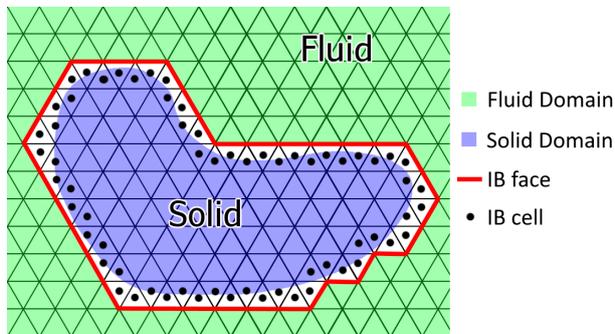


Figure 1: Example of the conservative cut around a generic solid body in a triangular grid.

Two different interpolation techniques used in this work are described in the following subsections.

3.2. The Unconstrained Interpolation Method

The standard method, that does not constrain continuity, interpolates each velocity component by a polynomial of the form:

$$u = a_0 + a_1x + a_2y + a_3x^2 + a_4y^2 + a_5xy \quad (6)$$

$$v = b_0 + b_1x + b_2y + b_3x^2 + b_4y^2 + b_5xy \quad (7)$$

For the least-squares interpolation a stencil with a relatively large number of points (six or higher) is required for each *ib face*. To assemble the stencil first the *fluid cell* containing the *ib face* is identified. Then all *fluid cells* that share vertices with that cell are added to the stencil. If any of the *fluid cells* in the stencil contains an *ib face*, then its associated solid point is also added to the stencil. The stencil for a generic *ib face* is shown in fig. 2.

The least-squares problem, for example for the first component of the velocity field can be written in matrix notation as:

$$\begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & y_1^2 & x_1y_1 \\ 1 & x_2 & y_2 & x_2^2 & y_2^2 & x_2y_2 \\ & & & \dots & & \\ 1 & x_n & y_n & x_n^2 & y_n^2 & x_ny_n \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ \dots \\ u_n \end{bmatrix} \quad (8)$$

Once the least-squares problem is solved and the polynomials' coefficients are known, both velocity components can be computed and applied to the *ib face* as a Dirichlet boundary condition. This process is repeated for each *ib face*.

3.3. Proposed Continuity Constrained Least Squares (CCLS) Interpolation Method

A new method for velocity interpolation consists of a modification of the previously explained least-squares method. The modifications consist in guaranteeing that the least-squares polynomials verify the continuity equation locally and also that each neighboring interpolation polynomial is continuous with adjacent polynomials, which guarantees global fluid continuity in the IB domain.

To strictly enforce continuity locally the two polynomials for the velocity components u and v are coupled. The coefficients of the two polynomials which appear in the divergence equation are replaced by common coefficients, in a way as to guarantee $\frac{\partial u}{\partial x} = -\frac{\partial v}{\partial y}$. To achieve a sharper interface a constraint is applied to the velocity of the main solid point (the solid point associated with the current *ib face*). This results in two polynomials of the form:

$$(u - u_s) = a_1x + a_2y + a_3x^2 + a_4y^2 + a_5xy + a_6x^2y \quad (9)$$

$$(v - v_s) = b_1x - a_1y + b_3x^2 - \frac{a_5}{2}y^2 - 2a_3xy - a_6xy^2 \quad (10)$$

Notice that two extra terms were added to the polynomials which allows a better overall fit without computational cost since Gauss elimination was already manually applied to take into account both the continuity constraint and the solid point's velocity constraint.

Besides enforcing local continuity it is also verified that guaranteeing that the polynomials are continuous in all the *ib* domain has a significant effect in suppressing SFOs. To achieve this, *compatibility points* between each two neighboring *ib faces* are generated and added to the stencil in a way that guarantees that the polynomials have the same value at these points.

To compute these *compatibility points* firstly the medium point between the two solid points (associated with each of the neighboring *ib faces*) is found. Then the *compatibility point* is found by computing the medium point between this *pseudo solid point* and the common vertex of the two neighboring *ib faces*. The compatibility points' position can be found with the following equation:

$$\mathbf{c} = 0.5\mathbf{v} + 0.25\mathbf{sp}_0 + 0.25\mathbf{sp}_1 \quad (11)$$

where \mathbf{c} is the *compatibility point*, \mathbf{v} is the common vertex between the two *ib faces*, and \mathbf{sp}_0 and \mathbf{sp}_1 are the solid points associated with each *ib face*.

Figure 2 illustrates the complete stencil for a generic *ib face*, including the *compatibility points* and auxiliary geometry to compute them.

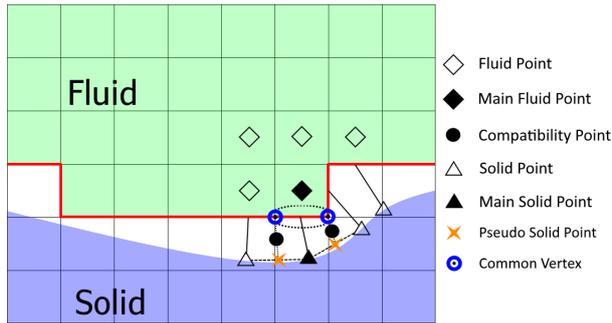


Figure 2: Diagram of a stencil used in the least-squares interpolation for the circled *ib face*.

The velocity at each *compatibility point* is unknown and the interpolation is performed iteratively. In the first iteration the least-squares problem is solved without *compatibility points*. Afterwards the velocity at the *compatibility points* is estimated by averaging the interpolation result from each of the polynomials of the two neighboring *ib faces*. A second iteration is then computed, with the *compatibility points* added to the least-squares problem with a low interpolation weight. This process is repeated, increasing the interpolation weight of the *compatibility points* until the two neighboring polynomials are continuous at these points. The proposed algorithm converges in five compatibility iterations, as will be shown in the results section.

When a cell in one iteration undergoes from being a *non-fluid cell* to being a *fluid cell*, henceforth called a *fresh cell*, a problem arises because there is

no time history from this cell. To solve this problem the previous iteration's least-squares polynomials are used to calculate a value of u and v at the *fresh cell's* centroid at the previous time instant.

This interpolation of history values creates a limitation in how much the boundary can move per time step due to the domain of the interpolated polynomial. Like most immersed boundary methods, it is imposed that the boundary cannot move more than one grid cell per time step, which corresponds to a body Courant number equal to 1, with body Courant given by the cell size h , the time step Δt and the body velocity U_{body} :

$$C_{body} = U_{body} \frac{\Delta t}{h} \quad (12)$$

3.4. Immersed Boundary Method in Unstructured Grids

The main difference when using the IB method with Cartesian vs unstructured grids is the conservative cut performed and the least-squares stencil. The immersed boundary topology varies wildly with the considered grid type as can be seen in fig. 3. The cut will affect the number of *ib faces* and *solid points*, leading to a better/worse defined body. It also affects the distance between the immersed boundary and the solid and therefore the interpolation polynomial's domain.

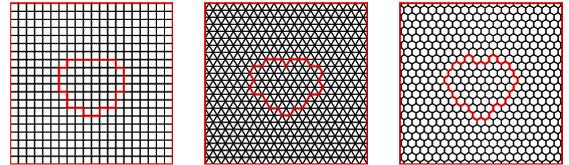


Figure 3: Conservative cut of a heart-shape in a Cartesian, triangular and polyhedral grids respectively.

The different number of faces per cell depending on grid type also affects the immersed boundary's performance due to the connectivity of the system's matrices in the bulk flow solver and also to the balance of the least-squares stencils, which depend on cell vertex neighbors, as well as number of *ib faces* per cell.

These subtle changes affect the performance of the IB method in ways that are hard to predict, so a comparison study is required to characterize the effects of the grid type on the results obtained.

4. Results and Discussion

4.1. Lid Cavity Flow with a Heart-Shape Body

The problem of the square lid-driven cavity with forcing term was chosen to validate the immersed boundary method in an unstructured framework.

This problem was chosen for validation purposes due to the availability of an analytic solution [25,26] that provides a way to accurately calculate error norms.

The problem consists of a square cavity of side length 1, with zero velocity on all boundaries except the upper one. The imposed lid velocity at the top boundary and the analytic solution can be found in the referenced works. A Reynolds number of $Re = 1$ was used.

To test the immersed boundary method, a solid body is introduced at the center of the cavity. This body was chosen with a complex heart looking shape (note that this is an abstract shape with no relation with biology) fig. 4. The complex geometry was chosen to prove the methods' robustness and independence of body shape. At the body's surface the analytic local velocity is imposed, which is transmitted to the fluid through the IB method. The fluid flow is not solved in the interior of the body.

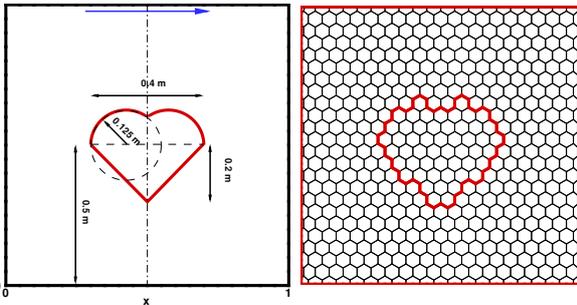


Figure 4: Diagram of heart-shape in square cavity, and resulting fluid/solid domain separation in a polyhedral grid.

Three different grid types are considered: Cartesian, triangular and polyhedral. For each grid type four different consecutively refined grids are created as regular as possible. The Cartesian grids feature 20×20 , 40×40 , 80×80 and 160×160 cells, and the triangular and polyhedral ones were generated to have comparable cell reference length h to the Cartesian one.

Figure 5 shows the error norms obtained when solving the problem with all grids, with and without the immersed body. The spatial convergence is of second order (Δx^2) independently of the grid type, which shows the robustness of the bulk flow unstructured solver. It can also be verified that the errors obtained with and without the IB body are not significantly different, which shows that the IB method does not introduce a significant error with any grid type, even with the complex geometry used in this case. This observation validates the method as an accurate and robust unstructured IB method, which produces good results independently of grid

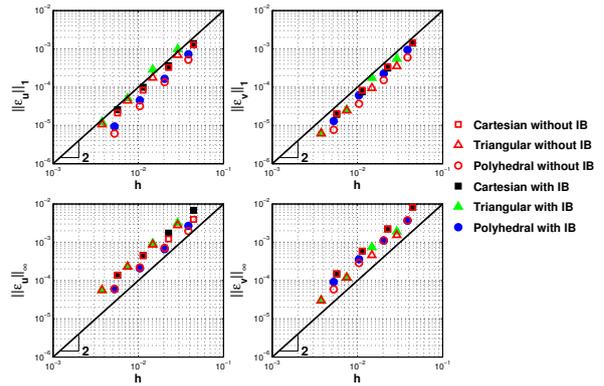


Figure 5: Error norms for different grid types with and without the immersed body heart-shape.

type or body geometry.

4.2. Force Oscillations in an Oscillating Cylinder in Static Flow with Cartesian Grid

To compare the proposed method with other SFO suppression related studies, the problem of an oscillating cylinder in static flow was chosen. The cylinder oscillates horizontally in a domain of $4D \times 4D$, with pressure outlet boundary conditions on the left and right outer boundaries, and fixed wall boundary conditions in the upper and lower boundaries. The problem was chosen in accordance to [4]. The cylinder's position obeys to:

$$x(t) = \frac{A_m}{2} \sin(2\pi ft) \quad (13)$$

$$A_m = \frac{D}{4} \quad (14) \quad f = 0.995 Hz \quad (15)$$

The Reynolds number in relation to the maximum translational velocity is equal to 78.5. Four regular Cartesian grids 64×64 , 128×128 , 192×192 and 256×256 were considered and four time steps were used: $\Delta t = 0.002T, 0.004T, 0.008T$ and $0.016T$. These grids and time steps are consistent with the ones used in [4], providing a way to compare the proposed method with an already established SFO suppression method.

Figure 6 shows the pressure force over time for both the proposed and the standard unconstrained method. The proposed method provides reasonably smooth results for the different grids and time steps, showing the SFOs have been eliminated. The standard unconstrained method shows strong oscillations for all tested grids and time steps. It is noticeable for both methods a significant decrease in SFOs with grid refinement and a slight decrease in SFOs with increasing time step. The observed behavior is in agreement with other authors [4,5].

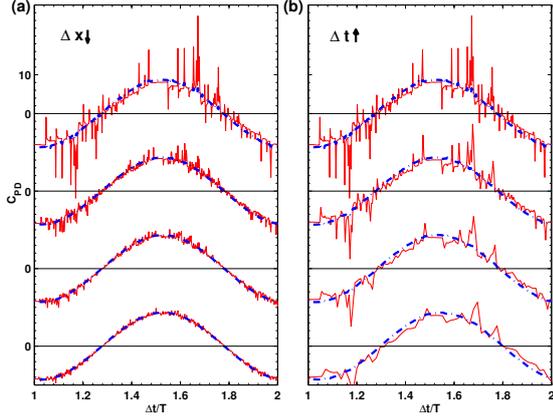


Figure 6: Time history of C_{PD} for four grids with constant time step (a) and four different time steps with constant grid (— standard method, - - - proposed method).

To measure the variation in oscillations when varying Δx and Δt the metric proposed by Seo et al. [4] is considered. $C_{PD}^{2\delta}$ is defined in the following way:

$$C_{PD}^{2\delta} = |C_{PD}^{n+1} - 2C_{PD}^n + C_{PD}^{n-1}| \quad (16)$$

The root mean square of this function is subsequently used to provide a comparative value on the magnitude of the oscillations in the force calculation.

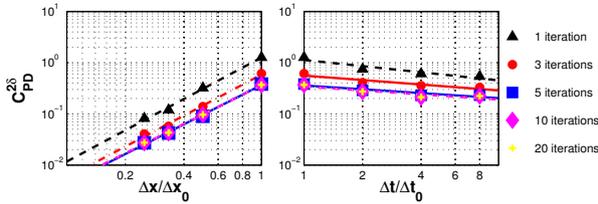


Figure 7: Variation of $C_{PD}^{2\delta}$ for different number of compatibilization iterations for four grids with constant time step (left) and four different time steps with constant grid (right).

Figure 7 shows the oscillations' metric plotted for different numbers of compatibilization iterations, where it can be seen that five iterations are sufficient to achieve the desired effect. Higher number of compatibilization iterations provide no apparent benefit. When five (or more) compatibilization iterations are performed the SFOs present correspond roughly to the continuity constrained interpolation method with no compatibilization for a grid with Δx halved. This correspondence validates the iterations' effectiveness.

Figure 8 shows a comparison between the proposed and the standard method, together with the

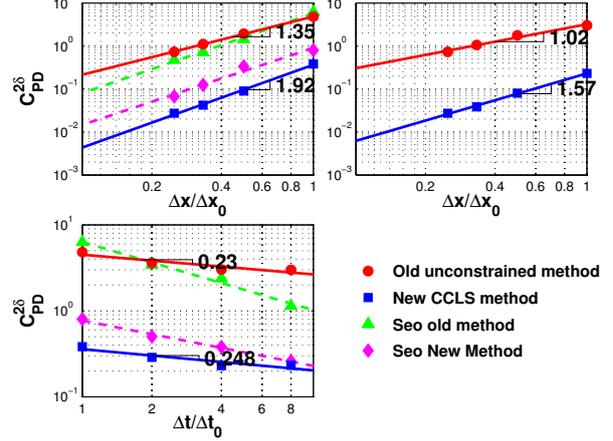


Figure 8: Variation of $C_{PD}^{2\delta}$ for four grids with constant time step (top left), four grids with constant body Courant number (top right) and four different time steps with constant grid (bottom left).

results of the original and proposed methods proposed by Seo et al. [4]. The SFO suppression provided by the proposed CCLS method is higher than the one obtained by Seo in [4]. With the proposed method the SFOs decrease faster with grid refinement than with the standard method and show roughly a second order slope in agreement with many authors' observations [4–6]. However the SFOs first order trend when increasing the time step is not found. This was also the case in [4], which reports the SFOs decrease with $\Delta t^{0.5}$, instead of an unexplained linear (Δt^1) behavior reported by other authors. Figure 8 (top right) shows for the present calculations that keeping constant the body Courant number SFOs decrease with a slope $\Delta x^{1.57}$. These results show that the proposed method is less sensitive to the time step used, and is capable of achieving smooth results faster when grid refinements are performed with body Courant constraints in comparison to other SFO suppression methods.

4.3. Oscillating Cylinder in Static Flow with Unstructured Grids

The problem of an oscillating cylinder in static flow is simulated in this section in order to validate the method in moving body conditions together with unstructured grids and to compare the produced oscillations by each grid type. The problem is similar to the one in the previous section with movement constants chosen in accordance with [27]. The movement amplitude A_m is 1.6D, where D is the cylinder diameter, and the frequency is 0.05Hz. The fluid viscosity is chosen to achieve a Reynolds number of 100, based on the maximum velocity, and a Keulegan-Carpenter number of 5.

4.3.1. Grid definition

The domain is chosen to keep the flow dynamics surrounding the cylinder unaffected by far-field effects. The Cartesian grids have a size of $10D \times 10D$, while the triangular and polyhedral ones are of circular shape with a diameter of $10D$. This circular shape is possible due to the flexibility of the unstructured grids and allows the same properties in terms of being distanced from the main flow region as the Cartesian square grid, while also decreasing total fluid grid area.

The grid was refined in the region of interest, which encompasses the cylinder's movement over time. The refinement helps not only in achieving proper spatial resolution in the region with higher vorticity, but also in maintaining a stable pressure field, while allowing for larger cell volume in regions of reduced interest and flow complexity, keeping the overall number of cells to a minimum. The local refinement in a Cartesian grid implies a deformation of whole grid lines, whether vertical or horizontal. As a consequence an unbalanced grid aspect ratio is obtained and significantly deteriorated grid quality namely in the unrefined region, see fig. 9. This problem does not exist in unstructured grids, where the very high grid generation flexibility allows for precise refined regions, together coarser cells unaffected in terms of grid quality. The grids used for this problem all feature a refined region over the cylinder's movement area and a grid quality parameter of neighbor cells' volume ratio below 1.1.

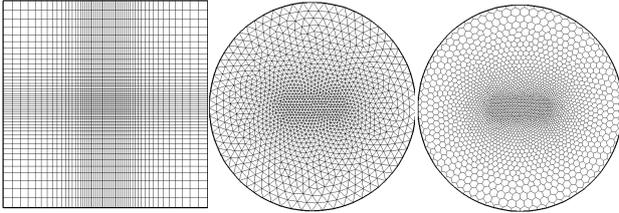


Figure 9: Different types of grids with refinement.

Four grids of each type were generated, featuring a cell count around 2500, 10000, 20000 and 40000. The coarsest grids can be seen in fig. 9 together with domain shapes, as well as the different refinement strategies.

4.3.2. Results of Oscillating Cylinder in Static Flow

Figure 10 shows each force component computed with the finest grid of each of the three different grids types and the smallest of considered time step $\Delta t = T/720$. The results show great agreement with computational results from [27] for all grid types.

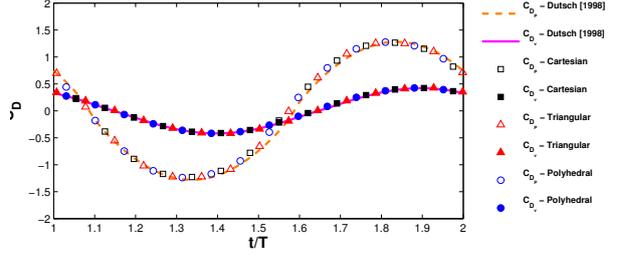


Figure 10: Pressure and viscous drag coefficients with the finest grid of each type.

A battery of simulations was ran with the different grids and different time steps. The simulations are organized in three sets, similar to the organization of the results in the previous section and the the same metric to measure the force oscillations also is used.

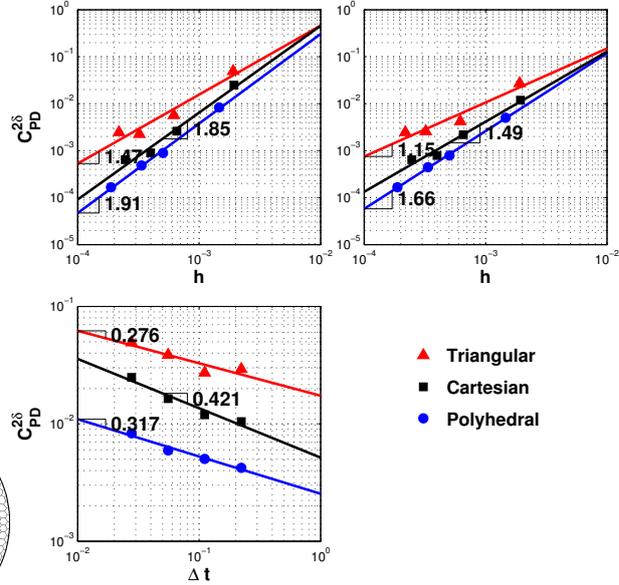


Figure 11: Comparison of force oscillations for various grids and time steps (on the top left: grid refinement with constant time step ($\Delta t = T/720$), top right: grid refinement with similar body Courant, and on the bottom: coarsest grid with varying time step).

Figure 11 shows the metric of oscillations obtained for all grids used. The results for the polyhedral grids are very similar to those seen for Cartesian grids in the previous section.

The oscillations' decrease rate is, however, not independent of grid type. Triangular grids show the worse oscillation decrease behavior, while polyhedral grids show significant faster decrease than Cartesian grids with grid refinement, whether the time step or the CFL number is fixed. It is note-

worthy to appreciate that for all simulations the polyhedral grids constantly show superior oscillation behavior compared with their Cartesian and triangular counterparts.

Polyhedral grids produce smaller oscillations than Cartesian ones, which in turn produce smaller oscillations than triangular grids. A possible explanation is related to the higher connectivity of the cells in a polyhedral grid, which have 6 face neighbor cells. This higher connectivity might stabilize the Poisson equation by reducing the temporal incoherence in boundary conditions at the immersed boundary. A second hypothesis is related to the IB interpolation stencils which in a polyhedral grid will feature more IB faces for a similar cell size, which in turn leads to more solid points and a better body resolution, which seems to be associated with smaller oscillations amplitude.

4.4. 2D Flapping wing

The flow over a flapping wing is simulated to demonstrate the robustness of the proposed method. This flow has been extensively studied, see e.g. Wang [28], Eldredge [29] and Albuquerque [30], and has also been commonly used in SFO investigations in IB methods [5, 6, 31, 32]. This test case consists of a 2D elliptical wing, translating in a sinusoidal movement on a plane with inclination θ_p relative to the horizontal axis, while simultaneously rotating around its center. The wing's position and pitch angle are given by:

$$x(t) = \frac{A_m}{2} \cos(2\pi ft) \cdot \cos(\theta_p) \quad (17)$$

$$y(t) = \frac{A_m}{2} \cos(2\pi ft) \cdot \sin(\theta_p) \quad (18)$$

$$\theta(t) = \frac{\pi}{4} \sin(2\pi ft) + \theta_0 \quad (19)$$

The simulated flow case corresponds to the one studied by [29], with wing movement constants $\theta_p = 0^\circ$, $A_m = 2.8c$, $f = 0.25\text{Hz}$ and $\theta_0 = \pi/2$, for a Reynolds number of 75 and a wing of elliptical cross-section with a thickness ratio of 10. A domain of $20c \times 20c$ with pressure outlet in all outer boundaries is used. The Cartesian grid contains 400×400 non-uniformly distributed cells, with $\Delta x = c/160$ in the finest region. The time step is $dt = T/800$, which results in a maximum body Courant number of 0.88. As was proposed by Eldredge, C_D is defined as the horizontal force always opposing the movement.

Figure 12 shows the instantaneous vorticity contours, nondimensionalized based on the wing chord and maximum translational velocity, at four instants of the flapping movement. An excellent agreement is observable with results from [29].

The forces obtained, as well as those in [29] are shown in fig. 13. An excellent agreement is seen in

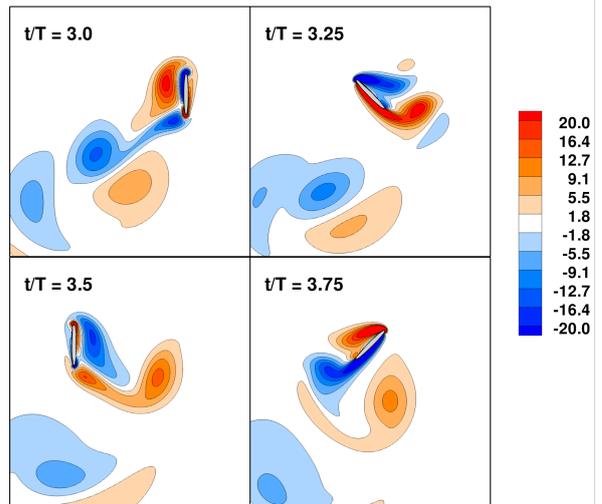


Figure 12: Instantaneous vorticity contours for second configuration of flapping wing problem.

both force components with the literature results. It shows also that the proposed CCLS method provides very smooth results of the force over time, unlike the standard unconstrained method and other immersed boundary methods.

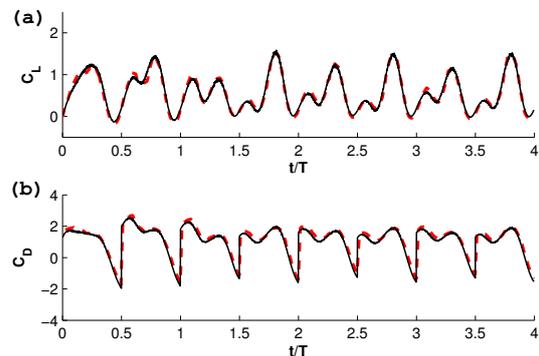


Figure 13: Lift (a) and drag (b) force coefficients' time history for the horizontal movement case (— proposed method, - - Eldredge [29]).

4.5. Triangular cavity with the rotation of an elliptical body

The previous subsections have shown the validity and accuracy of the proposed IB method with a wide range of grid types, as well as the superior qualities of polyhedral grids in reducing spurious force oscillations. The use of unstructured grids body fitted to a complex fixed boundary, in conjunction with an IB method for a moving body of complex geometry allows for the computation of arbitrarily complex problems. In this subsection one such case is shown consisting of the rotation of an elliptical body inside a triangular lid-driven cavity.

The parameters of the problem were chosen in accordance with the work reported in [33]. The cavity has the shape of an equilateral triangle of side l . In one of the triangle's sides tangential velocity is imposed, the other two sides are defined as static solid walls. The Reynolds number based on the triangle's hydraulic diameter and imposed velocity is $Re = 500$. Figure 14 shows the results obtained for

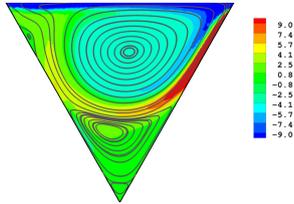


Figure 14: Streamlines and nondimensional vorticity contours of lid-driven triangular cavity at $Re=500$.

the static problem. The vorticity contours together with the flow streamlines are in great agreement with results from [33].

A rotating elliptical body is inserted in the triangular cavity. The ellipse's center matches with the triangle's horizontal center and is placed $0.3463l$ above the triangle's bottom vertex (the body position can be seen in fig. 15). The ellipse has a chord of $c = 0.15l$, and a thickness ratio of 7.5. It rotates clockwise with a prescribed angular velocity of $\omega = 50rad/s$ around it's center.

A polyhedral grid fitted to the outside triangular geometry is generated, with a circular refined region encompassing the immersed body's movement. The grid features 43 282 cells and the refined region has a cell size that results in the immersed body being described by roughly 800 *ib faces*. A time step of $\Delta t = T/838$ was used, which results in a body Courant of 0.74.

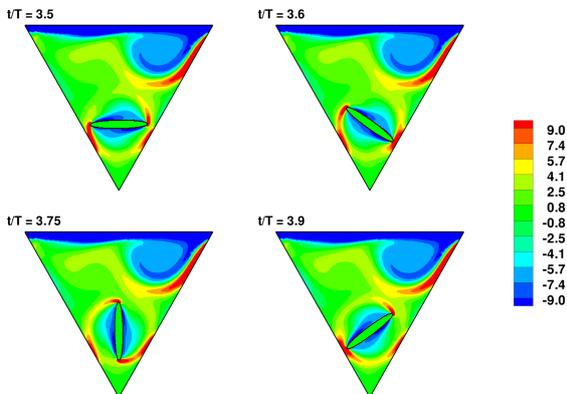


Figure 15: Instantaneous adimensional vorticity contours of lid-driven triangular cavity at $Re=1000$ with rotating elliptical shape.

Figure 15 shows the instantaneous vorticity contours. The introduced body generates a vortex with vorticity comparable to the main vortex in the upper cavity region resulting in a pseudo-static flow, where only the region touched by the body has flow field with significant change over time.

The forces acting on the ellipse also show periodic behavior as seen in fig. 16, however with double of the rotating motion's frequency due to the ellipse's and the movement's symmetry.

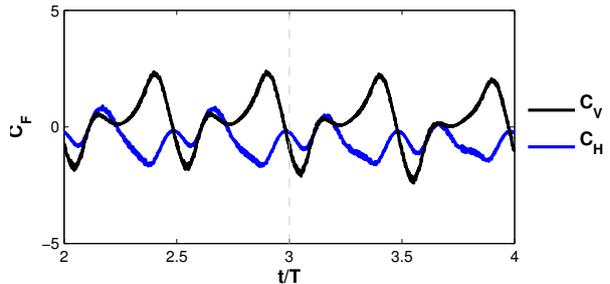


Figure 16: Horizontal and vertical force coefficients acting on elliptical body over time.

5. Conclusions

A new discrete forcing, flow reconstruction immersed boundary method, in a finite volume unstructured grid framework was presented. This new technique enforces local continuity through the use of a continuity constraint in the interpolation least-squares polynomials, as well as polynomial compatibility through an iterative method. The method is suitable for a domain fitted unstructured grid, with extremely flexible local refinement possibilities, together with immersed moving bodies, meaning no re-meshing is required.

The proposed CCLS method was shown to have the same convergence rate as the bulk flow solver independently of grid type and was then validated using both experimental data and literature computational results from body fitted methods. The new method was demonstrated to improve IB methods by eliminating the problem of pressure oscillations.

When it comes to spurious oscillations the polyhedral grids were seen to be superior because the oscillations have an inferior magnitude compared to those produced by both Cartesian and triangular ones.

References

- [1] Charles S Peskin. Flow patterns around heart valves: a numerical method. *Journal of Computational Physics*, 10(2):252–271, 1972.
- [2] Markus Uhlmann. An immersed boundary method with direct forcing for the simulation of particulate flows. *Journal of Computational Physics*, 209(2):448–476, 2005.
- [3] Jianming Yang and Elias Balaras. An embedded-boundary formulation for large-eddy simulation of tur-

- bulent flows interacting with moving boundaries. *Journal of Computational Physics*, 215(1):12–40, 2006.
- [4] Jung Hee Seo and Rajat Mittal. A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations. *Journal of Computational Physics*, 230(19):7347–7363, 2011.
- [5] Jongho Lee, Jungwoo Kim, Haechon Choi, and Kyung-Soo Yang. Sources of spurious force oscillations from an immersed boundary method for moving-body problems. *Journal of Computational Physics*, 230(7):2677–2695, 2011.
- [6] Haoxiang Luo, Hu Dai, Paulo JSA Ferreira de Sousa, and Bo Yin. On the numerical oscillation of the direct-forcing immersed-boundary method for moving boundaries. *Computers & Fluids*, 56:61–76, 2012.
- [7] Xiaolei Yang, Xing Zhang, Zhilin Li, and Guo-Wei He. A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations. *Journal of Computational Physics*, 228(20):7821–7836, 2009.
- [8] Jungwoo Kim, Dongjoo Kim, and Haechon Choi. An immersed-boundary finite-volume method for simulations of flow in complex geometries. *Journal of Computational Physics*, 171(1):132–150, 2001.
- [9] Liang Ge and Fotis Sotiropoulos. A numerical method for solving the 3D unsteady incompressible Navier–Stokes equations in curvilinear domains with complex immersed boundaries. *Journal of Computational Physics*, 225(2):1782–1809, 2007.
- [10] Lin Sun, Sanjay R Mathur, and Jayathi Y Murthy. An unstructured finite-volume method for incompressible flows with complex immersed boundaries. *Numerical Heat Transfer, Part B: Fundamentals*, 58(4):217–241, 2010.
- [11] Kevin Wang, Arthur Rallu, J-F Gerbeau, and Charbel Farhat. Algorithms for interface treatment and load computation in embedded boundary methods for fluid and fluid-structure interaction problems. *International Journal for Numerical Methods in Fluids*, 67(9):1175–1206, 2011.
- [12] Rainald Löhner, Joseph D Baum, Eric Mestreau, Dmitri Sharov, Charles Charman, and Daniele Pelessone. Adaptive embedded unstructured grid methods. *International Journal for Numerical Methods in Engineering*, 60(3):641–660, 2004.
- [13] Rainald Löhner, Sunil Appanaboyina, and Juan R Cebal. Comparison of body-fitted, embedded and immersed solutions of low Reynolds-number 3-D incompressible flows. *International Journal for Numerical Methods in Fluids*, 57(1):13–30, 2008.
- [14] Iman Borazjani, Liang Ge, and Fotis Sotiropoulos. Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3D rigid bodies. *Journal of Computational physics*, 227(16):7587–7620, 2008.
- [15] Anvar Gilmanov, Trung Bao Le, and Fotis Sotiropoulos. A numerical approach for simulating fluid structure interaction of flexible thin shells undergoing arbitrarily large deformations in complex domains. *Journal of Computational Physics*, 300:814–843, 2015.
- [16] J. P. Magalhães, Duarte MS Albuquerque, José MC Pereira, and José CF Pereira. Adaptive mesh finite-volume calculation of 2D lid-cavity corner vortices. *Journal of Computational Physics*, 243:365–381, 2013.
- [17] Duarte MS Albuquerque, José MC Pereira, and José CF Pereira. Residual least-squares error estimate for unstructured h-adaptive meshes. *Numerical Heat Transfer, Part B: Fundamentals*, 67(3):187–210, 2015.
- [18] Duarte MS Albuquerque. Numerical computation of incompressible flows on adaptive and unstructured grids. *PhD Thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa*, 2013.
- [19] J. P. Magalhães. An adaptive framework for the numerical simulation of environmental flows in complex geometries. *PhD Thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa*, 2011.
- [20] Suhas V Patankar and D Brian Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer*, 15(10):1787–1806, 1972.
- [21] Raad I Issa. Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics*, 62(1):40–65, 1986.
- [22] F Juretić and AD Gosman. Error analysis of the finite-volume method with respect to mesh type. *Numerical Heat Transfer, Part B: Fundamentals*, 57(6):414–439, 2010.
- [23] Per-Olof Persson and Gilbert Strang. A simple mesh generator in matlab. *SIAM review*, 46(2):329–345, 2004.
- [24] H. Jasak. *Error analysis and estimation in the Finite Volume method with applications to fluid flows*. PhD thesis, Imperial College, 1996.
- [25] T Shih, C Tan, and B Hwang. Effects of grid staggering on numerical schemes. *International Journal for Numerical Methods in Fluids*, 9(2):193–212, 1989.
- [26] M Kobayashi, José MC Pereira, and José CF Pereira. A conservative Finite-Volume second-order accurate projection method on hybrid unstructured grids. *Journal of Computational Physics*, 150:40–75, 1999.
- [27] H Dütsch, F Durst, S Becker, and H Lienhart. Low-Reynolds number flow around an oscillating circular cylinder at low Keulegan-Carpenter numbers. *Journal of Fluid Mechanics*, 360:249–271, 1998.
- [28] Z Jane Wang. Two dimensional mechanism for insect hovering. *Physical Review Letters*, 85(10):2216, 2000.
- [29] Jeff D Eldredge. Numerical simulation of the fluid dynamics of 2D rigid body motion with the vortex particle method. *Journal of Computational Physics*, 221(2):626–648, 2007.
- [30] Duarte MS Albuquerque, José MC Pereira, and José CF Pereira. Calculation of a deformable membrane airfoil in hovering flight. *Computer Modeling in Engineering & Sciences (CMES)*, 72(4):337–366, 2011.
- [31] Y Zhang, CH Zhou, and JQ Ai. Improvement of mass conservation and reduction of spurious oscillations for the local DFD method. *International Journal of Computational Fluid Dynamics*, 29(3-5):257–271, 2015.
- [32] Jinmo Lee and Donghyun You. An implicit ghost-cell immersed boundary method for simulations of moving body problems with control of spurious force oscillations. *Journal of Computational Physics*, 233:295–314, 2013.
- [33] N Mai-Duy and T Tran-Cong. A high-order upwind control-volume method based on integrated RBFs for fluid-flow problems. *International Journal for Numerical Methods in Fluids*, 67(12):1973–1992, 2011.