Journal of Computational Physics 229 (2010) 1043–1076

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

The LS-STAG method: A new immersed boundary/level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties

Yoann Cheny, Olivier Botella *

LEMTA, Nancy-University, CNRS, 2, Avenue de la Forêt de Haye, B.P. 160, 54504 Vandœuvre-lés-Nancy, France

ARTICLE INFO

Article history:
Received 10 June 2008
Received in revised form 5 October 2009
Accepted 6 October 2009
Available online 14 October 2009

Keywords:
Incompressible viscous flows
Complex geometries
Immersed boundary methods
Cut-cell methods
Finite volume methods

ABSTRACT

This paper concerns the development of a new Cartesian grid/immersed boundary (IB) method for the computation of incompressible viscous flows in two-dimensional irregular geometries. In IB methods, the computational grid is not aligned with the irregular boundary, and of upmost importance for accuracy and stability is the discretization in cells which are cut by the boundary, the so-called "cut-cells". In this paper, we present a new IB method, called the LS-STAG method, which is based on the MAC method for staggered Cartesian grids and where the irregular boundary is sharply represented by its level-set function. This implicit representation of the immersed boundary enables us to calculate efficiently the geometry parameters of the cut-cells. We have achieved a novel discretization of the fluxes in the cut-cells by enforcing the strict conservation of total mass, momentum and kinetic energy at the discrete level. Our discretization in the cut-cells is consistent with the MAC discretization used in Cartesian fluid cells, and has the ability to preserve the five-point Cartesian structure of the stencil, resulting in a highly computationally efficient method. The accuracy and robustness of our method is assessed on canonical flows at low to moderate Reynolds number: Taylor–Couette flow, flows past a circular cylinder, including the case where the cylinder has forced oscillatory rotations. Finally, we will extend the LS-STAG method to the handling of moving immersed boundaries and present some results for the transversely oscillating cylinder flow in a free-stream.

1. Introduction

Much attention has recently been devoted to the extension of Cartesian grid flow solvers to complex geometries by immersed boundary (IB) methods (see [28,39] for recent reviews). In these methods, the irregular boundary is not aligned with the computational grid, and the treatment of the *cut-cells*, cells of irregular shape which are formed by the intersection of the Cartesian cells by the immersed boundary, remains an important issue. Indeed, the discretization in these cut-cells should be designed such that: (a) the global stability and accuracy of the original Cartesian method are not severely diminished and (b) the high computational efficiency of the structured solver is preserved.

Two major classes of IB methods can be distinguished on the basis of their treatment of cut-cells. Classical IB methods such as the *momentum forcing method* introduced by Mohd-Yusof and co-workers [41,14], use a finite-volume/difference structured solver in Cartesian cells away from the irregular boundary, and discard the discretization of flow equations in

* Corresponding author.
E-mail address: Olivier.Botella@ensem.inpl-nancy.fr (O. Botella).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.10.007

the cut-cells. Instead, special interpolations are used for setting the value of the dependent variables in the latter cells. Thus, strict conservation of quantities such as mass, momentum or kinetic energy is not observed near the irregular boundary. The most severe manifestations of these shortcomings is the occurrence of non-divergence free velocities or unphysical oscillations of the pressure in the vicinity of the immersed boundary [43,30]. Numerous revisions of these interpolations are still proposed for improving the accuracy and consistency of this class of IB methods [30,3,49,43].

A second class of IB methods (also called *cut-cell methods* or simply *Cartesian grid methods*, see [66,59,61,32,12,9,40]) aims for actually discretizing the flow equations in cut-cells. The discretization in the cut-cells is usually performed by *ad hoc* treatments which have more in common with the techniques used on curvilinear or unstructured body-conformal grids than Cartesian techniques. Most notable is the cell merging technique used by Ye et al. [66] and Chung [9] that merges a cut-cell with a neighboring Cartesian cell to form a new polygonal cell with more than four neighbors. The discretization stencil in this newly formed cell loses thus the five-point structure (in 2D) of Cartesian methods. Such treatments of the cut-cells generate a non-negligible bookkeeping to discretize the flow equations and actually solve them, and it is difficult to evaluate the impact of these treatments on the computational cost of the flow simulations.

The purpose of this article is to present a new IB method for incompressible viscous flows which takes the best aspects of both classes of IB methods. This method, called the LS-STAG method, is based on the *symmetry preserving* finite-volume method by Verstappen and Veldman [63], which has the ability to preserve on non-uniform staggered Cartesian grids the conservation properties (for total mass, momentum and kinetic energy) of the original MAC method [24]. The LS-STAG method has the following distinctive features:

- A sharp representation of the immersed boundary is obtained by using a signed distance function (*i.e.* the level-set function [46,47]) for its implicit representation. Level-set methods were devised by Osher and Sethian [48] for the solution of computational physics problems involving dynamic interfaces. So far for incompressible flows, the main application areas of level-set methods have been the computation of two-phase flows [56]. In the present paper, the level-set function enables us to easily compute all relevant geometry parameters of the computational cells, reducing thus the bookkeeping associated to the handling of complex geometries.
- In contrast to classical IB methods, flow variables are actually computed in the cut-cells, and not interpolated. Furthermore, the LS-STAG method has the ability to discretize the fluxes in Cartesian and cut-cells in a *consistent and unified fashion*: there is no need for deriving an *ad hoc* treatment for the cut-cells, which would be totally disconnected from the basic MAC discretization used in the Cartesian cells.
- For building our discretization, we have required the strict conservation of global quantities such as total mass, momentum and kinetic energy in the whole fluid domain, which are crucial properties for obtaining physically realistic numerical solutions [1,42,63]. To achieve these preservation properties up to the cut-cells, we had to precisely take into account the terms acting on the immersed boundary in the global conservation equations, at both continuous and discrete levels. As a result, the convective, pressure and viscous fluxes have been unambiguously determined by these requirements, and the boundary conditions at the immersed boundary have been incorporated into these fluxes with a consistent manner.
- From the algorithmic point of view, one of the main consequences is that the LS-STAG discretization preserves the five-point structure of the original Cartesian method. This property allowed the use of an efficient black box multigrid solver for structured grids [62], where no *ad hoc* modifications had to be undertaken for taking account of the immersed boundary.

We also mention that a first attempt at constructing an energy-conserving IB method from the ideas of Verstappen and Veldman can be found in [12]. In this paper however most of the computational aspects of the method has been skipped: it appears that computation of the geometry parameters of the cut-cells, shape of the velocity control volumes, imposition of the boundary conditions at the IB surface and computation of the diffusive terms are different than in the LS-STAG method.

The paper is organized as follows. In Section 2, we recall the notations and salient properties of the staggered Cartesian mesh, and then we present the LS-STAG mesh, its extension for the handling of immersed boundaries. Section 3 presents the LS-STAG discretization in the case the immersed boundary is steady. First, we will recall the global conservation laws for total mass, momentum and kinetic energy that will be used for deriving the LS-STAG method. Then, we will present the discretization of the continuity equation, which is valid in both cut-cells and Cartesian cells. As a matter of fact, we shall observe that the consistency of the discrete continuity equation is a crucial point for building an energy and momentum preserving method for incompressible flows. In the next subsections, we will impose kinetic energy conservation upon our numerical scheme for completely characterizing the discrete pressure and convective fluxes in the cut-cells, and total momentum conservation for the determination of the viscous fluxes. We mention that the discretization of the viscous fluxes has been by far the most intricate part of the LS-STAG discretization in the cut-cells. Section 4 is devoted to numerical tests on canonical flows at low to moderate Reynolds number for assessing the accuracy and robustness of the LS-STAG method. Comparisons with an unstructured solver in terms of CPU time and accuracy will be given. Finally, we will present some results for one of the most appealing features of IB methods: the ability to compute flows with immersed moving boundaries on fixed cartesian grids, without the need for domain remeshing at each time step.

2. Preliminaries and description of the LS-STAG mesh

Let Ω be a rectangular computational domain and Γ its surface. The governing equations are the incompressible Navier–Stokes equations in integral form. In the following, we will consider the finite-volume discretization of the continuity equation:

$$\int_\Gamma \boldsymbol{v}\cdot\boldsymbol{n}\,\mathrm{d}S = 0, \tag{1}$$

where $\boldsymbol{v} = (u, v)$ is the velocity, and the momentum equations in the x and y directions, respectively:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega u\,\mathrm{d}V + \int_\Gamma (\boldsymbol{v}\cdot\boldsymbol{n})u\,\mathrm{d}S + \int_\Gamma p\boldsymbol{e}_x\cdot\boldsymbol{n}\,\mathrm{d}S - \int_\Gamma \nu\nabla u\cdot\boldsymbol{n}\,\mathrm{d}S = 0, \tag{2a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega v\,\mathrm{d}V + \int_\Gamma (\boldsymbol{v}\cdot\boldsymbol{n})v\,\mathrm{d}S + \int_\Gamma p\boldsymbol{e}_y\cdot\boldsymbol{n}\,\mathrm{d}S - \int_\Gamma \nu\nabla v\cdot\boldsymbol{n}\,\mathrm{d}S = 0, \tag{2b}$$

where p is the pressure and ν is the kinematic viscosity.

2.1. The staggered MAC mesh for Cartesian geometries

The Cartesian method on which our IB method is based is the second-order finite-volume discretization of Verstappen and Veldman [63], which has the ability to preserve on non-uniform Cartesian cells the conservation properties (for total mass, momentum and kinetic energy) of the original MAC method on a uniform staggered grid [24]. The staggered arrangement of the unknowns in a Cartesian cell is represented in Fig. 1. The rectangular computational Ω is partitioned into Cartesian cells $\Omega_{i,j} =]x_{i-1}, x_i[\times]y_{j-1}, y_j[$, whose volume is $V_{i,j} = \Delta x_i \Delta y_j$ and center is $\boldsymbol{x}^{\mathrm{c}}_{ij} = (x^{\mathrm{c}}_i, y^{\mathrm{c}}_j)$. The surface $\Gamma_{i,j}$ of cell $\Omega_{i,j}$ is subdivided into four elementary plane faces as:

$$\Gamma_{i,j} = \Gamma^{\mathrm{e}}_{i,j} \cup \Gamma^{\mathrm{w}}_{i,j} \cup \Gamma^{\mathrm{n}}_{i,j} \cup \Gamma^{\mathrm{s}}_{i,j}, \tag{3}$$

by using the usual compass notations (*e.g.* [17]). Cell $\Omega_{i,j}$ is used as a control volume for discretizing the continuity equation (1), whereas the staggered cell $\Omega^u_{i,j} =]x^{\mathrm{c}}_i, x^{\mathrm{c}}_{i+1}[\times]y_{j-1}, y_j[$ is the control volume for the x-momentum equation (2a). For subdividing the surface $\Gamma^u_{i,j}$ of this control volume, we first decompose the north and south faces of $\Omega_{i,j}$ as $\Gamma^{\mathrm{n}}_{i,j} = \Gamma^{\mathrm{n,w}}_{i,j} \cup \Gamma^{\mathrm{n,e}}_{i,j}$ and $\Gamma^{\mathrm{s}}_{i,j} = \Gamma^{\mathrm{s,w}}_{i,j} \cup \Gamma^{\mathrm{s,e}}_{i,j}$, respectively, and then write:

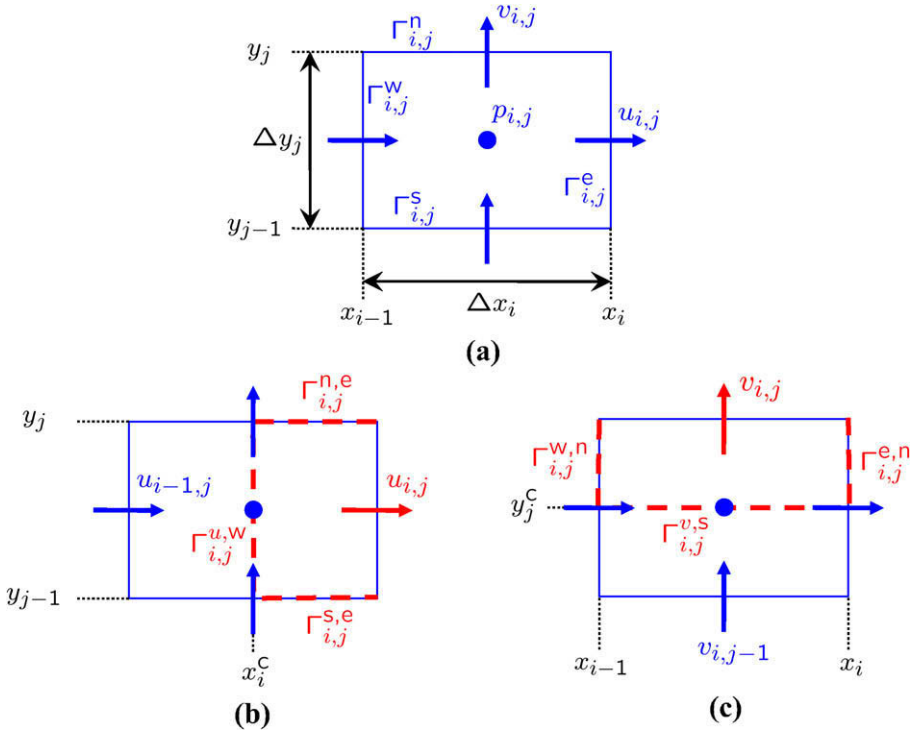


Fig. 1. Staggered arrangement of the variables for (a) a Cartesian cell $\Omega_{i,j}$, and representation of control volumes for (b) $u_{i,j}$ and (c) $v_{i,j}$. The control volumes $\Omega^u_{i,j} \in \Omega_{i,j} \cup \Omega_{i+1,j}$ and $\Omega^v_{i,j} \in \Omega_{i,j} \cup \Omega_{i,j+1}$ are to be completed with their complementary part in $\Omega_{i+1,j}$ and $\Omega_{i,j+1}$, respectively.

$$\Gamma^{u}_{i,j} = \Gamma^{u,\mathrm{w}}_{i,j} \cup \Gamma^{u,\mathrm{e}}_{i,j} \cup \left(\Gamma^{\mathrm{s,e}}_{i,j} \cup \Gamma^{\mathrm{s,w}}_{i+1,j}\right) \cup \left(\Gamma^{\mathrm{n,e}}_{i,j} \cup \Gamma^{\mathrm{n,w}}_{i+1,j}\right). \tag{4}$$

An analogous decomposition holds for the faces of the control volume $\Omega^{v}_{i,j}$, which is used for discretizing the y-momentum equation (2b). These notations will be useful for describing the LS-STAG mesh.

2.2. The LS-STAG mesh for immersed geometries

We consider now an irregular solid domain Ω^{ib} which is embedded in the computational domain Ω, such that $\Omega^{\mathrm{f}} = \Omega \setminus \Omega^{\mathrm{ib}}$ represents the fluid domain where the Navier–Stokes equations are to be discretized. To keep track of the irregular boundary Γ^{ib}, we employ a signed distance function $\phi(\boldsymbol{x})$ (*i.e.* the level-set function [46,47]) such that $\phi(\boldsymbol{x})$ is negative in the fluid region Ω^{f}, $\phi(\boldsymbol{x})$ is positive in the solid region Ω^{ib}, and such that the boundary Γ^{ib} corresponds to the zero level-set of this function, *i.e.*:

$$\phi(\boldsymbol{x}) \equiv \begin{cases} -\Delta, & \boldsymbol{x} \in \Omega^{\mathrm{f}}, \\ 0, & \boldsymbol{x} \in \Gamma^{\mathrm{ib}}, \\ +\Delta, & \boldsymbol{x} \in \Omega^{\mathrm{ib}}, \end{cases} \tag{5}$$

where Δ represents the distance between $\boldsymbol{x}$ and the nearest point on the immersed boundary.

This leads to the modification of the MAC mesh that is described in Fig. 2, and that will be subsequently referred to as the *LS-STAG mesh*. In each cut-cell $\Omega_{i,j}$, the immersed boundary is represented by a line segment whose extremities are defined by linear interpolation of the variable $\phi_{i,j}$, which takes the value of the level-set function $\phi(x_i, y_j)$ at the upper right corner of the cell. We use notations similar to the Cartesian method for the faces of the cut-cells. For example in Fig. 2, the faces of the trapezoidal cut-cell $\Omega_{i,j}$ are denoted:

$$\Gamma_{i,j} = \Gamma^{\mathrm{w}}_{i,j} \cup \Gamma^{\mathrm{e}}_{i,j} \cup \Gamma^{\mathrm{s}}_{i,j} \cup \Gamma^{\mathrm{ib}}_{i,j}, \tag{6}$$

where $\Gamma^{\mathrm{ib}}_{i,j}$ represents the solid north face of the cut-cell. As it will be justified later, the velocity unknowns are exactly located in the middle of the fluid part of the faces. In Fig. 2, the discrete pressure $p_{i,j}$ is located at the intersection of the velocity locations. This location is used for visualization purpose only, and will never be used in the discretization. In effect, we will find out in Section 3.3 that the discrete pressure is piecewise constant in each cut-cell, as in some mixed finite element methods (*e.g.* [50]), and thus does not need to be precisely located in the cut-cells.

As observed in Fig. 2, there are three basic types of cut-cells: *trapezoidal cells* such as $\Omega_{i,j}$ or $\Omega_{i+1,j}$, *triangular cells* (*i.e.* $\Omega_{i-1,j+1}$) and *pentagonal cells* (*i.e.* $\Omega_{i-1,j}$). The discretization of the momentum equations will be performed in the staggered control volumes $\Omega^{u}_{i,j}$ and $\Omega^{v}_{i,j}$, whose shape has to be adapted to each type of cut-cells. For example in Fig. 2, the faces of the control volume $\Omega^{u}_{i,j}$ read:

$$\Gamma^{u}_{i,j} = \Gamma^{u,\mathrm{w}}_{i,j} \cup \Gamma^{u,\mathrm{e}}_{i,j} \cup \left(\Gamma^{\mathrm{s,e}}_{i,j} \cup \Gamma^{\mathrm{s,w}}_{i+1,j}\right) \cup \left(\Gamma^{\mathrm{ib,e}}_{i,j} \cup \Gamma^{\mathrm{ib,w}}_{i+1,j}\right), \tag{7}$$

where the solid faces $\Gamma^{\mathrm{ib,e}}_{i,j} \cup \Gamma^{\mathrm{ib,w}}_{i+1,j}$ are formed with two halves of the solid face of the neighboring trapezoidal cut-cells $\Gamma^{\mathrm{ib,e}}_{i,j} \subset \Gamma^{\mathrm{ib}}_{i,j}$ and $\Gamma^{\mathrm{ib,w}}_{i+1,j} \subset \Gamma^{\mathrm{ib}}_{i+1,j}$. For the other type of cut-cells, these control volumes will be constructed from the six halves of generic control volumes that we represent in Fig. 3. In this figure, the irregular shape of the staggered control volumes is given for representation purpose only, and their geometric parameters, such as their actual volume or shape of the vertical

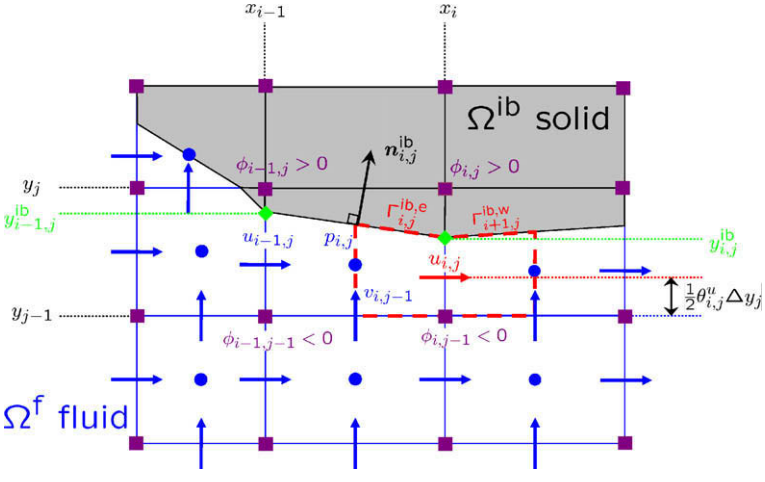


Fig. 2. Staggered arrangement of the variables near the trapezoidal cut-cell $\Omega_{i,j}$ on the LS-STAG mesh.

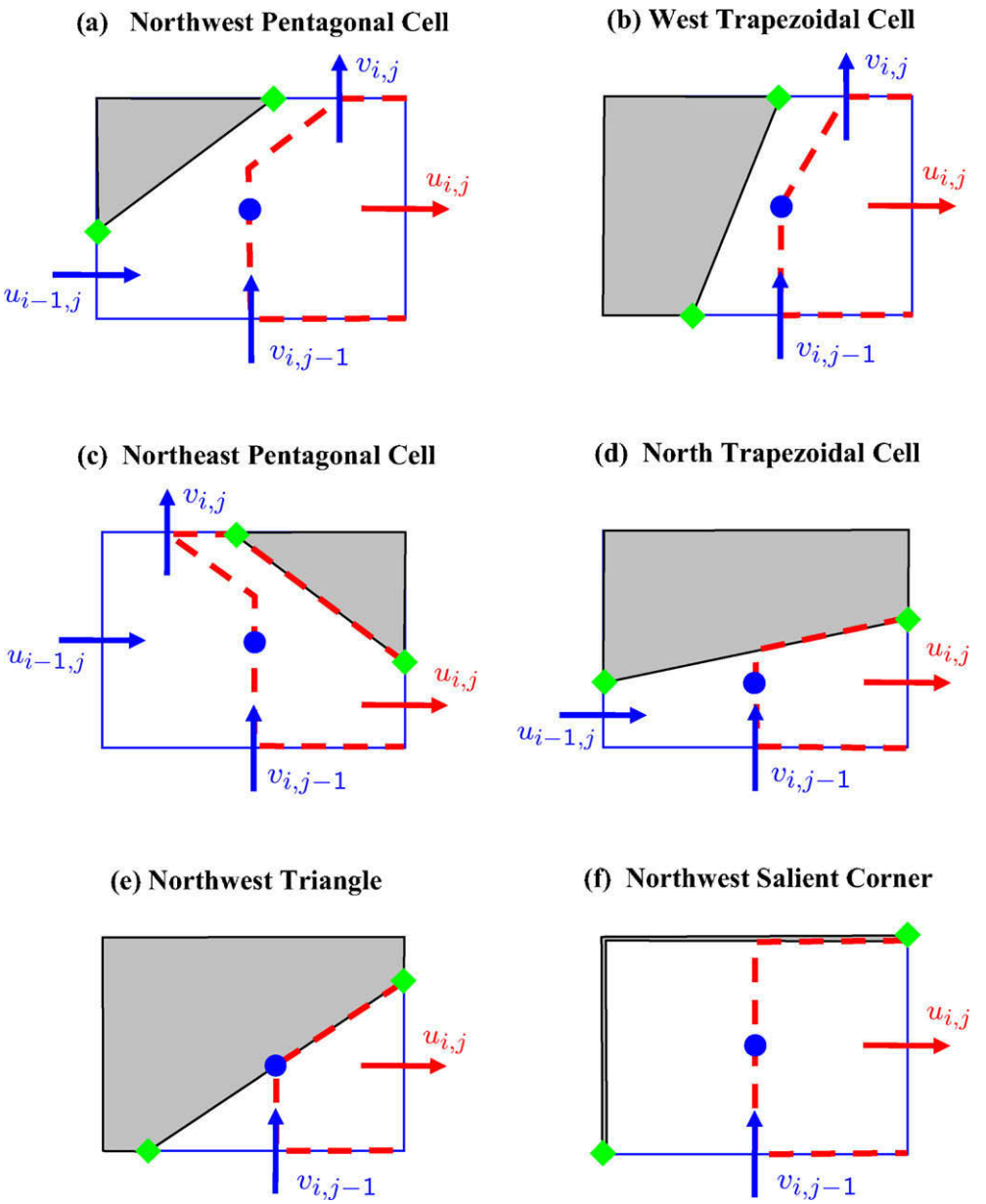


Fig. 3. Basic types of half control volume for the velocity unknown u_{ij} inside the cut-cell Ω_{ij}. The diamonds (◆) denote the locations of the discretization of the velocity boundary conditions. Note that the re-entrant corner of a Cartesian mesh is a particular case of pentagonal cell (c) when $\phi_{ij} = 0$, and that Case (f) (which corresponds to $\phi_{i-1\,j-1} = \phi_{i-1,j} = \phi_{ij} = 0$ and $\phi_{ij-1} < 0$) corresponds to the particular case of a salient corner. Only the definition of the cell volume distinguishes the latter cell from the liming case of triangle (e) defined by $\phi_{i-1\,j-1} = \phi_{ij} = 0,\ \phi_{i-1\,j} > 0$ and $\phi_{ij-1} < 0$.

faces $\Gamma^{u,w}_{i,j}$ and $\Gamma^{u,e}_{i,j}$ are never used by the LS-STAG discretization: instead, we will employ arguments based on the strict conservation of global quantities of the flow, such as conservation of total mass and kinetic energy for discretizing the momentum equations in the cut-cells. For complexness, we present in Fig. 4 all the possible combinations of the half control volumes of Fig. 3 for forming the control volume Ω^u_{ij}. It is important to mention that the LS-STAG discretization does not need to be individually adapted to each of the cases depicted in this figure. Instead, the discretization will be built for the half control volumes of Fig. 3, such that any combination yields a consistent discretization of the momentum equations with the aforementioned global conservation properties.

However, the LS-STAG method relies on a sharp representation of the geometry of the cut-cells Ω_{ij}. In this respect, the level-set function will prove to be a very efficient tool for calculating the geometric parameters of a cut-cell, such as its volume or the projected areas of its faces. A quantity that will be extensively used for calculating these parameters is the fluid portion of the faces of cell Ω_{ij}. For example in Fig. 2, by using one-dimensional linear interpolation of $\phi(x_i, y)$ in $[y_{j-1}, y_j]$, we calculate the length $y^{ib}_{ij} - y_{j-1}$ of the portion of face Γ^{e}_{ij} that belongs to the fluid domain as:

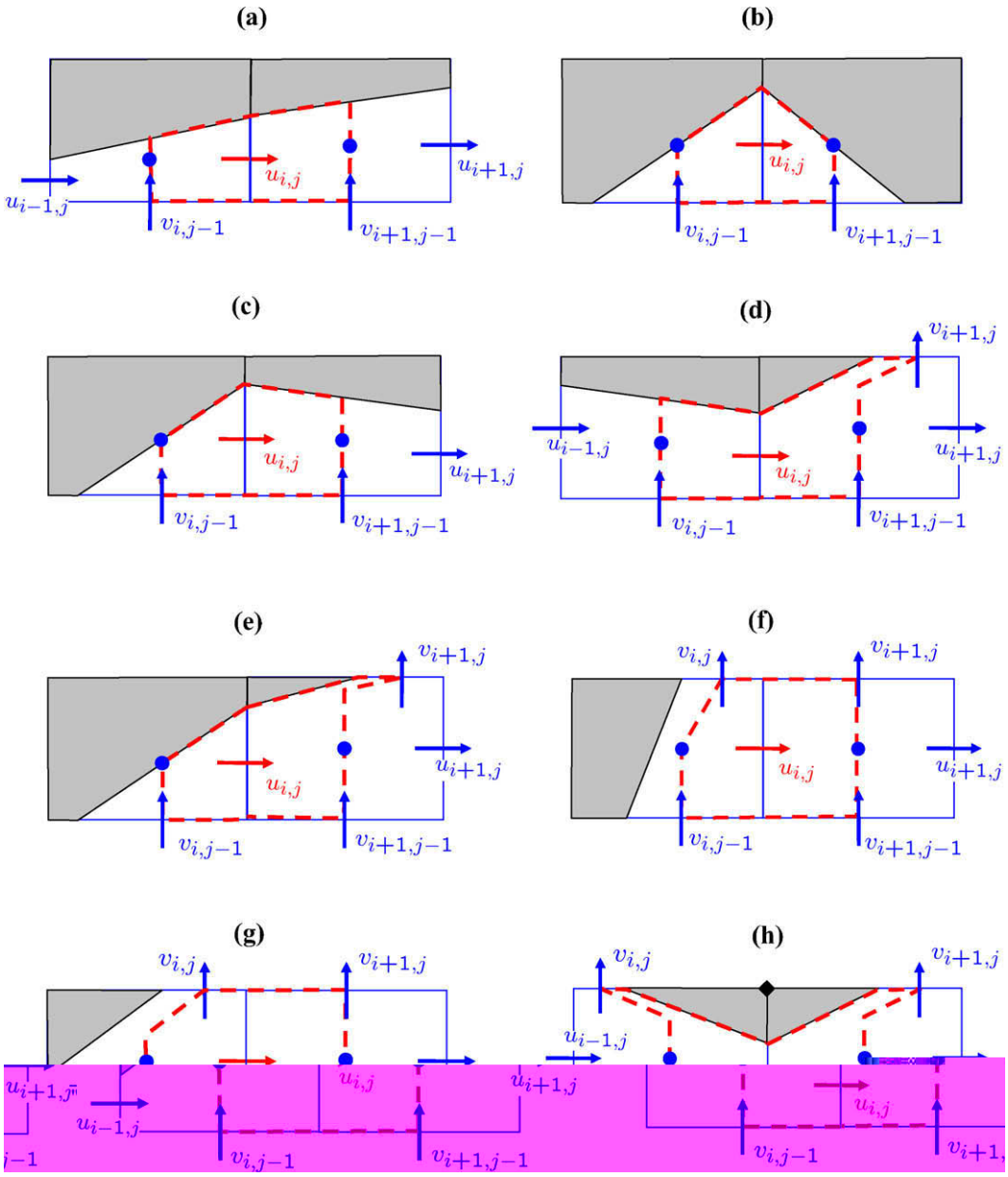


Fig. 4. All possible combinations of the half control volumes of Fig. 3 for forming the control volume $\Omega^u_{i,j}$. Case (h) has to be excluded from the actual flow computations since it would give rise to a non-unique definition of the differential quotient $\partial v/\partial x|_{i,j}$, which is stored at the upper right corner of cut-cell $\Omega_{i,j}$. This case corresponds to an oscillation of the level-set ($\phi_{i-1,j} < 0$, $\phi_{i,j} > 0$ and $\phi_{i+1,j} < 0$). It can be easily filtered prior to the flow computation by setting $\phi_{i,j} = \frac{1}{2}(\phi_{i-1,j} + \phi_{i+1,j})$. This particular case happens very scarcely in the meshes we actually use, since it would correspond to a mesh that is too coarse when compared to the local curvature of the immersed boundary (see [47, Section 1.4]).

$$y^{\mathrm{ib}}_{i,j} - y_{j-1} = \theta^u_{i,j}\Delta y_j, \quad \text{with } \theta^u_{i,j} = \frac{\phi_{i,j-1}}{\phi_{i,j-1} - \phi_{i,j}} \text{ since } \phi\left(x_i, y^{\mathrm{ib}}_{i,j}\right) = 0.$$

The scalar quantities $\theta^u_{i,j}$ and $\theta^v_{i,j}$, which take values in $[0,1]$, will subsequently be called the *cell-face fraction ratios*. They represent the fluid portion of the east and north faces $\Gamma^{\mathrm{e}}_{i,j}$ and $\Gamma^{\mathrm{n}}_{i,j}$, respectively. They will be extensively used for detecting if the discrete velocities $u_{i,j}$ and $v_{i,j}$ belong to the fluid domain, and for discretizing the surface and volume integrals in the Navier–Stokes equations (1) and (2). The cell-face fraction ratios also appear in the analytic expression of the volume $V_{i,j}$ of cut-cell $\Omega_{i,j}$, which is given in Table 1 for the three basic types of cut-cells. We note that $V_{i,j}$ correspond to the *VOF* function which is used for the simulation of multiphasic flows, *e.g.* [53].

Table 1
Analytical formula of the volume for the basic cut-cells of Fig. 3. The volumes of the other cut-cells can be easily deduced from the ones displayed.

Type of cut-cell	Volume
(c) Northeast pentagonal cell	$V_{i,j} = \left[\theta^v_{i,j} + \frac{1}{2}\left(1+\theta^u_{i,j}\right)\left(1-\theta^v_{i,j}\right)\right]\Delta x_i \Delta y_j$
(d) North trapezoidal cell	$V_{i,j} = \frac{1}{2}\left(\theta^u_{i,j} + \theta^u_{i-1,j}\right)\Delta x_i \Delta y_j$
(e) Northwest triangle	$V_{i,j} = \frac{1}{2}\theta^u_{i,j}\theta^v_{i,j-1}\Delta x_i \Delta y_j$
(f) Northwest salient corner	$V_{i,j} = \Delta x_i \Delta y_j$

3. The LS-STAG discretization for stationary immersed geometries

3.1. Global conservation laws for viscous incompressible flows

As early as the 1950s, it was recognized that for physically realistic integration of dynamical systems such as the fluid dynamics equations, linear and quadratic invariants of the continuous equations should be conserved by the numerical scheme [1,34]. For the incompressible Navier–Stokes equations (1) and (2), these flow invariants are the total mass in the whole fluid domain $\int_{\Omega^f} \nabla \cdot \boldsymbol{v}\,\mathrm{d}V$, total momentum $\mathbf{P}(t) = \int_{\Omega^f} \boldsymbol{v}\,\mathrm{d}V$ and, in the case of vanishing viscosity, total kinetic energy $E_c(t) = \frac{1}{2}\int_{\Omega^f} |\boldsymbol{v}|^2\,\mathrm{d}V$.

Conservation equation for total mass and momentum are obtained by a straightforward integration of the Navier–Stokes equations in the whole fluid domain Ω^f. After integrating by parts of the volume integrals, one gets:

$$\int_{\Gamma^{ib}} \boldsymbol{v}\cdot\boldsymbol{n}\,\mathrm{d}S = 0, \tag{8a}$$

$$\frac{\mathrm{d}\mathbf{P}}{\mathrm{d}t} = -\int_{\Gamma^{ib}} \boldsymbol{v}\boldsymbol{v}\cdot\boldsymbol{n}\,\mathrm{d}S - \mathbf{F}, \tag{8b}$$

where $\mathbf{F} = (F_x, F_y)$ represents the hydrodynamic force acting on the immersed boundary, such that:

$$F_x = \int_{\Gamma^{ib}} \left[p - \nu\frac{\partial u}{\partial x}\right]\boldsymbol{e}_x\cdot\boldsymbol{n}\,\mathrm{d}S - \int_{\Gamma^{ib}} \nu\frac{\partial u}{\partial y}\boldsymbol{e}_y\cdot\boldsymbol{n}\,\mathrm{d}S, \tag{9a}$$

$$F_y = -\int_{\Gamma^{ib}} \nu\frac{\partial v}{\partial x}\boldsymbol{e}_x\cdot\boldsymbol{n}\,\mathrm{d}S + \int_{\Gamma^{ib}} \left[p - \nu\frac{\partial v}{\partial y}\right]\boldsymbol{e}_y\cdot\boldsymbol{n}\,\mathrm{d}S. \tag{9b}$$

At the discrete level, total mass is trivially conserved on the MAC mesh because of the staggering of the velocities, and so it will be on the LS-STAG mesh. On non-staggered grids however, we mention that the discrete equivalent of (8a) is not verified, and the mass dissipation of non-staggered methods remains an issue [55,64]. At the discrete level, total momentum is conserved by any numerical method if the momentum equation is written in its conservative form and if the property of local conservation of the numerical fluxes at cell faces holds. If the issue of the implementation of boundary conditions is to be included in the analysis, we may consider that total momentum is conserved if the RHS of the discrete counterpart of (8b) is consistent with the computation of the hydrodynamic force that acts on the solid boundary. This issue is mostly pertinent for non-body conforming methods such as IB methods.

For incompressible flows, the conservation of kinetic energy is a consequence of the Navier–Stokes equations, and the equation for $E_c(t)$ is obtained by multiplying the momentum equation with $\boldsymbol{v}$ and integrating in the whole fluid domain. After integration by parts of the volume integrals, this conservation equation reads:

$$\frac{\mathrm{d}E_c}{\mathrm{d}t} = \int_{\Omega^f}\left[\left(\frac{|\boldsymbol{v}|^2}{2} + p\right)\nabla\cdot\boldsymbol{v} - \nu|\nabla\boldsymbol{v}|^2\right]\mathrm{d}V - \int_{\Gamma^{ib}}\left(\frac{|\boldsymbol{v}|^2}{2} + p - \nu\nabla\boldsymbol{v}\right)\boldsymbol{v}\cdot\boldsymbol{n}\,\mathrm{d}S. \tag{10}$$

Thanks to the continuity equation, the only term remaining in the volume integral involves the viscous stresses, and expresses the loss of energy by viscous dissipation. The non-vanishing terms in the surface integrals show that the pressure and convective terms only influence the kinetic energy budget by their action at the immersed boundary. On uniform Cartesian grids, it is well known the original MAC method with central differencing of the convective term is *energy preserving*, *i.e.* the discrete kinetic energy budget of the scheme mimics (10). However, for more general type of meshes the construction of energy preserving methods is not a trivial task. In [42], Morinishi et al. have extended this property to higher-order finite-difference discretizations on uniform Cartesian grids. More recently, Verstappen and Veldman [63] constructed an energy preserving finite-volume method on non-uniform grids by observing that the discrete kinetic budget mimics (10) if the convective and viscous terms are discretized with skew-symmetric and symmetric positive-definite operators, respectively. In the next sections, we will use similar arguments for building an energy (and momentum) preserving discretization on the LS-STAG mesh. We mention that for constructing this discretization up to the cut-cells, we had to precisely take into account the boundary integrals in (8b), (9) and (10). To our knowledge, these boundary terms have always been neglected in previous studies, with the exception of the recent work by Jameson [29].

In the following, we will focus on semi-discrete conservation properties only, *i.e.* we will ignore issues related to the violation of these properties by the time-stepping scheme, as the other studies cited in this paragraph do. However, we mention that quadratic invariants such as the total kinetic energy can be totally conserved in an unsteady computation if implicit Runge–Kutta (IRK) schemes at Gaussian points are used, of which the well known midpoint rule is a particular case [2]. Nevertheless, several issues arise when IRK schemes are applied to the incompressible Navier–Stokes equations: the implicit nature of the convective discretization, the loss of accuracy which is observed on the computed pressure, and the fact that they are not stiffly accurate which hampers their stability properties. Thus, the design of a practical and energy conservative time-stepping method for incompressible viscous flows is left for future work.

3.2. Discretization of the continuity equation

As in the Cartesian method of Ref. [63], the starting point of the LS-STAG discretization concerns the mass conservation law (1) in cell $\Omega_{i,j}$. For any fluid cell (cut-cell or Cartesian), we denote its faces as $\Gamma_{i,j} = \Gamma^{w}_{i,j} \cup \Gamma^{e}_{i,j} \cup \Gamma^{s}_{i,j} \cup \Gamma^{n}_{i,j} \cup \Gamma^{ib}_{i,j}$, and decompose the continuity equation as the net mass flux through each of these faces:

$$\dot{m}^{i,j} \equiv -\bar{u}_{i-1,j} + \bar{u}_{i,j} - \bar{v}_{i,j-1} + \bar{v}_{i,j} + \overline{U}^{ib}_{i,j} = 0. \tag{11}$$

In this equation, $\overline{U}^{ib}_{i,j} \equiv \int_{\Gamma^{ib}_{i,j}} \boldsymbol{v}^{ib} \cdot \boldsymbol{n}^{ib}_{i,j} \, dS$ denotes the mass flux through the solid part of the cell boundary, where $\boldsymbol{v}^{ib}$ is the velocity datum prescribed at the IB boundary. This boundary mass flux may be non-zero for non-homogeneous boundary conditions only. The mass flow through the fluid part of the faces is denoted with a bar: for example, the flow through face $\Gamma^{e}_{i,j}$ of Fig. 2 is:

$$\bar{u}_{i,j} \equiv \int_{\Gamma^{e}_{i,j}} \boldsymbol{v} \cdot \boldsymbol{e}_x \, dS = \int_{y_{j-1}}^{y^{ib}_{i,j}} u(x_i, y) \, dy. \tag{12}$$

In order to easily discretize this integral, we first locate the discrete unknown $u_{i,j}$ in the middle of the fluid part of the face as:

$$u_{i,j} \equiv u\left(x_i, y_{j-1} + \frac{1}{2}\theta^{u}_{i,j}\Delta y_j\right). \tag{13}$$

Then, by using midpoint quadrature, we obtain:

$$\bar{u}_{i,j} \cong \theta^{u}_{i,j}\Delta y_j u_{i,j}, \tag{14}$$

and following analogous discretizations for the other faces, the discrete continuity equation reads:

$$\dot{m}^{i,j} \equiv \Delta y_j\left(\theta^{u}_{i,j}u_{i,j} - \theta^{u}_{i-1,j}u_{i-1,j}\right) + \Delta x_i\left(\theta^{v}_{i,j}v_{i,j} - \theta^{v}_{i,j-1}v_{i,j-1}\right) + \overline{U}^{ib}_{i,j} = 0. \tag{15}$$

We now turn to the discretization of the boundary term as:

$$\overline{U}^{ib}_{i,j} \cong u^{ib}_{i,j}[n_x\Delta S]^{ib}_{i,j} + v^{ib}_{i,j}[n_y\Delta S]^{ib}_{i,j}, \tag{16}$$

where $[n_x\Delta S]^{ib}_{i,j}$ and $[n_y\Delta S]^{ib}_{i,j}$ are the projected areas of the solid face of the cut-cell in the horizontal and vertical directions, respectively, and velocity $\boldsymbol{v}^{ib}_{i,j} = \left(u^{ib}_{i,j}, v^{ib}_{i,j}\right)$ represents an approximation of the velocity on the solid boundary $\Gamma^{ib}_{i,j}$ of the cut-cell. This last term is calculated with the trapezoidal rule, for example in Fig. 2:

$$\boldsymbol{v}^{ib}_{i,j} = \frac{1}{2}\boldsymbol{v}^{ib}\left(x_i, y^{ib}_{i,j}\right) + \frac{1}{2}\boldsymbol{v}^{ib}\left(x_{i-1}, y^{ib}_{i-1,j}\right),$$

where the velocity boundary condition $\boldsymbol{v}^{ib}(x,y)$ is discretized at the extremities of line segment $\Gamma^{ib}_{i,j}$ in the cut-cell. The projected areas are readily calculated from the cell-face fraction ratios as:

$$[n_x\Delta S]^{ib}_{i,j} = \left(\theta^{u}_{i-1,j} - \theta^{u}_{i,j}\right)\Delta y_j, \quad [n_y\Delta S]^{ib}_{i,j} = \left(\theta^{v}_{i,j-1} - \theta^{v}_{i,j}\right)\Delta x_i. \tag{17}$$

We mention that this discretization of the continuity equation is valid for any type of cut-cells, and in the particular case of a Cartesian fluid cell (such as the cell-face fraction ratios are equal to 0 or 1 only), Eq. (15) reduces to the discrete continuity equation of the original MAC method.

In the following, it will be useful to write the discrete continuity equation in its matrix form:

$$\mathcal{D}U + \overline{U}^{ib} = 0, \tag{18}$$

where each line (i,j) of this system corresponds to Eq. (15) written in cell $\Omega_{i,j}$, and the vectors U and $\overline{U}^{ib}$ contains the velocity unknowns $(u_{i,j}, v_{i,j})$ and the discretization of the boundary terms, respectively.

3.2.1. Discrete conservation of total mass

Now we are able to show that this discretization of the continuity equation conserves total mass in the most general case of the LS-STAG method: *i.e.* when the IB boundary has the time-dependent motion of a rigid body:

$$\boldsymbol{v}^{\mathrm{ib}}(\boldsymbol{x},t)=\boldsymbol{V}^{\mathrm{ib}}(t)+\boldsymbol{\Omega}^{\mathrm{ib}}(t)\times\boldsymbol{Ox}, \tag{19}$$

where $\boldsymbol{O}$ is a reference point of the immersed solid Ω^s, vectors $\boldsymbol{V}^{\mathrm{ib}}(t)$ and $\boldsymbol{\Omega}^{\mathrm{ib}}(t)$ are given translation and angular velocities. We also have to make two assumptions on the computational domain. The first one expresses that the immersed boundary Γ^{ib} is a closed surface, *i.e.*:

$$\int_{\Gamma^{\mathrm{ib}}}\boldsymbol{n}^{\mathrm{ib}}\,\mathrm{d}S=\sum_{\text{Cut-cells}\Omega_{i,j}}\int_{\Gamma^{\mathrm{ib}}_{i,j}}\boldsymbol{n}^{\mathrm{ib}}_{i,j}\,\mathrm{d}S=\boldsymbol{0}, \tag{20a}$$

and the second one states that the fluid domain is not dilatable:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega^{\mathrm{f}}}\mathrm{d}V=\sum_{\text{Cut-cells}\Omega_{i,j}}\int_{\Gamma^{\mathrm{ib}}_{i,j}}\boldsymbol{v}^{\mathrm{ib}}(\boldsymbol{x},t)\cdot\boldsymbol{n}^{\mathrm{ib}}_{i,j}\,\mathrm{d}S=0. \tag{20b}$$

In these equations the integrand is linear in space since $\boldsymbol{v}^{\mathrm{ib}}(\boldsymbol{x},t)$ is given by (19), so midpoint and trapezoidal rules give exact quadrature:

$$\sum_{\text{Cut-cells}\Omega_{i,j}}[\boldsymbol{n}\Delta\boldsymbol{S}]^{\mathrm{ib}}_{i,j}=\boldsymbol{0}, \tag{21a}$$

$$\sum_{\text{Cut-cells}\Omega_{i,j}}\boldsymbol{\Omega}^{\mathrm{ib}}(t)\times\frac{\boldsymbol{Oa}^{\mathrm{ib}}_{i,j}+\boldsymbol{Ob}^{\mathrm{ib}}_{i,j}}{2}\cdot[\boldsymbol{n}\Delta\boldsymbol{S}]^{\mathrm{ib}}_{i,j}=0, \tag{21b}$$

where $\boldsymbol{a}^{\mathrm{ib}}_{i,j}$ and $\boldsymbol{b}^{\mathrm{ib}}_{i,j}$ denote the extremities of line segment $\Gamma^{\mathrm{ib}}_{i,j}$ in a cut-cell.

The discrete counterpart of the conservation of total mass (8a) amounts to summing the discrete continuity equation (11) in all fluid cells. In matrix form it reads:

$$\mathbb{1}^{\mathrm{T}}\mathcal{D}U+\mathbb{1}^{\mathrm{T}}\overline{U}^{\mathrm{ib}}=0, \tag{22}$$

where $\mathbb{1}$ is the constant vector. For proving this identity, we first simplify its left-hand-side by using the local conservativity of the mass fluxes at fluid faces, such that only the fluxes $\overline{U}^{\mathrm{ib}}_{i,j}$ at the IB boundary remains:

$$\mathbb{1}^{\mathrm{T}}\mathcal{D}U+\mathbb{1}^{\mathrm{T}}\overline{U}^{\mathrm{ib}}=\mathbb{1}^{\mathrm{T}}\overline{U}^{\mathrm{ib}}.$$

Then, after integrating the velocity datum (19) with the trapezoidal rule one finally obtains:

$$\mathbb{1}^{\mathrm{T}}\mathcal{D}U+\mathbb{1}^{\mathrm{T}}\overline{U}^{\mathrm{ib}}=\sum_{\text{Cut-cells}\Omega_{i,j}}\boldsymbol{V}^{\mathrm{ib}}(t)\cdot[\boldsymbol{n}\Delta\boldsymbol{S}]^{\mathrm{ib}}_{i,j}+\boldsymbol{\Omega}^{\mathrm{ib}}(t)\times\frac{\boldsymbol{Oa}^{\mathrm{ib}}_{i,j}+\boldsymbol{Ob}^{\mathrm{ib}}_{i,j}}{2}\cdot[\boldsymbol{n}\Delta\boldsymbol{S}]^{\mathrm{ib}}_{i,j}=0,$$

thanks to identities (21a) and (21b). Thus, total mass is trivially conserved by the LS-STAG method. In the following, we will see that the other global conservation properties of Section 3.1 will not be as easily satisfied. Instead, we will have to impose some constraints on the discretization of the momentum equation such that total momentum and kinetic energy be discretely conserved.

3.3. Momentum equation I: energy preserving discretization

Now, we turn to the discretization of the momentum equation (2), whose semi-discrete matrix representation reads:

$$\frac{\mathrm{d}}{\mathrm{d}t}(\mathcal{M}U)+\mathcal{C}[\overline{U}]U+\mathcal{G}P-\nu\mathcal{K}U+S^{\mathrm{ib,c}}-\nu S^{\mathrm{ib,v}}=0, \tag{23}$$

where the diagonal mass matrix $\mathcal{M}$ is built from the volume of the fluid cells, matrix $\mathcal{C}[\overline{U}]$ represents the discretization of the convective fluxes, $\mathcal{G}$ is the discrete pressure gradient, $\mathcal{K}$ represents the diffusive term, $S^{\mathrm{ib,c}}$ and $S^{\mathrm{ib,v}}$ are source terms arising from the boundary conditions of the convective and viscous terms, respectively. These different terms will now be constructed such that total momentum and kinetic energy are discretely conserved.

We consider first the conservation of the kinetic energy $E_{\mathrm{c}}(t)=\frac{1}{2}\int_{\Omega^{\mathrm{f}}}|\boldsymbol{v}|^2\,\mathrm{d}V$, that we discretize with the trapezoidal rule in each fluid cell $\Omega_{i,j}$:

$$E_{\mathrm{c}}(t)\cong E^h_{\mathrm{c}}(t)=\frac{1}{2}U^{\mathrm{T}}\mathcal{M}U+\frac{1}{2}U^{\mathrm{ib,T}}\mathcal{M}^{\mathrm{ib}}U^{\mathrm{ib}}, \tag{24}$$

where $\mathcal{M}$ is the diagonal mass matrix that appears in the discrete momentum equation (23), and $U^{\mathrm{ib,T}}\mathcal{M}^{\mathrm{ib}}U^{\mathrm{ib}}$ is the contribution of the boundary conditions, which are assumed to be steady. For each line (i,j) of the discrete system (23), the trapezoidal rule gives the value of the diagonal coefficient of the mass matrix in the horizontal and vertical directions:

$$[\mathcal{M}^x]_{\mathrm{P}}(i,j)=\frac{1}{2}V_{i,j}+\frac{1}{2}V_{i+1,j},\quad [\mathcal{M}^y]_{\mathrm{P}}(i,j)=\frac{1}{2}V_{i,j}+\frac{1}{2}V_{i,j+1}, \tag{25}$$

where the subscript P refers to the main diagonal elements in the usual compass notation (*e.g.* [17]). These expressions show that, in the cut-cells at least, the mass matrix for $u_{i,j}$ and $v_{i,j}$ is not constructed from the actual areas of $\Omega^u_{i,j}$ and $\Omega^v_{i,j}$.

The conservation equation for $E^h_c(t)$ is obtained after time-differentiation of (24), then by using the discrete momentum equation (23), we finally obtain:

$$\frac{dE^h_c}{dt} = -U^T \frac{\mathcal{C}[\overline{U}]^T + \mathcal{C}[\overline{U}]}{2} U - P^T \mathcal{G}^T U - U^T \frac{\nu(\mathcal{K}^T + \mathcal{K})}{2} U - U^T (S^{ib,c} - \nu S^{ib,v}). \tag{26}$$

In order to obtain an expression similar to (10), the viscous term $-U^T(\mathcal{K}^T + \mathcal{K})U$ should mimic the viscous dissipation of the kinetic energy budget, and thus should always be strictly negative. This feature is obtained as soon as the matrix $\mathcal{K}^T + \mathcal{K}$ is positive definite. For a finite-volume method, this is obtained as soon as the discrete diffusive flux is stable and consistent [13]. Note that the symmetry of $\mathcal{K}$ is not required. If, in addition, we impose that the discretization of the convective terms leads to a skew-symmetric matrix:

$$\mathcal{C}[\overline{U}] = -\mathcal{C}[\overline{U}]^T, \tag{27}$$

and that, as in the finite element method, the pressure gradient be dual to the divergence operator (see Eq. (18)):

$$\mathcal{G} = -\mathcal{D}^T, \tag{28}$$

we finally observe that the boundary terms only affects the kinetic energy budget when the viscosity vanishes:

$$\frac{dE^h_c}{dt} = -P^T \overline{U}^{ib} - U^T S^{ib,c}. \tag{29}$$

3.3.1. Discretization of the pressure gradient

In the above equation, the term $P^T\overline{U}^{ib}$ represents the discretization of the pressure term $\int_{\Gamma^{ib}} p\boldsymbol{v}\cdot\boldsymbol{n}\,dS$ in Eq. (10), and condition (28) allows us to completely determine the discrete pressure gradient in control volumes $\Omega^u_{i,j}$ and $\Omega^v_{i,j}$ from the discrete divergence operator (15):

$$\int_{\Gamma^u_{i,j}} p\boldsymbol{e}_x\cdot\boldsymbol{n}\,dS \cong [\mathcal{G}^x P]_{i,j} = \theta^u_{i,j}\Delta y_j (p_{i+1,j} - p_{i,j}), \tag{30a}$$

$$\int_{\Gamma^v_{i,j}} p\boldsymbol{e}_y\cdot\boldsymbol{n}\,dS \cong [\mathcal{G}^y P]_{i,j} = \theta^v_{i,j}\Delta x_i (p_{i,j+1} - p_{i,j}). \tag{30b}$$

These formulae are valid for any type of fluid cells, and in the particular case of Cartesian fluid cells (such that the cell-face fraction ratios are equal to 1), one recovers the finite-difference gradient of the MAC method:

$$[\mathcal{G}^x P]_{i,j} = \frac{p_{i+1,j} - p_{i,j}}{\frac{1}{2}\Delta x_{i+1} + \frac{1}{2}\Delta x_i} [\mathcal{M}^x]_P(i,j),$$

where $[\mathcal{M}^x]_P(i,j) = (\frac{1}{2}\Delta x_{i+1} + \frac{1}{2}\Delta x_i)\Delta y_j$ for the Cartesian control volume $\Omega^u_{i,j}$.

In the cut-cells however, it is not possible to interpret formulas (30) as finite-difference quotients for $p_{i,j}$ located at the centroids of the cut-cells. Instead, the LS-STAG discretization has much in common with the P1 nonconforming/P0 finite element method, where the pressure is approximated with a piecewise constant polynomial with degrees of freedom at the elements centroid [50]. As a consequence, $p_{i,j}$ is a valid approximation of the pressure *anywhere* inside cut-cell $\Omega_{i,j}$, even on its solid face. Note that an equivalence of this assumption in a Cartesian mesh is that the pressure gradient is zero at solid boundaries. In the next Section, we shall observe that the normal viscous stresses are discretized similarly.

3.3.2. Skew-symmetric discretization of the convective fluxes

For the x-momentum equation (2a), the skew-symmetry property (27) of $\mathcal{C}[\overline{U}]$ imposes upon the discretization of the convective term, that we write in a Cartesian control volume away from the immersed boundary as the following five-point scheme:

$$\int_{\Gamma^u_{i,j}} (\boldsymbol{v}\cdot\boldsymbol{n})u\,dS \cong \mathcal{C}[\overline{U}]_W(i,j)u_{i-1,j} + \mathcal{C}[\overline{U}]_E(i,j)u_{i+1,j} + \mathcal{C}[\overline{U}]_P(i,j)u_{i,j} + \mathcal{C}[\overline{U}]_S(i,j)u_{i,j-1} + \mathcal{C}[\overline{U}]_N(i,j)u_{i,j+1}, \tag{31}$$

must verify the following conditions:

$$\mathcal{C}[\overline{U}]_P(i,j) = 0, \tag{32a}$$

$$\mathcal{C}[\overline{U}]_E(i,j) = -\mathcal{C}[\overline{U}]_W(i+1,j), \tag{32b}$$

$$\mathcal{C}[\overline{U}]_N(i,j) = -\mathcal{C}[\overline{U}]_S(i,j+1). \tag{32c}$$

The central discretization of the MAC method trivially verifies these conditions on uniform meshes. Other popular discretization, such as discretizations of upwind type, are known to violate this condition, resulting in adding artificial vis-

cosity to the scheme. Recently, Verstappen and Veldman [63] have proposed a skew-symmetric discretization, also coined *symmetry preserving*, which enforces conditions (32c) on non-uniform Cartesian meshes. This discretization will be the building block of the LS-STAG discretization of the convective term in the cut-cells. For the Cartesian control volume $\Omega^u_{i,j}$ of Fig. 1, the skew-symmetric discretization consists in writing the convective term as the net flux through its four elementary faces:

$$\int_{\Gamma^u_{i,j}} (\boldsymbol{v}\cdot\boldsymbol{n})u\,\mathrm{d}S = -\int_{\Gamma^{u,\mathrm{w}}_{i,j}} (\boldsymbol{v}\cdot\boldsymbol{e}_x)u\,\mathrm{d}y + \int_{\Gamma^{u,\mathrm{e}}_{i,j}} (\boldsymbol{v}\cdot\boldsymbol{e}_x)u\,\mathrm{d}y - \int_{\Gamma^{\mathrm{s,e}}_{i,j}\cup\Gamma^{\mathrm{s,w}}_{i+1,j}} (\boldsymbol{v}\cdot\boldsymbol{e}_y)u\,\mathrm{d}x + \int_{\Gamma^{\mathrm{n,e}}_{i,j}\cup\Gamma^{\mathrm{n,w}}_{i+1,j}} (\boldsymbol{v}\cdot\boldsymbol{e}_y)u\,\mathrm{d}x, \tag{33}$$

Each of these terms are discretized with the help of the discrete mass fluxes (14), for example for the east face:

$$\int_{\Gamma^{u,\mathrm{e}}_{i,j}} (\boldsymbol{v}\cdot\boldsymbol{e}_x)u\,\mathrm{d}y \cong \frac{\bar{u}_{i,j}+\bar{u}_{i+1,j}}{2}u_{\mathrm{e}}, \tag{34}$$

where u_{e} is a characteristic value of u on $\Gamma^{u,\mathrm{e}}_{i,j}$, which has to be obtained by interpolation of the discrete velocity unknowns. As observed by Verstappen and Veldman, the only possible way to verify the skew-symmetric conditions (32c) is to use central interpolation with equal weighting:

$$u_{\mathrm{e}} = \frac{u_{i,j}+u_{i+1,j}}{2}. \tag{35}$$

Analogous interpolations are obtained on the other faces, for example on the south face:

$$\int_{\Gamma^{\mathrm{s,e}}_{i,j}\cup\Gamma^{\mathrm{s,w}}_{i+1,j}} (\boldsymbol{v}\cdot\boldsymbol{e}_y)u\,\mathrm{d}x \cong \frac{\bar{v}_{i,j-1}}{2}u_{\mathrm{s}} + \frac{\bar{v}_{i+1,j-1}}{2}u_{\mathrm{s}}, \tag{36}$$

with $u_{\mathrm{s}} = (u_{i,j-1}+u_{i,j})/2$. Now, by using the property of local conservativity of the fluxes through fluid faces:

$$\int_{\Gamma^{u,\mathrm{w}}_{i,j}} (\boldsymbol{v}\cdot\boldsymbol{e}_x)u\,\mathrm{d}y = \int_{\Gamma^{u,\mathrm{e}}_{i-1,j}} (\boldsymbol{v}\cdot\boldsymbol{e}_x)u\,\mathrm{d}y, \quad \int_{\Gamma^{\mathrm{n,e}}_{i,j}} (\boldsymbol{v}\cdot\boldsymbol{e}_y)u\,\mathrm{d}x = \int_{\Gamma^{\mathrm{s,e}}_{i,j+1}} (\boldsymbol{v}\cdot\boldsymbol{e}_y)u\,\mathrm{d}x, \tag{37}$$

one gets, after identification with (31), the following coefficients of the discretization:

$$\mathcal{C}[\overline{U}]_{\mathrm{P}}(i,j) = \frac{1}{4}\dot{m}^{i,j} + \frac{1}{4}\dot{m}^{i+1,j}, \tag{38a}$$

$$\mathcal{C}[\overline{U}]_{\mathrm{W}}(i,j) = -\frac{1}{4}\bar{u}_{i-1,j} - \frac{1}{4}\bar{u}_{i,j}, \quad \mathcal{C}[\overline{U}]_{\mathrm{E}}(i,j) = \frac{1}{4}\bar{u}_{i,j} + \frac{1}{4}\bar{u}_{i+1,j}, \tag{38b}$$

$$\mathcal{C}[\overline{U}]_{\mathrm{S}}(i,j) = -\frac{1}{4}\bar{v}_{i,j-1} - \frac{1}{4}\bar{v}_{i+1,j-1}, \quad \mathcal{C}[\overline{U}]_{\mathrm{N}}(i,j) = \frac{1}{4}\bar{v}_{i,j} + \frac{1}{4}\bar{v}_{i+1,j}, \tag{38c}$$

which verifies the antisymmetry conditions (32c) when the discrete continuity equation is verified in $\Omega_{i,j}$ and $\Omega_{i+1,j}$. Any type of interpolation other than (35), for example an upwind discretization, would violate theses conditions.

In the cut-cells, the skew-symmetric discretization given by (31) and (38c) must be modified in order to take into account the boundary conditions on the immersed boundary. This discretization would prove to be more complicated to build than for the pressure gradient, because we could not obtain a unique formula which would be valid for any type of cut-cells: instead, the discretization should be constructed in each of the half generic control volumes of Fig. 3 such as the skew-symmetry condition (27) be verified for any combinations of these half control volumes.

Let us consider the case of the control volume $\Omega^u_{i,j}$ of Fig. 2, whose north solid boundary $\Gamma^{\mathrm{ib,e}}_{i,j} \cup \Gamma^{\mathrm{ib,w}}_{i+1,j}$ is built from two halves of trapezoidal cut-cells. For this control volume, the discretization of the convective term must take the form:

$$\int_{\Gamma^u_{i,j}} (\boldsymbol{v}\cdot\boldsymbol{n})u\,\mathrm{d}S \cong \mathcal{C}[\overline{U}]_{\mathrm{W}}(i,j)u_{i-1,j} + \mathcal{C}[\overline{U}]_{\mathrm{E}}(i,j)u_{i+1,j} + \mathcal{C}[\overline{U}]_{\mathrm{P}}(i,j)u_{i,j} + \mathcal{C}[\overline{U}]_{\mathrm{S}}(i,j)u_{i,j-1} + S^{\mathrm{ib,c}}_{i,j}, \tag{39}$$

where $\mathcal{C}[\overline{U}]_{\mathrm{N}}(i,j)$ is discarded since the velocity unknown $u_{i,j+1}$ does not exist in the fluid domain. The skew-symmetry condition (27) reads for this control volume:

$$\mathcal{C}[\overline{U}]_{\mathrm{P}}(i,j) = 0, \quad \mathcal{C}[\overline{U}]_{\mathrm{E}}(i,j) = -\mathcal{C}[\overline{U}]_{\mathrm{W}}(i+1,j). \tag{40}$$

The discretization that verifies these conditions is obtained by decomposing the boundary of the control volume as in Eq. (7), and writing the convective term as the net flux through each of these faces:

$$\int_{\Gamma^u_{i,j}} (\boldsymbol{v}\cdot\boldsymbol{n})u\,\mathrm{d}S = -\int_{\Gamma^{u,\mathrm{w}}_{i,j}} (\boldsymbol{v}\cdot\boldsymbol{e}_x)u\,\mathrm{d}y + \int_{\Gamma^{u,\mathrm{e}}_{i,j}} (\boldsymbol{v}\cdot\boldsymbol{e}_x)u\,\mathrm{d}y - \int_{\Gamma^{\mathrm{s,e}}_{i,j}\cup\Gamma^{\mathrm{s,w}}_{i+1,j}} (\boldsymbol{v}\cdot\boldsymbol{e}_y)u\,\mathrm{d}x + \int_{\Gamma^{\mathrm{ib,e}}_{i,j}\cup\Gamma^{\mathrm{ib,w}}_{i+1,j}} (\boldsymbol{v}\cdot\boldsymbol{n}^{\mathrm{ib}})u\,\mathrm{d}S. \tag{41}$$

The fluxes through each of the fluid faces are given by Eqs. (34), (36) and (37), whereas the fluxes through each half of solid face $\Gamma^{\mathrm{ib,e}}_{i,j}$ and $\Gamma^{\mathrm{ib,w}}_{i+1,j}$ are discretized separately as:

$$\int_{\Gamma^{\mathrm{ib,e}}_{i,j}} \left(\boldsymbol{v}\cdot\boldsymbol{n}^{\mathrm{ib}}_{i,j}\right) u \,\mathrm{d}S \cong \frac{\overline{U}^{\mathrm{ib}}_{i,j}}{2}\left(\frac{1}{2}\underline{u_{i,j}} + \frac{1}{2}\underline{\underline{u\left(x_i, y^{\mathrm{ib}}_{i,j}\right)}}\right), \tag{42a}$$

$$\int_{\Gamma^{\mathrm{ib,w}}_{i+1,j}} \left(\boldsymbol{v}\cdot\boldsymbol{n}^{\mathrm{ib}}_{i+1,j}\right) u \,\mathrm{d}S \cong \frac{\overline{U}^{\mathrm{ib}}_{i+1,j}}{2}\left(\frac{1}{2}\underline{u_{i,j}} + \frac{1}{2}\underline{\underline{u\left(x_i, y^{\mathrm{ib}}_{i,j}\right)}}\right). \tag{42b}$$

In these expressions, the terms underlined once contribute to the diagonal coefficient $\mathcal{C}[\overline{U}]_{\mathrm{P}}(i,j)u_{i,j}$, in order to recover the expression of the discrete continuity in $\Omega_{i,j}$ and $\Omega_{i+1,j}$, whereas the terms which are twice underlined contribute to the source term $S^{\mathrm{ib,c}}_{i,j}$. As a result, the discretization of the convective term in this control volume is also given by (38c), with the exception that:

$$\mathcal{C}[\overline{U}]_{\mathrm{N}}(i,j) = 0, \quad S^{\mathrm{ib,c}}_{i,j} = \frac{1}{2}\left(\frac{\overline{U}^{\mathrm{ib}}_{i,j}}{2} + \frac{\overline{U}^{\mathrm{ib}}_{i+1,j}}{2}\right) u\left(x_i, y^{\mathrm{ib}}_{i,j}\right). \tag{43}$$

The antisymmetry conditions (40) is thus verified, and we may consider that the source term $S^{\mathrm{ib,c}}_{i,j}u_{i,j}$ that arises in the kinetic energy budget (29) corresponds to a discrete approximation of the term $\int_{\Gamma^{\mathrm{ib}}} |\boldsymbol{v}|^2 \boldsymbol{v}\cdot\boldsymbol{n}/2 \,\mathrm{d}S$ written on the solid boundary $\Gamma^{\mathrm{ib,e}}_{i,j} \cup \Gamma^{\mathrm{ib,w}}_{i+1,j}$ of the control volume.

For the other types of half control volumes depicted in Fig. 3, the skew-symmetric discretization is given in Appendix A. This discretization is built such that the skew-symmetry condition (27) is verified for any combinations of these half control volumes.

3.4. Momentum equation II: Discretization of the viscous fluxes based on the conservation of total momentum

For the x-momentum equation (2a), the viscous terms written in control volume $\Omega^u_{i,j}$ reads:

$$\int_{\Gamma^u_{i,j}} \nabla u \cdot \boldsymbol{n} \,\mathrm{d}S = \int_{\Gamma^u_{i,j}} \frac{\partial u}{\partial x}\boldsymbol{e}_x \cdot \boldsymbol{n} \,\mathrm{d}S + \int_{\Gamma^u_{i,j}} \frac{\partial u}{\partial y}\boldsymbol{e}_y \cdot \boldsymbol{n} \,\mathrm{d}S. \tag{44}$$

We aim to discretizing these terms in the cut-cells such that the simplicity of the five-point structure of the MAC method be preserved. We have found that the discretization of the viscous fluxes is much more intricate than for the convective fluxes. As a matter of fact, we will need to impose the conservation of total momentum (8b) at the discrete level for completely characterizing theses terms: this is equivalent to stating that the discretization of the viscous fluxes in the cut-cell be consistent with the discretization of the hydrodynamic forces (9) acting on the immersed boundary.

First and foremost, we had to make a distinction between the discretization of the normal stress fluxes, *e.g.* $\int_{\Gamma^u_{i,j}} \partial u/\partial x \boldsymbol{e}_x \cdot \boldsymbol{n} \,\mathrm{d}S$ and the shear stress fluxes, *e.g.* $\int_{\Gamma^u_{i,j}} \partial u/\partial y \boldsymbol{e}_y \cdot \boldsymbol{n} \,\mathrm{d}S$. This distinction is also observed in Cartesian fluid cells since, due to the staggering of the velocity unknowns on the MAC mesh, the normal stresses:

$$\left.\frac{\partial u}{\partial x}\right|_{i,j} = \frac{u_{i,j} - u_{i-1,j}}{\Delta x_i}, \quad \left.\frac{\partial v}{\partial y}\right|_{i,j} = \frac{v_{i,j} - v_{i,j-1}}{\Delta y_j}, \tag{45}$$

are naturally located at the center of cell $\Omega_{i,j}$, while the shear stresses:

$$\left.\frac{\partial u}{\partial y}\right|_{i,j} = \frac{u_{i,j+1} - u_{i,j}}{\frac{1}{2}\Delta y_{j+1} + \frac{1}{2}\Delta y_j}, \quad \left.\frac{\partial v}{\partial x}\right|_{i,j} = \frac{v_{i+1,j} - v_{i,j}}{\frac{1}{2}\Delta x_{i+1} + \frac{1}{2}\Delta x_i}, \tag{46}$$

are located at its upper right corner (see Fig. 5).

On the LS-STAG mesh, it is very natural to locate the shear stresses at the vertices of the cut-cells as shown on Fig. 6. Note that for the case of pentagonal cells, $\partial u/\partial y|_{i,j}$ and $\partial v/\partial x|_{i,j}$ are computed at distinct vertices.

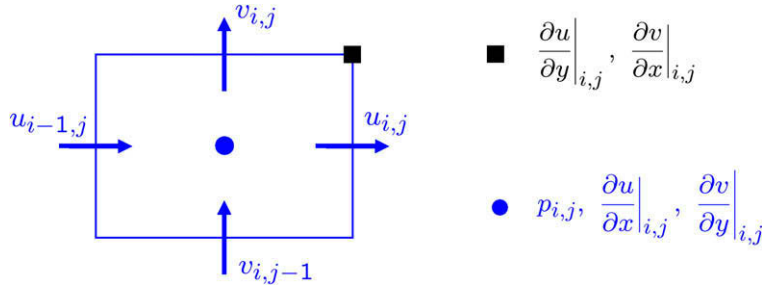


Fig. 5. Location of the pressure and viscous stresses in Cartesian cell $\Omega_{i,j}$.

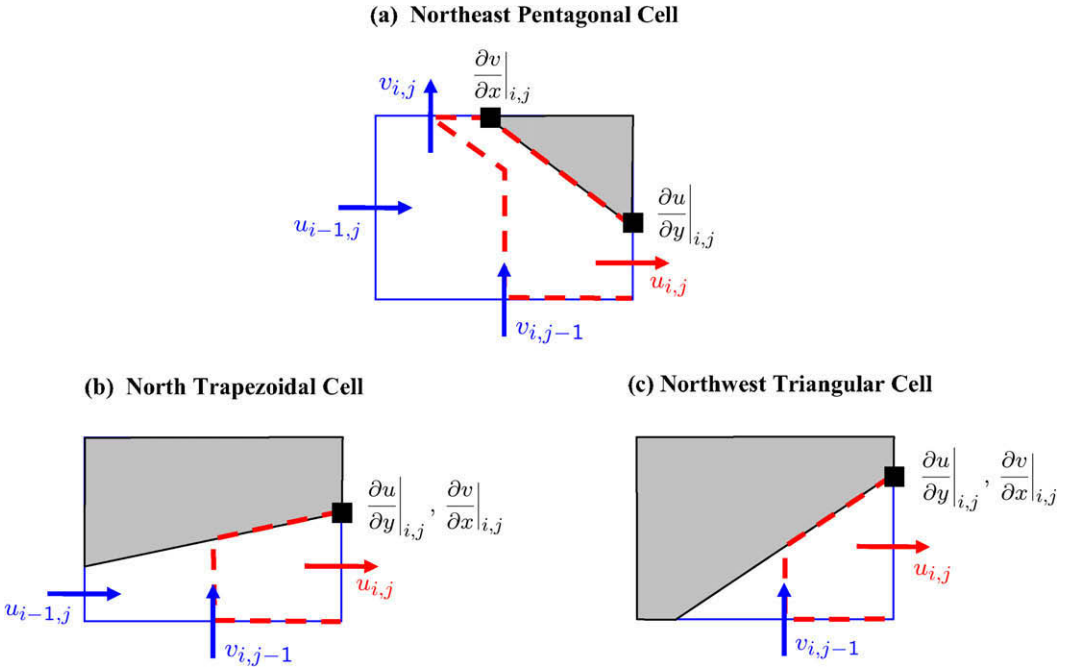


Fig. 6. Location of the shear stresses in the three generic cut-cells $\Omega_{i,j}$. The location on the other cut-cells of Fig. 3 can be easily deduced from this figure.

For the normal stresses in contrast, we consider that these terms have a physical origin (diagonal part of the Cauchy stress tensor) and a mathematical regularity similar to the pressure. In consequence, their treatment should be consistent with the pressure discretization discussed in the previous section: we assume that the normal stresses $\partial u/\partial x|_{i,j}$ and $\partial v/\partial y|_{i,j}$ take constant values in cut-cell $\Omega_{i,j}$, and do not need to be precisely located inside the cut-cell.

As a conclusion to this paragraph, we mention that the LS-STAG discretization of the viscous stresses has much in common with the finite element method for viscoelastic flows of Saramito [52], where the normal stresses are discretized with piecewise constant polynomials with a degree of freedom at the elements centroid, while the shear stresses are discretized with a linear continuous polynomial with degrees of freedom at the elements vertices.

3.4.1. Discretization of the normal stress fluxes

We now turn to the discretization of the normal stress flux $\int_{\Gamma^u_{i,j}} \partial u/\partial x \boldsymbol{e}_x \cdot \boldsymbol{n}\,\mathrm{d}S$ in the cut-cells. For this term, a geometric-based formula would consist in writing this term as the net flux through the east $\Gamma^{u,\mathrm{e}}_{i,j}$ and west $\Gamma^{u,\mathrm{w}}_{i,j}$ faces, and then discretize each of these terms with a differential quotient, for example in Fig. 6(a) and (b):

$$\int_{\Gamma^{u,\mathrm{w}}_{i,j}} \frac{\partial u}{\partial x}\boldsymbol{e}_x \cdot \boldsymbol{n}\,\mathrm{d}S \cong \Delta y^{u,\mathrm{w}}_{i,j} \frac{u_{i,j} - u_{i-1,j}}{\Delta x_i}, \tag{47}$$

where the area $\Delta y^{u,\mathrm{w}}_{i,j}$ is yet to be defined. All our efforts in this direction gave disappointing results in terms of numerical accuracy. The reason is that the LS-STAG mesh is not admissible in the sense of Eymard et al. [13] for the normal stresses in the cut-cells: the line joining the location of $u_{i-1,j}$ and $u_{i,j}$ is not orthogonal to the face $\Gamma^{u,\mathrm{w}}_{i,j}$ in the trapezoidal cell of Fig. 6(b). This feature is also observed for pentagonal cut-cells (see Fig. 6(a)), and has the consequence to render approximation (47) non-consistent and thus to yield large numerical errors.

In order to improve the consistency of this term, we use the fact that the discrete normal stresses should be consistent with the discrete pressure, as argued above, and thus the normal stress flux shall be discretized with an expression similar to the pressure gradient (30a):

$$\int_{\Gamma^u_{i,j}} \frac{\partial u}{\partial x}\boldsymbol{e}_x \cdot \boldsymbol{n}\,\mathrm{d}S \cong \theta^u_{i,j}\Delta y_j \left(\left.\frac{\partial u}{\partial x}\right|_{i+1,j} - \left.\frac{\partial u}{\partial x}\right|_{i,j} \right). \tag{48}$$

The discretization has to be completed with a differential quotient for $\partial u/\partial x|_{i,j}$. This quotient is constructed by requiring that Green's theorem:

$$\int_{\Omega_{i,j}} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \mathrm{d}V = \int_{\Gamma_{i,j}} \boldsymbol{v} \cdot \boldsymbol{n}\,\mathrm{d}S, \tag{49}$$

be valid at the discrete level in a cut-cell, since it is trivially verified by the MAC method in a Cartesian cell. After a straightforward discretization of the integrals and comparison with the continuity equation (15), one gets:

$$\left.\frac{\partial u}{\partial x}\right|_{i,j} \cong \frac{\theta^u_{i,j}u_{i,j} - \theta^u_{i-1,j}u_{i-1,j} + \left(\theta^u_{i-1,j} - \theta^u_{i,j}\right)u^{\mathrm{ib}}_{i,j}}{V_{i,j}/\Delta y_j}, \tag{50a}$$

and an analogous expression holds for $\partial v/\partial y|_{i,j}$:

$$\left.\frac{\partial v}{\partial y}\right|_{i,j} \cong \frac{\theta^v_{i,j}v_{i,j} - \theta^v_{i,j-1}v_{i,j-1} + \left(\theta^v_{i,j-1} - \theta^v_{i,j}\right)v^{\mathrm{ib}}_{i,j}}{V_{i,j}/\Delta x_i}. \tag{50b}$$

These expressions are valid for any type of cut-cells, with the boundary conditions naturally imbedded. These formulae reduce to the standard finite-difference quotients for a Cartesian fluid cell (see *e.g.* (45)).

3.4.2. A first discretization of the shear stress fluxes

In contrast, the discretization of the shear stress flux $\int_{\Gamma^u_{i,j}} \partial u/\partial y \boldsymbol{e}_y \cdot \boldsymbol{n}\,\mathrm{d}S$ may seem simpler because the LS-STAG mesh is admissible for this term (see Fig. 2). The shear stress term can thus be written as the net flux through the north and south faces, for example far from the immersed boundary:

$$\int_{\Gamma^u_{i,j}} \frac{\partial u}{\partial y}\boldsymbol{e}_y \cdot \boldsymbol{n}\,\mathrm{d}S = \int_{\Gamma^{\mathrm{n,e}}_{i,j}\cup\Gamma^{\mathrm{n,w}}_{i+1,j}} \frac{\partial u}{\partial y}\,\mathrm{d}x - \int_{\Gamma^{\mathrm{s,e}}_{i,j}\cup\Gamma^{\mathrm{s,w}}_{i+1,j}} \frac{\partial u}{\partial y}\,\mathrm{d}x. \tag{51}$$

Application of the midpoint rule gives for the north face for example:

$$\int_{\Gamma^{\mathrm{n,e}}_{i,j}\cup\Gamma^{\mathrm{n,w}}_{i+1,j}} \frac{\partial u}{\partial y}\,\mathrm{d}x \cong \left(\Delta x^{\mathrm{n,e}}_{i,j} + \Delta x^{\mathrm{n,w}}_{i+1,j}\right)\left.\frac{\partial u}{\partial y}\right|_{i,j}, \tag{52}$$

where for the purpose of local conservation of the fluxes, the areas $\Delta x^{\mathrm{n,e}}_{i,j}$ and $\Delta x^{\mathrm{n,w}}_{i+1,j}$ represent only the fluid part of the faces, *i.e.*:

$$\Delta x^{\mathrm{n,e}}_{i,j} = \frac{1}{2}\theta^v_{i,j}\Delta x_i, \quad \Delta x^{\mathrm{n,w}}_{i+1,j} = \frac{1}{2}\theta^v_{i+1,j}\Delta x_{i+1}. \tag{53}$$

The quotient $\partial u/\partial y|_{i,j}$, located at the upper right corner of cell $\Omega_{i,j}$ (see Fig. 5), is computed by differentiating the interpolation polynomial of $u(x_i,\cdot)$ in the vertical direction:

$$\left.\frac{\partial u}{\partial y}\right|_{i,j} = \frac{u_{i,j+1} - u_{i,j}}{\frac{1}{2}\theta^u_{i,j+1}\Delta y_{j+1} + \frac{1}{2}\theta^u_{i,j}\Delta y_j}. \tag{54}$$

This finite-difference quotient is much similar to the usual one (Eq. (46)). Formulae (52)–(54) are valid if $u_{i,j+1}$ is present in the fluid domain, *i.e.* if $\theta^u_{i,j+1} > 0$: this is the case of the Cartesian cell of Fig. 1(b), and the cut-cells in Fig. 3(a) and (b).

In the case where the north face is solid and thus $u_{i,j+1}$ does not exist (case where $\theta^u_{i,j+1} = 0$, for the cut-cells of Fig. 3(c)–(f)), these formulae have to be modified for taking into account the boundary conditions in the fashion of the *ghost fluid method* for elliptic equations [18]. For example, on the north face:

$$\int_{\Gamma^{\mathrm{ib,e}}_{i,j}\cup\Gamma^{\mathrm{ib,w}}_{i+1,j}} \frac{\partial u}{\partial y}\,\mathrm{d}x \cong \left(\Delta x^{\mathrm{ib,e}}_{i,j} + \Delta x^{\mathrm{ib,w}}_{i+1,j}\right)\left.\frac{\partial u}{\partial y}\right|_{i,j}, \tag{55}$$

with the one-sided differential quotient:

$$\left.\frac{\partial u}{\partial y}\right|_{i,j} = \frac{u\left(x_i, y^{\mathrm{ib}}_{i,j}\right) - u_{i,j}}{\frac{1}{2}\theta^u_{i,j}\Delta y_j}. \tag{56}$$

Note that in (55) the integration areas $\Delta x^{\mathrm{ib,e}}_{i,j}$ and $\Delta x^{\mathrm{ib,w}}_{i+1,j}$ on the solid face are yet to be defined.

For completeness, we mention that the discretization of the viscous term has a five-point structure, and in the case where $\theta^u_{i,j+1} = 0$, it has a form analogous to (39):

$$\int_{\Gamma^u_{i,j}} \nabla u \cdot \boldsymbol{n}\,\mathrm{d}S \cong \mathcal{K}_{\mathrm{W}}(i,j)u_{i-1,j} + \mathcal{K}_{\mathrm{E}}(i,j)u_{i+1,j} + \mathcal{K}_{\mathrm{P}}(i,j)u_{i,j} + \mathcal{K}_{\mathrm{S}}(i,j)u_{i,j-1} + S^{\mathrm{ib,v}}_{i,j}, \tag{57}$$

with:

$$\mathcal{K}_{\mathrm{W}}(i,j)=\frac{\theta^u_{i,j}\Delta y_j\theta^u_{i-1,j}}{V_{i,j}/\Delta y_j},\quad \mathcal{K}_{\mathrm{E}}(i,j)=\frac{\theta^u_{i,j}\Delta y_j\theta^u_{i+1,j}}{V_{i+1,j}/\Delta y_j}, \tag{58a}$$

$$\mathcal{K}_{\mathrm{S}}(i,j)=\frac{\theta^v_{i,j}\Delta x_i+\theta^v_{i+1,j}\Delta x_{i+1}}{\theta^u_{i,j}\Delta y_j+\theta^u_{i,j-1}\Delta y_{j-1}}, \tag{58b}$$

$$\mathcal{K}_{\mathrm{P}}(i,j)=-\frac{\left(\theta^u_{i,j}\right)^2\Delta y_j}{V_{i,j}/\Delta y_j}-\frac{\left(\theta^u_{i,j}\right)^2\Delta y_j}{V_{i+1,j}/\Delta y_j}-\mathcal{K}_{\mathrm{S}}(i,j)-\frac{\Delta x^{\mathrm{ib,e}}_{i,j}+\Delta x^{\mathrm{ib,w}}_{i+1,j}}{\frac{1}{2}\theta^u_{i,j}\Delta y_j}, \tag{58c}$$

$$S^{\mathrm{ib,v}}_{i,j}=\theta^u_{i,j}\Delta y_j\left[\frac{\left(\theta^u_{i,j}-\theta^u_{i+1,j}\right)}{V_{i+1,j}/\Delta y_j}u^{\mathrm{ib}}_{i+1,j}-\frac{\left(\theta^u_{i-1,j}-\theta^u_{i,j}\right)}{V_{i,j}/\Delta y_j}u^{\mathrm{ib}}_{i,j}\right]+\frac{\Delta x^{\mathrm{ib,e}}_{i,j}+\Delta x^{\mathrm{ib,w}}_{i+1,j}}{\frac{1}{2}\theta^u_{i,j}\Delta y_j}u\left(x_i,y^{\mathrm{ib}}_{i,j}\right). \tag{58d}$$

We observe that the discretization is symmetric (*i.e.* $\mathcal{K}_{\mathrm{W}}(i+1,j)=\mathcal{K}_{\mathrm{E}}(i,j)$), as it is the case for the MAC method.

For completing our discretization of the viscous terms, we have to define the values of the integration areas $\Delta x^{\mathrm{ib,e}}_{i,j}$ and $\Delta x^{\mathrm{ib,w}}_{i+1,j}$ for the shear stress flux at the boundary (55). These values will be determined independently for each type of cut-cells in Fig. 6, by requiring that the shear stress fluxes at the immersed boundary correspond to the discretization of the shear part of the hydrodynamic forces (9). This is equivalent to requiring that total momentum be discretely conserved on the LS-STAG mesh.

3.4.3. Discrete conservation of total momentum and computation of hydrodynamic forces

As done in Section 3.3 for the kinetic energy, the total momentum $\mathbf{P}(t)=\int_{\Omega^{\mathrm{f}}}\boldsymbol{v}\,\mathrm{d}V$ is discretized with the trapezoidal rule to give:

$$\mathbf{P}(t)\cong P^h(t)=\mathbb{1}^{\mathrm{T}}\mathcal{M}U+\mathbb{1}^{\mathrm{T}}\mathcal{M}^{\mathrm{ib}}U^{\mathrm{ib}}, \tag{59}$$

where $\mathbb{1}$ is the constant vector. The conservation equation for $P^h(t)$ is obtained by multiplication of the semi-discrete scheme (23) with vector $\mathbb{1}$:

$$\frac{\mathrm{d}P^h}{\mathrm{d}t}=-[\mathbb{1}^{\mathrm{T}}\mathcal{C}[\overline{U}]U+\mathbb{1}^{\mathrm{T}}S^{\mathrm{ib,c}}]-[\mathbb{1}^{\mathrm{T}}\mathcal{G}P-\mathbb{1}^{\mathrm{T}}v\left(\mathcal{K}U+S^{\mathrm{ib,v}}\right)]. \tag{60}$$

This expression is the semi-discrete version of Eq. (8b): the quadratic terms in the RHS correspond to the summation of the convective, pressure and viscous fluxes from all control volumes. Since the property of local conservativity of the fluxes holds at fluid faces, all terms cancel out except those appearing at solid boundary faces of the cut-cells. These remaining terms should correspond to the forces that act on the immersed boundary.

Most relevant to our discussion is the discretization of the viscous fluxes, and thus we will focus on the viscous force: the LS-STAG method we be qualified to be *momentum conserving* if the non-zero terms in sum $[\mathbb{1}^{\mathrm{T}}\mathcal{G}P-\mathbb{1}^{\mathrm{T}}v(\mathcal{K}U+S^{\mathrm{ib,v}})]$ correspond to the discretization of the hydrodynamic force (9). This discretization is obtained by approximating the surface integrals in (9b) and (9a), respectively, as:

$$F^h_x=\sum_{\text{Cut-cells}\Omega_{i,j}}[n_x\Delta S]^{\mathrm{ib}}_{i,j}\left(p_{i,j}-v\left.\frac{\partial u}{\partial x}\right|_{i,j}\right)-v\mathrm{Quad}^{\mathrm{ib}}_{i,j}\left(\frac{\partial u}{\partial y}\boldsymbol{e}_y\cdot\boldsymbol{n}\right), \tag{61a}$$

$$F^h_y=\sum_{\text{Cut-cells}\Omega_{i,j}}-v\mathrm{Quad}^{\mathrm{ib}}_{i,j}\left(\frac{\partial v}{\partial x}\boldsymbol{e}_x\cdot\boldsymbol{n}\right)+[n_y\Delta S]^{\mathrm{ib}}_{i,j}\left(p_{i,j}-v\left.\frac{\partial v}{\partial y}\right|_{i,j}\right). \tag{61b}$$

In these equations, the quadrature of the pressure and normal stress term has been performed by observing that these terms are constant in the cut-cells and using the midpoint rule; as a result the same formula is valid for all types of cut-cells. In contrast, the quadrature of the shear stresses (denoted $\mathrm{Quad}^{\mathrm{ib}}_{i,j}$) has to be adapted to each type of cut-cells. This quadrature, based on the location of the shear stresses in Fig. 6 and the trapezoidal rule, is fully described in Appendix B. For our discussion, it is relevant to note that the portion of drag and lift acting on the solid part of the trapezoidal cut-cell of Fig. 6(b) is, respectively:

$$F^h_x\Big|_{i,j}=\left(\theta^u_{i-1,j}-\theta^u_{i,j}\right)\Delta y_j\left[p_{i,j}-v\left.\frac{\partial u}{\partial x}\right|_{i,j}\right]-\frac{v\Delta x_i}{2}\left[\left.\frac{\partial u}{\partial y}\right|_{i-1,j}+\left.\frac{\partial u}{\partial y}\right|_{i,j}\right], \tag{62a}$$

$$F^h_y\Big|_{i,j}=-\frac{v\left(\theta^u_{i-1,j}-\theta^u_{i,j}\right)\Delta y_j}{2}\left[\left.\frac{\partial v}{\partial x}\right|_{i-1,j}+\left.\frac{\partial v}{\partial x}\right|_{i,j}\right]+\Delta x_i\left[p_{i,j}-v\left.\frac{\partial v}{\partial y}\right|_{i,j}\right]. \tag{62b}$$

Now, let us examine closely the viscous part of the global momentum equation (60), whose contribution in the x direction that reads:

$$[\mathbb{1}^{\mathrm{T}}\mathcal{G}P - \mathbb{1}^{\mathrm{T}}\nu(\mathcal{K}U + S^{\mathrm{ib,v}})]_{|x} = \sum_{\mathrm{CVs}\,\Omega^u_{i,j}} \int_{\Gamma^u_{i,j}} \left[p - \nu\frac{\partial u}{\partial x}\right]\boldsymbol{e}_x \cdot \boldsymbol{n}\,\mathrm{d}S - \int_{\Gamma^u_{i,j}} \nu\frac{\partial u}{\partial y}\boldsymbol{e}_y \cdot \boldsymbol{n}\,\mathrm{d}S, \tag{63}$$

should correspond to the drag force F^h_x given by (61a). The easiest part to inspect is the contribution of the normal stresses, since a unique formula (Eqs. (30) and (50)) is valid for these terms in all computational cells:

$$\sum_{\mathrm{CVs}\,\Omega^u_{i,j}} \theta^u_{i,j}\Delta y_j \left(p_{i+1,j} - p_{i,j} - \nu\left[\left.\frac{\partial u}{\partial x}\right|_{i+1,j} - \left.\frac{\partial u}{\partial x}\right|_{i,j}\right]\right).$$

After reindexation for making appear a sum on the computational cells, we observe that pressure and normal stresses cancel out in fluid cells such that $\theta^u_{i,j} = \theta^u_{i-1,j} = 1$, and only the following terms in the cut-cells do remain:

$$\sum_{\text{Cut-cells}\,\Omega_{i,j}} \left(\theta^u_{i-1,j} - \theta^u_{i,j}\right)\Delta y_j \left(p_{i,j} - \nu\left.\frac{\partial u}{\partial x}\right|_{i,j}\right).$$

This sum is exactly the contribution of the normal stresses to the discrete drag force (61a). A similar inspection holds for the lift component (61b), ensuring thus that the normal stress contribution to the total momentum budget is recovered.

For the shear stress contribution, the fluxes at all fluid faces cancel out and only fluxes at the immersed boundary remains in sum (63), their exact expression depending on the type of cut-cell. On the solid part $\Gamma^{\mathrm{ib,w}}_{i,j} \cup \Gamma^{\mathrm{ib,e}}_{i,j}$ of cut-cell $\Omega_{i,j}$ in Fig. 2, we use Eq. (55) for writing the shear stress contribution as:

$$-\nu\left[\Delta x^{\mathrm{ib,w}}_{i,j}\left.\frac{\partial u}{\partial y}\right|_{i-1,j} + \Delta x^{\mathrm{ib,e}}_{i,j}\left.\frac{\partial u}{\partial y}\right|_{i,j}\right].$$

After comparison with the drag force (62a), we are able to unambiguously determine the integration areas of the shear stress as:

$$\Delta x^{\mathrm{ib,w}}_{i,j} = \Delta x^{\mathrm{ib,e}}_{i,j} = \frac{1}{2}\Delta x_i. \tag{64}$$

A similar inspection for the other type of cut-cells gives the values reported in Table 2 for the shear stress flux $\int_{\Gamma^{\mathrm{ib,e}}_{i,j}\cup\Gamma^{\mathrm{ib,w}}_{i+1,j}} \partial u/\partial y\,\mathrm{d}x$. Similar values can be deduced for the shear stress flux in the y-momentum equation $\int_{\Gamma^{\mathrm{ib,n}}_{i,j}\cup\Gamma^{\mathrm{ib,s}}_{i,j+1}} \partial v/\partial x\,\mathrm{d}y$. This will finish the description of what will be subsequently be called the "original" LS-STAG discretization of the viscous stress.

3.4.4. A "complete" discretization of the shear stresses based on the strict conservation of total momentum

However, if we further investigate the correspondence between the shear stress fluxes at the immersed boundary in the momentum equation and the discrete shear force (61), one may notice slight discrepancies. For example in the trapezoidal cut-cell of Fig. 6(b), the shear stress fluxes at the immersed boundary that appear in the momentum equation for $u_{i,j}$ and $v_{i,j-1}$ are, respectively:

$$-\frac{\nu\Delta x_i}{2}\left[\left.\frac{\partial u}{\partial y}\right|_{i-1,j} + \left.\frac{\partial u}{\partial y}\right|_{i,j}\right] \quad \text{and} \quad 0. \tag{65}$$

If we compare these terms with the shear force in (62a) and (62b), we observe that the following part is missing from the momentum equation for $v_{i,j-1}$:

$$-\frac{\nu\left(\theta^u_{i-1,j} - \theta^u_{i,j}\right)\Delta y_j}{2}\left[\left.\frac{\partial v}{\partial x}\right|_{i-1,j} + \left.\frac{\partial v}{\partial x}\right|_{i,j}\right]. \tag{66}$$

Note that the contribution of this term to the global lift force should be small, since it vanishes when the solid face is horizontal $\left(\theta^u_{i-1,j} = \theta^u_{i,j}\right)$. Nevertheless, the "original" LS-STAG method does not completely conserve the global momentum. The

Table 2
Integration areas for the shear stress flux (55) on the solid part of the cut-cell of Fig. 6. The integration areas for the other cut-cells of Fig. 3 can be easily deduced from the ones displayed.

Type of cut-cell	$\Delta x^{\mathrm{ib,w}}_{i,j}$	$\Delta x^{\mathrm{ib,e}}_{i,j}$
(a) Northeast pentagonal cell	0	$\left(1 - \theta^v_{i,j}\right)\Delta x_i$
(b) North trapezoidal cell	$\frac{1}{2}\Delta x_i$	$\frac{1}{2}\Delta x_i$
(c) Northwest triangular cell	$\frac{1}{2}\theta^v_{i,j-1}\Delta x_i$	$\frac{1}{2}\theta^v_{i,j-1}\Delta x_i$

reason is that for building this discretization we have assumed that the shear stress discretization yields a five-point stencil, and thus the contribution of terms $\partial v/\partial x|_{i-1,j}$ and $\partial v/\partial x|_{ij}$ should be ignored in the stencil for v_{ij-1}.

An alternate version of the LS-STAG method would be to retain terms such as (66) in the discretization. The difficulty is now to compute the new terms that appear in the discrete momentum equations. For example, the term $\partial v/\partial x|_{ij}$ in Fig. 6(b) may only be computed if we know the type of the neighboring cut-cell $\Omega_{i+1,j}$. If $\Omega_{i+1,j}$ is pentagonal, as in Fig. 4(d), then the shear stress can be computed with the one-sided formula:

$$\left.\frac{\partial v}{\partial x}\right|_{ij} = \frac{v_{i+1,j} - v\left(x^{\mathrm{ib}}_{ij}, y_j\right)}{\frac{1}{2}\theta^{v}_{i+1,j}\Delta x_{i+1}}. \tag{67}$$

The only other possibility is that $\Omega_{i+1,j}$ be a pentagonal cell, as in Fig. 4(a). In this case, we cannot use formula (67). Instead, we simply set $\partial v/\partial x|_{ij} = 0$, as it is verified when the solid face is horizontal.

This alternate version, the we will subsequently call the "complete" LS-STAG method, has the ability to be strictly momentum conserving. However, compared to the "original" LS-STAG method, the presence of the supplementary shear stress terms on the solid boundary severely complicates the coding of the method, and it also enlarges the size of the stencil. In the computations presented in the next section, these terms will be handled with an explicit time-stepping. These computations will also prove that the "complete" LS-STAG method shows only a marginal improvement over the "original" method, and we recommend thus the latter method.

3.5. Time stepping method and solution of the linear systems

The time integration of the differential algebraic system (18) and (23) is performed with a semi-implicit projection method based on the Adams–Bashforth/second-order backward differentiation formula (AB/BDF 2) scheme. This projection scheme is defined by the following two steps:

$$\mathcal{M}\frac{3\widetilde{U} - 4U^n + U^{n-1}}{2\Delta t} + 2\mathcal{C}[\overline{U}^n]U^n - \mathcal{C}[\overline{U}^{n-1}]U^{n-1} - \mathcal{D}^{\mathrm{T}}P^n - \nu\mathcal{K}\widetilde{U} = 0, \tag{68}$$

where $\widetilde{U}$ is a prediction of the velocity at time $t_{n+1} = (n+1)\Delta t$, then:

$$\frac{3}{2}\mathcal{M}\frac{U^{n+1} - \widetilde{U}}{\Delta t} - \mathcal{D}^{\mathrm{T}}(P^{n+1} - P^n) = 0, \tag{69a}$$

$$\mathcal{D}U^{n+1} + \overline{U}^{\mathrm{ib},n+1} = 0. \tag{69b}$$

Numerical tests in [6] shows that this scheme is $\mathcal{O}(\Delta t^2)$ accurate for both velocity and pressure.

The projection step (69) leads to solving following Poisson equation for the pressure potential $\Phi = 2\Delta t(P^{n+1} - P^n)/3$:

$$\mathcal{A}\Phi = \mathcal{D}\widetilde{U} + \overline{U}^{\mathrm{ib},n+1}, \quad \mathcal{A} \equiv -\mathcal{D}\mathcal{M}^{-1}\mathcal{D}^{\mathrm{T}}, \tag{70}$$

which is a symmetric linear system whose five-point stencil reads:

$$A_{\mathrm{E}}(i,j) = \frac{\left(\theta^{u}_{ij}\Delta y_j\right)^2}{\frac{1}{2}V_{ij} + \frac{1}{2}V_{i+1,j}}, \quad A_{\mathrm{W}}(i,j) = A_{\mathrm{E}}(i-1,j), \tag{71a}$$

$$A_{\mathrm{N}}(i,j) = \frac{\left(\theta^{v}_{ij}\Delta x_i\right)^2}{\frac{1}{2}V_{ij} + \frac{1}{2}V_{ij+1}}, \quad A_{\mathrm{S}}(i,j) = A_{\mathrm{N}}(i,j-1), \tag{71b}$$

$$A_{\mathrm{P}}(i,j) = -A_{\mathrm{E}}(i,j) - A_{\mathrm{W}}(i,j) - A_{\mathrm{N}}(i,j) - A_{\mathrm{S}}(i,j). \tag{71c}$$

In the case of a Cartesian fluid cell, the usual pressure equation of the MAC method is recovered. We mention that the pressure equation (70) is valid in the whole computational domain, fluid or solid cells alike, and in the latter cells the linear system reads:

$$0 \times \Phi_{ij} = 0, \tag{72}$$

which is a consequence of the fact that the pressure is defined up to an additive constant. In order to alleviate this indetermination in actual computations, we add to the diagonal coefficient $A_{\mathrm{P}}(i,j)$ a small real constant δ whose magnitude has the order of the machine roundoff, and we solve (70) in the whole computational domain with a standard solver for elliptic equations on Cartesian grids. In the computations we present next, we have used the black-box multigrid/BiCGSTAB solver of van Kan et al. [62]. No modifications of the solver had to be undertaken for taking the immersed boundary into account, since we did not observe a significant loss in the performance of the solver when compared to Cartesian computations. For the simulations presented in the following sections, the pressure equation was typically solved in 2–3 iterations with this solver.

The prediction step (68) amounts to solving a Helmholtz equation for $\widetilde{U}$, which is performed by the same black-box multigrid/BiCGSTAB solver. This linear system is easier to solve than the pressure equation since its diagonal dominance increases with the Reynolds number.

4. Numerical results

4.1. Taylor–Couette flow

First, the spatial accuracy of the LS-STAG method is assessed on the Taylor–Couette flow between two concentric circular cylinders, as described in Fig. 7(left). The flow dynamics is governed by the Taylor number Ta, which is the ratio between the centrifugal force and the viscous force:

$$Ta = \frac{\omega^2 \left(\frac{R_1+R_2}{2}\right)(R_2-R_1)^3}{\nu^2}. \tag{73}$$

Below the stability threshold $Ta_c = 1712$ [23], the steady stable solution is purely orthoradial, such that its Cartesian components read for $r = \sqrt{(x-x_c)^2+(y-y_c)^2} \in [R_1, R_2]$:

$$u_{ex}(x,y) = -K\left(\frac{R_2^2}{r^2}-1\right)(y-y_c), \tag{74a}$$

$$v_{ex}(x,y) = K\left(\frac{R_2^2}{r^2}-1\right)(x-x_c), \tag{74b}$$

$$p_{ex}(x,y) = K^2\left(\frac{r^2}{2}-\frac{R_2^4}{2r^2}-R_2^2\log r^2\right), \tag{74c}$$

where $K = \frac{\omega R_1^2}{R_2^2-R_1^2}$.

In order to build the level-set function $\phi(x,y)$ that represents the fluid domain Ω_f, we have used the *Constructive Solid Geometry* (*CSG*) method for constructing complex domains out of basic geometries such as circles, hyperplanes and spheres, which are sufficiently simple for having an analytical expression for their level-set function [25,47]. The boolean *CSG* operations on basic geometries such as intersection, union or complementary part can then be expressed as algebraic operations on their level-set functions [47]. For example, let Ω_1 and Ω_2 be the inner region of cylinders Γ_1 and Γ_2, whose level-set function is, respectively:

$$\phi_1(x,y) = R_1 - r, \tag{75a}$$

$$\phi_2(x,y) = R_2 - r. \tag{75b}$$

Then, the fluid domain of the Taylor–Couette geometry can be constructed as $\Omega_f = \Omega_2 \setminus \Omega_1$, and its level-set function is simply $\phi(x,y) = \max(\phi_2(x,y), -\phi_1(x,y))$.

The computational domain is a square of side length $10R_1$, covered with a uniform mesh of N square computational cells of size h in each direction (see Fig. 7(right)). The center of the concentric cylinders is set at $x_c = 0.013,\ y_c = 0.023$ slightly off the center of the computational domain, such as it never corresponds to a corner or centroid of a computational cell. Thus, the numerical error we measure are free of any superconvergence effects, since the natural symmetries of the meshes and

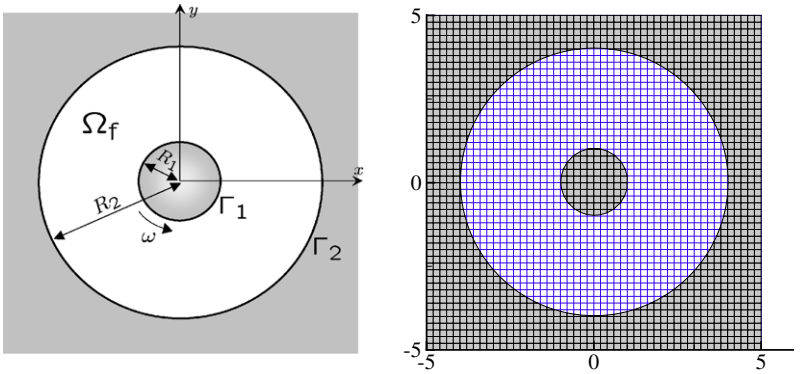


Fig. 7. Geometry and computational domain for the Taylor–Couette flow. At left, the fluid domain Ω_f is confined between two concentric cylinders Γ_1 and Γ_2 of center (x_c, y_c), radius R_1 and $R_2 = 4R_1$, respectively, where only the inner cylinder Γ_1 moves with the angular velocity ω. At right, the *LS-STAG* mesh for $R_1 = 1$ and $N = 50$ cells in each direction.

the computational solution are broken. For the flow at $Ta = 1000$, we have compared the results of the two variants of the LS-STAG method (original and complete, see Section 3.4) with the so-called *staircase* method, which corresponds to a stepwise approximation of complex geometries with Cartesian cells. This last method is easily obtained from our numerical code by imposing the cell-face fraction ratios to be equal to 1 in the cut-cells, while the discretization in the Cartesian cells is unchanged. We will see that even though the cut-cells represent only a small fraction of the computational cells, these modifications will greatly affect the numerical solution in the whole fluid domain.

For the streamwise velocity u, we have measured the L_∞ spatial error up to the cut-cells, *i.e.*:

$$E_h(u) = \max_{\text{CVs } \Omega^u_{i,j}} \left| u_{i,j} - u_{\text{ex}}\left(x_i, y_j + \frac{1}{2}\theta^u_{i,j}\Delta y_j\right)\right|, \tag{76}$$

whereas for the pressure, due to the indefinite location of the pressure in the cut-cells, we have reported the L_∞ error in the Cartesian cells only. Fig. 8(left) shows the error for the velocity measured on 90% of the fluid cells away from the immersed boundaries. The two LS-STAG variants shows a similar second-order accuracy, much better than the first-order accuracy of the staircase method. When the error of the LS-STAG method is measured on the whole computational domain (Fig. 8(right)), the L_∞ error is slightly higher, showing that maximal error occurs in the vicinity of the cut-cells, and the order of accuracy drops to being superlinear only. This is certainly an effect of the piecewise constant approximation of the normal stresses and the pressure in the cut-cells. A similar trend is observed for the L_∞ error of the pressure displayed in Fig. 9. Fig. 10 shows the pointwise error at mesh points along the horizontal radius, for $x \in [R_1, R_2]$. Firstly, we observe that the crude treatment of the immersed boundaries for the staircase method pollutes the solution in the whole fluid domain, whereas for the LS-STAG methods the largest errors stay confined to the vicinity of the immersed boundaries, and most notably to the inner moving cylinder Γ_1. Secondly, we observe that the complete variant of the LS-STAG method (which conserves global momentum) gives only a marginal improvement in terms of pointwise accuracy.

4.2. Flow past a circular cylinder

The robustness of the LS-STAG method and its ability to compute unsteady flows at higher Reynolds number is now evaluated on the flow past a circular cylinder in a free-stream. The Reynolds number Re is based on the free-stream velocity U_∞ and the diameter D of the domain. The flow configuration is described in Fig. 11(left). In all our simulations, the upstream boundary is set at the distance $X_u = 8D$ from the obstacle, the outflow boundary at distance $X_d = 15D$, and the blockage ratio D/A is equal to $1/12$. Our previous studies [7,8] have shown that this computational domain was sufficiently wide for obtaining results that are independent of the domain size. In order to make a grid refinement study, we used a sequence of non-uniform meshes whose salient properties are summarized in Table 3. All meshes use a similar block uniform grid of cell size h/D in the vicinity of the cylinder, as shown in Fig. 11(right). Our simulations with mesh M4 were found to give accurate results for the range of Reynolds number $[40 - 1000]$ we considered. The other meshes range from very coarse (mesh M1 with only 12 cut-cells around the immersed boundary, which is used for the steady flow at $Re = 40$) to extremely fine (mesh M5 with more than 400 cut-cells). This last mesh is mainly used for validating the results obtained on coarser meshes. Table 3 also reports the proportion of the various type of computational cells (solid, Cartesian or cut-cells) found in these meshes. We observe that the proportion of solid cells is very low and diminishes as the meshes gets bigger. Thus, the additional CPU and memory costs that their treatment entails is negligible compared to a body-fitted method. In the meshes we used for the

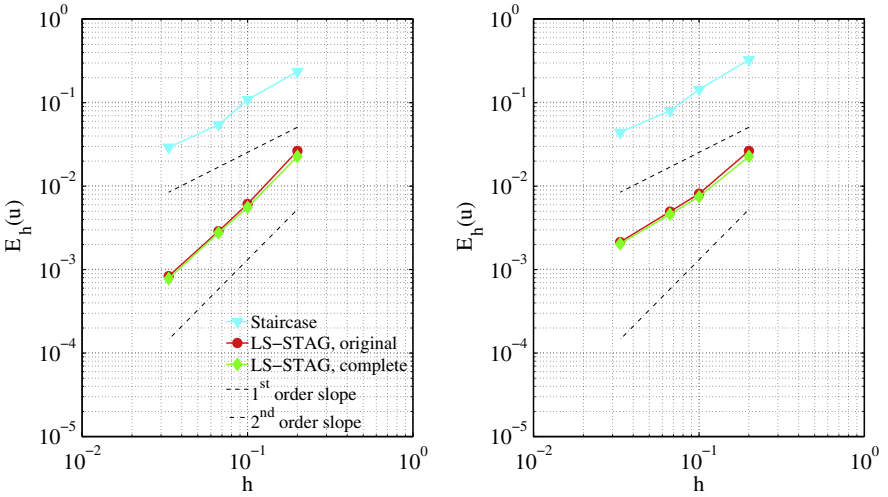


Fig. 8. L_∞ norm of the error for the streamwise velocity u versus grid size $h = \frac{10R_1}{N}$. At left: on 90% of the fluid domain; at right: on the whole fluid domain.

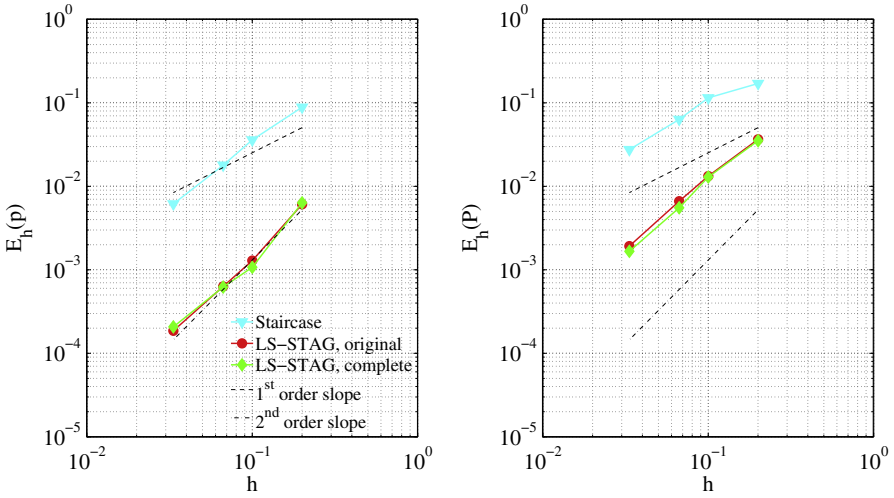


Fig. 9. L_∞ norm of the error for pressure p versus grid size $h = \frac{10R_1}{N}$. At left: on 65% of the fluid domain; at right on 90% of fluid domain.

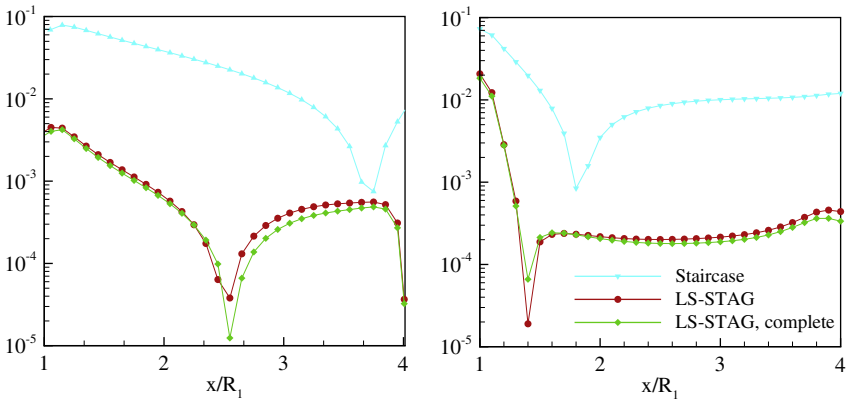


Fig. 10. Pointwise error profiles along the horizontal axis $y = 0.023$ for the v velocity (left) and the pressure (right).

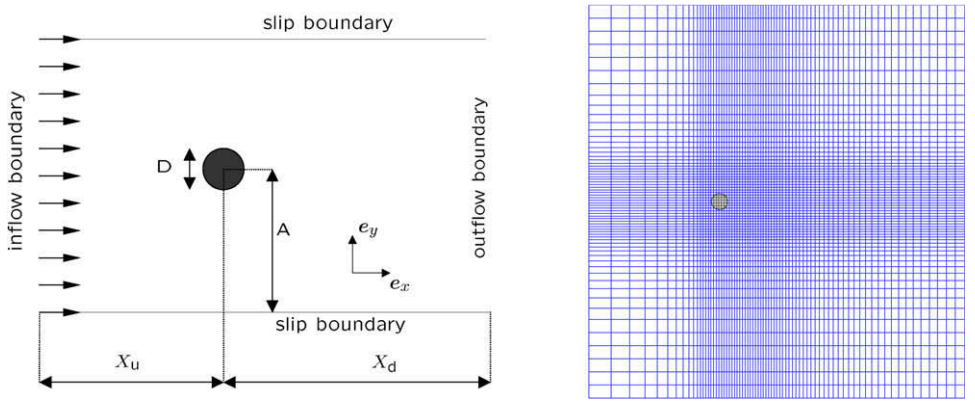


Fig. 11. Computational domain and grid for the flow past a circular cylinder.

Taylor–Couette flow, the proportion of cut-cells was similar, while the proportion of solid cells was around 50% whatever the size of the mesh. At last, Table 4 shows that the cut-cells almost have an equal proportion of triangular, trapezoidal and pentagonal cells: thus, an accurate discretization should be performed for all cells in Fig. 3.

Table 3
Salient properties of the meshes used for the circular cylinder flow. The percentage of the various type of cells with respect to their total number is given in brackets. The values of the time step are valid for all computations of Sections 4.2 and 4.3.

Mesh	$N_x \times N_y$	h/D	$\Delta t U_\infty/D$	Type of cells			
				Number of cells	Cartesian cells	Solid cells	Cut-cells
M1	36×34	0.32	10^{-2}	1224	1208 (98.7%)	4 (0.3%)	12 (1.0%)
M2	74×65	0.16	10^{-2}	4810	4767 (99.1%)	19 (0.4%)	24 (0.5%)
M3	150×130	0.08	10^{-2}	19,500	19,350 (99.2%)	100 (0.5%)	50 (0.3%)
M4	300×260	0.04	10^{-2} [a]	78,000	77,460 (99.3%)	440 (0.6%)	100 (0.1%)
M5	550×350	0.01	2×10^{-3} [b]	192,500	184,452 (95.8%)	7644 (4.0%)	404 (0.2%)

[a] $\Delta t = 5 \times 10^{-3} D/U_\infty$ for the computations at $Re = 1000$.
[b] $\Delta t = 10^{-3} D/U_\infty$ during the initial transients ($tU_\infty/D \in [0, 1]$) for the computations at $Re = 1000$.

Table 4
Inventory of the three basic types of cut-cells found in the Taylor–Couette (T-C) and circular cylinder (C-C) meshes, and their percentage with respect to the total number of cut-cells.

Simulation	Total number of cut-cells	Type cut-cells		
		Triangular cells	Trapezoidal cells	Pentagonal cells
T-C, $N = 100$	400	116 (29.0%)	168 (42.0%)	116 (29.0%)
T-C, $N = 300$	1192	344 (28.9%)	504 (42.3%)	344 (28.9%)
C-C, M2	24	4 (16.7%)	12 (50.0%)	8 (33.3%)
C-C, M4	100	28 (28.0%)	40 (40.0%)	32 (32.0%)

For comparing our results on the circular cylinder flow, we have selected well-established numerical investigations conducted with boundary-fitted [27,5,26], IB [35] and cut-cell methods [40]. Experimental results are taken from the monograph by Zdravkovich [67]. Following the derivation of the CFL condition on unstructured mesh in [4], the CFL number in fluid cell $\Omega_{i,j}$ is:

$$\mathrm{CFL}_{i,j} = \left[(\bar{u}_{i,j})^+ + (-\bar{u}_{i-1,j})^+ + (\bar{v}_{i,j})^+ + (-\bar{v}_{i,j-1})^+ + \left(\overline{U}^{\mathrm{ib}}_{i,j}\right)^+ \right] \frac{\Delta t}{V_{i,j}}, \tag{77}$$

where $(\cdot)^+ = \max(\cdot, 0)$. This condition reduces to the conventional CFL condition in Cartesian fluid cells. For the values of the time step given in Table 3, all computations were performed with a maximal value of CFL equal to 0.5. We have observed that this value is not obtained in a cut-cell, but rather in a Cartesian cell away from the wake region. This shows that the occurrence of small cut-cells in the mesh does not hamper the stability properties of our semi-implicit time-stepping.

First, we performed a grid convergence study of the various methods (staircase, LS-STAG and LS-STAG complete) for the steady flow at $Re = 40$. For both variants of the LS-STAG method, the forces were computed with the same quadrature (Eq. (61)). Fig. 12 shows the spatial convergence of the drag coefficient $C_D = F_x/\frac{1}{2}\rho U_\infty^2$ and the length of the recirculation bubble L_w/D. The staircase method gives very inaccurate results on the coarser meshes (no recirculation zone is observed on mesh M1), whereas the LS-STAG methods gives acceptable results for all meshes. As shown in Table 5, the results of both LS-STAG methods compare well with established results from the literature (which are typically in the range $[1.50 - 1.54]$) and, once again, the difference between the two variants is undistinguishable. From now on, we will only report computations performed with the original LS-STAG method.

Unsteady flows at $Re = 100$, 200 and 1000 have been computed on the M4 and M5 meshes. For breaking the symmetry of the flow and efficiently triggering the vortex shedding, we use as initial condition a discontinuous flow field equal to U_∞ in the upper half of the domain, and 0 in the lower half. The flow reaches an asymptotically periodic state at $t = 50D/U_\infty$, then we start computing the force coefficients at each time step until $t = 350D/U_\infty$. The Strouhal number St is computed as the first harmonic of the power spectrum of the lift coefficient, with a frequency resolution of $\pm 1.67 \times 10^{-3}$ since the length of the time signal is equal to 300 units. Tables 6 and 7 give salient results computed with the LS-STAG and staircase method. On the M4 mesh for $Re = 100$ and 200 and on the M5 mesh for $Re = 1000$, the LS-STAG method gives excellent agreement with the published results. It is also quite remarkable to observe that the staircase method gives marginally acceptable results, even for $Re = 1000$. This is certainly due to the fact that the staircase method inherits the conservation and stability properties of the LS-STAG method, and only the treatment of the immersed boundary differs.

4.3. Flow past a circular cylinder with forced oscillatory rotations

For assessing the validity of our treatment of non-homogeneous boundary conditions, we have compute the flow past a circular cylinder with forced rotary oscillations given by the angular velocity:

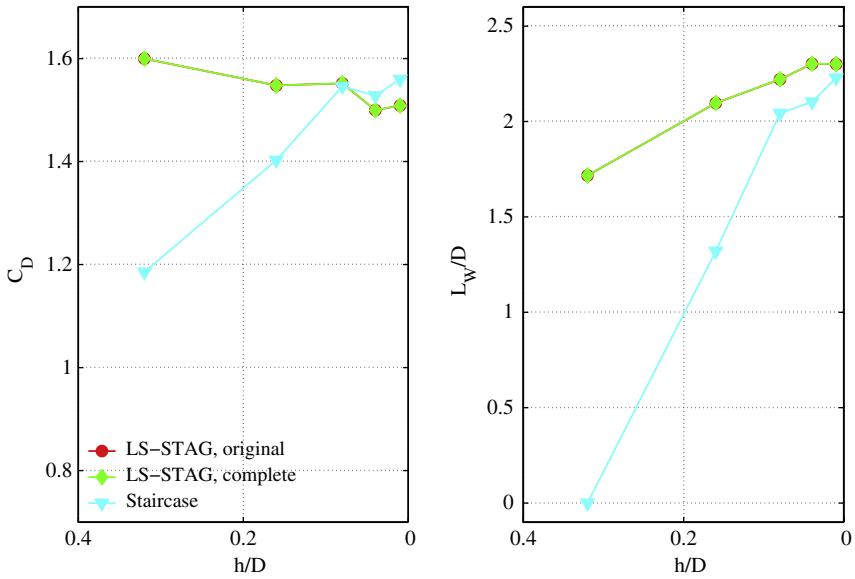


Fig. 12. Drag coefficient and recirculation bubble length for the steady flow at $Re = 40$ computed with the various meshes of Table 3.

Table 5
Results for the flow at $Re = 40$ obtained on the two finest meshes and comparison with established results from the literature.

	L_w/D	C_D
M4, original	2.300	1.500
M4, complete	2.300	1.500
M4, staircase	2.101	1.527
M5, original	2.299	1.508
M5, complete	2.299	1.508
M5, staircase	2.226	1.559
Experiments [67]	–	1.48–1.70
Bergmann et al. [5]	2.26	1.682
Henderson [27]	–	1.545
He et al. [26]	–	1.505
Linnick and Fasel [35]	2.23	1.54
Mittal et al. [40]	–	1.53

Table 6
Comparison of time averaged drag coefficient $\overline{C}_D$ and corresponding oscillation amplitude $\pm\Delta C_D$ with established results from the literature.

Re	100	200	1000
M4	1.322 ± 0.009	1.332 ± 0.044	1.493 ± 0.227
M5	1.317 ± 0.009	1.327 ± 0.045	1.530 ± 0.229
M4, staircase	1.323 ± 0.009	1.346 ± 0.044	1.610 ± 0.198
Experiments [67]	1.21–1.41	–	–
Bergmann et al. [5]	1.410	1.390	1.505
Henderson [27]	1.350	1.341	1.509
He et al. [26]	1.353	1.356	1.519
Linnick and Fasel [35]	1.34 ± 0.009	1.34 ± 0.044	–
Mittal et al. [40]	1.35	–	1.45

$$\omega(t) = \omega_0 \sin\left(2\pi S_e t \frac{U_\infty}{D}\right), \tag{78}$$

where the two forcing parameters are the amplitude ω_0 of the angular velocity and the forcing Strouhal number S_{rme}. This forcing has been introduced by Tokumaru and Dimotakis [58], who observed experimentally for the flow at $Re = 15,000$ a drag reduction as high as 80% for certain values of these forcing parameters. In the computational studies of Refs. [26,5], these two parameters have been used as control parameters for drag minimisation in the 2D laminar regime. For the flow

at $Re = 200$ and $Re = 1000$, the optimal control of the Navier–Stokes equations by He et al. [26] yielded a drag reduction of about 30% and 60%, respectively, compared to the unforced flow ($\omega_0 = 0, S_e = 0$). More recently, Bergmann et al. [5] used a less computationally demanding control algorithm, which employed a reduced-order model of the Navier–Stokes equations, and also observed a 30% drag reduction at $Re = 200$ for slightly differently forcing parameters (($\omega_0 = 8.50, S_e = 0.74$) instead of ($\omega_0 = 6.0, S_e = 0.74$) found by He et al. [26]).

We have undertaken the computation of the forced flow at $Re = 200$ and $Re = 1000$ for these various sets of parameters with the LS-STAG and staircase methods. The computational parameters and grids are the same as in the previous Section. For $Re = 200$ first, Fig. 13 compares the time histories of the drag coefficient of the unforced, forced (at the optimal values of Ref. [26]) and basic flow, where the basic flow is the steady, symmetric but unstable flow which generates *a priori* the lowest drag force ($C_D^{basic} = 0.864$). As in [26,5], we observe that the forced flow observes a locking regime, where the natural shedding frequency is given by the forcing frequency, that greatly reduces the drag force on the cylinder. Table 8 summarizes the drag forces computed at various values of the forcing parameters. This table shows that the LS-STAG method predicts maximal drag reduction for $Re = 200$ at the optimal values of He et al. [26] and, in addition, we mention that an overall excellent agreement is observed with the computations of these authors, for both unforced (Table 6) and forced flows. For future reference, we report in Table 9 the peak value of the lift coefficient for the various flows we have computed. Finally, Fig. 14 displays the vortex shedding observed in the unforced, forced and basic flow at $Re = 1000$. The characteristic vortex shed-

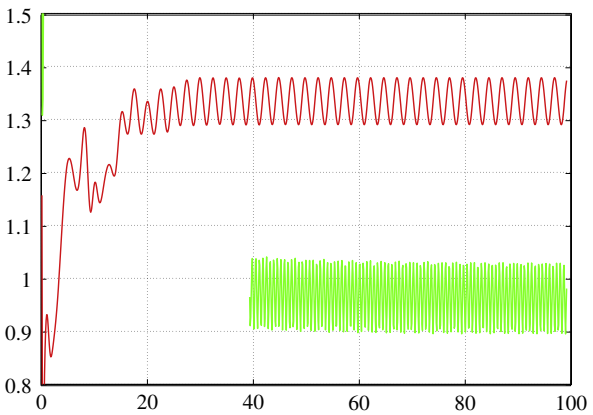


Table 9
Peak value of the lift coefficient C_L^{max} for unforced and forced flows at various Reynolds number.

Re	100	200			1000	
$(\omega_0; S_e)$	(0; 0)	(0; 0)	(6.0; 0.74)	(8.50; 0.74)	(0; 0)	(5.50; 0.625)
M4	0.348	0690	0.358	0.510	1.305	0.054
M5	0.349	0.710	0.337	0.491	1.482	0.103
M4, staircase	0.332	0.669	0.330	0.537	1.332	1.037
Norberg [44]	0.28–0.41	0.56–0.85	–	–	–	–
Linnick and Fasel [35]	0.333	0.70	–	–	–	–

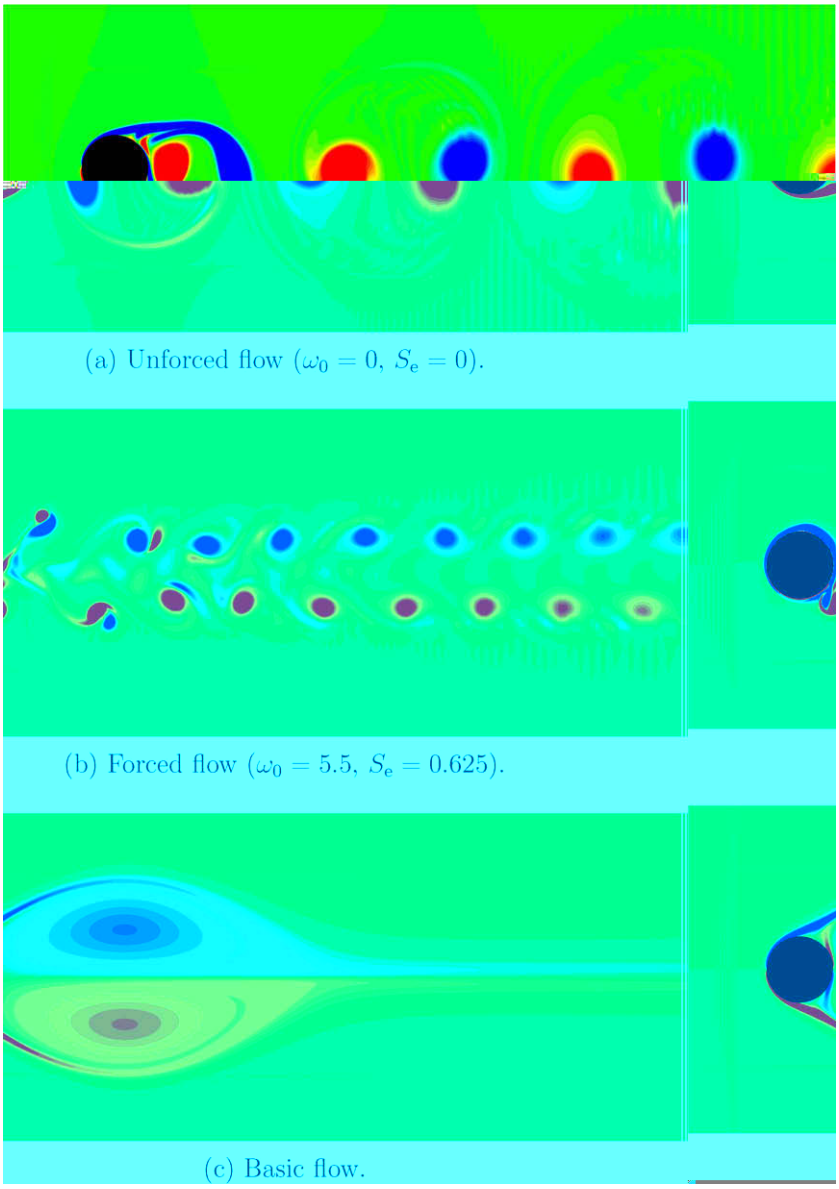


Fig. 14. Vorticity contours in the wake of the cylinder at $Re = 1000$, computed by the LS-STAG method on the M5 mesh.

ding of the unforced flow has been dramatically altered by the optimal oscillatory rotations, leading to a quasi-symmetric flow pattern which is much different than the unstable steady solution. This last figure shows also that vorticity production is confined to the near vicinity of the cylinder, and further illustrates the high fidelity treatment of the immersed boundary of

the LS-STAG method. The staircase method, on the other hand, gives inaccurate values of the force signals amplitude (see Tables 8 and 9), and its suitability for being the basic solver in an optimal control loop is questionable.

4.4. Application to flows with immersed moving boundaries

In order to show the versatility of the LS-STAG method, we report in this section a first attempt for computing flows with moving immersed boundaries. Indeed, the computation of such flows on fixed Cartesian grids may become a competitive alternative, in terms of CPU time and accuracy, to body-conformal methods that need frequent mesh and solution updates during the flow computation (see the reviews [11,57] and references therein). First, we present the main modifications of the LS-STAG method for handling moving boundaries. Then, we apply the method on a flow with prescribed boundary motion: the transverse oscillation of a cylinder in a free-stream [20,22].

4.4.1. Modifications of the LS-STAG method for immersed moving boundaries

The LS-STAG method for moving boundaries has common ground with the arbitrary Lagrangian–Eulerian (ALE) method widely used by finite-volume methods on body-conformal meshes [11]. The ALE method considers a computational domain that tracks the moving boundary, such that the grid follows the boundary motion in the Lagrangian fashion, while it is held fixed in a Eulerian manner sufficiently far from it. The ALE equations of the flow motion written in the computational domain are similar to the Navier–Stokes equations (1) and (2) except that the convective term now reads:

$$\int_{\Gamma(t)} ([\boldsymbol{v}-\boldsymbol{v}^{\mathrm{g}}]\cdot\boldsymbol{n})\boldsymbol{v}\,\mathrm{d}S,$$

where $\boldsymbol{v}^{\mathrm{g}}$ is the computational domain (or grid) velocity, which is related to the arbitrary motion of the mesh nodes. In a series of papers, Farhat and co-workers [33,21,16,15] have shown that a ALE numerical scheme preserves the accuracy and stability of its fixed grid counterpart if the grid velocity and mesh update procedure are designed such that the so-called *geometric conservation law* (GCL):

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega(t)}\mathrm{d}V=\int_{\Gamma(t)}\boldsymbol{v}^{\mathrm{g}}\cdot\boldsymbol{n}\,\mathrm{d}S, \tag{79}$$

is enforced at the discrete level. This equation is obtained by requiring that a uniform flow is solution to the ALE equations, and states that the change in volume of a cell during a time interval must be equal to the volume that has been swept by the cell boundary.

The enforcement of the GCL is a challenging task when the whole computational grid is deformed by the motion of its boundary. For the LS-STAG method in contrast, this condition is easier to fulfill because only the grid nodes on the solid part $\Gamma^{\mathrm{ib}}_{i,j}(t)$ of a cut-cell are moving. Thus, the ALE convective flux through the solid faces of control volume $\Omega^{u}_{i,j}(t)$ is:

$$\int_{\Gamma^{\mathrm{ib,e}}_{i,j}(t)\cup\Gamma^{\mathrm{ib,w}}_{i+1,j}(t)}\left([\boldsymbol{v}-\boldsymbol{v}^{\mathrm{g}}]\cdot\boldsymbol{n}^{\mathrm{ib}}_{i,j}\right)u\,\mathrm{d}S, \tag{80}$$

while the expression on the other faces is unchanged since the normal motion of the fluid cell faces is zero. The discretization of (80) follows the line of the skew-symmetric discretization of Section 3.3 and Appendix A. For example, the equivalent of Eq. (42a) for the half solid face $\Gamma^{\mathrm{ib,e}}_{i,j}(t)$ of control volume $\Omega^{u}_{i,j}(t)$ in Fig. 2 is:

$$\int_{\Gamma^{\mathrm{ib,e}}_{i,j}(t)}\left([\boldsymbol{v}-\boldsymbol{v}^{\mathrm{g}}]\cdot\boldsymbol{n}^{\mathrm{ib}}_{i,j}\right)u\,\mathrm{d}S\cong\frac{\overline{U}^{\mathrm{ib}}_{i,j}-\overline{U}^{\mathrm{g}}_{i,j}}{2}\left(\frac{1}{2}u_{i,j}+\frac{1}{2}u\left(x_i,y^{\mathrm{ib}}_{i,j}\right)\right), \tag{81}$$

where $\overline{U}^{\mathrm{g}}_{i,j}$ is the discrete grid flux for cut-cell $\Omega_{i,j}(t)$, that will be defined such that a discrete GCL be enforced.

For our computations with moving boundaries, we have considered a first-order variant of the projection scheme (68) and (69), whose prediction step reads:

$$\frac{\mathcal{M}^{n+1}\widetilde{U}-\mathcal{M}^{n}U^{n}}{\Delta t}+\mathcal{C}[\overline{U}^{n}-\overline{U}^{\mathrm{g},n}]U^{n}-\nu\mathcal{K}^{n+1}\widetilde{U}=0, \tag{82}$$

where the coefficients of the scheme are now time-dependent and have to be computed with the level-set of the immersed boundary at the corresponding time level. The discrete grid flux $\overline{U}^{\mathrm{g},n}$ is then computed by requiring that the constant solution $U=\mathbb{1}$ be a solution to the discrete scheme (82), as it is already observed on fixed grids. In the cut-cells, this requirement leads to:

$$\frac{[\mathcal{M}^{x}]^{n+1}_{\mathrm{P}}(i,j)-[\mathcal{M}^{x}]^{n}_{\mathrm{P}}(i,j)}{\Delta t}-\frac{\overline{U}^{\mathrm{g},n}_{i,j}+\overline{U}^{\mathrm{g},n}_{i+1,j}}{2}=0, \tag{83}$$

and, by using the definition (25) of the mass matrix, we completely determine the discrete grid flux for cut-cell $\Omega^{n+1}_{i,j}$ as:

$$\overline{U}^{g,n}_{i,j} = \frac{V^{n+1}_{i,j} - V^{n}_{i,j}}{\Delta t}. \tag{84}$$

This relation is the discrete version of the GCL (79) for the implicit Euler scheme (82); the projection step (69) does not alter this condition. At each time cycle, the discrete grid flux $\overline{U}^{g,n}_{i,j}$ can be easily computed from the discrete GCL and incorporated to the convective discretization of Appendix A in the fashion of Eq. (81).

Another issue which is mostly relevant to discretizations on fixed Cartesian grid is the occurrence of "freshly cleared" cells, fluid cells which were inside the solid at the previous time step. In these cells, the use of conventional time-stepping is not possible, and several ad hoc procedure are proposed in [60,61,3,9] for approximating the flow variables. In the present paper, if the control volume $\Omega^{u,n+1}_{i,j}$ is "freshly cleared" (*i.e.* $\theta^{u,n}_{i,j} = 0$ and $\theta^{u,n+1}_{i,j} > 0$), then we simply set the provisional velocity $\tilde{u}_{i,j}$ equal to the boundary velocity instead of using the prediction step (82), while the projection step is unchanged. Note that this simple procedure does not guaranty the discrete conservation of momentum or kinetic energy. A more accurate procedure, left for future work, would be to use in the freshly cleared cells a fully implicit semi-Lagrangian time-stepping [50].

Nevertheless, we have checked the spatial accuracy of the present implementation of the method by designing a test problem with moving boundaries as done in Refs. [61,65]. We have considered a rigid circular cylinder of diameter D that oscillates horizontally in a square computational domain of dimension $3D \times 3D$ with no-slip walls, such that the prescribed motion of the cylinder center reads:

$$\dot{x}_c(t) = -U_{\max} \sin\left(2\pi \frac{U_{\max}}{D}\right), \quad \dot{y}_c(t) = 0, \tag{85}$$

$$x_c(0) = \frac{D}{2\pi}, \quad y_c(0) = 0, \tag{86}$$

where $U_{\max}$ is the maximal velocity. Simulations have been performed at $Re = U_{\max} D/\nu = 100$ on four uniform grids: 30×30, 90×90, 150×150 and 450×450, the latter being used as the reference solution. A small time step of $\Delta t = 10^{-4} D/U_{\max}$ has been chosen in order to avoid temporal errors effect. The flow has been simulated for one oscillation period, and the L_∞ norm of the spatial error obtained at the end of the computations is shown in Fig. 15. This figure clearly shows that first-order accuracy is obtained for both component of the velocity. Thus, we observe only a slight decrease in accuracy compared to the case of stationary immersed boundaries, where superlinear convergence in the L_∞ norm was observed (see Fig. 8).

4.4.2. Application: transversely oscillating cylinder in a free-stream

To validate our method we computed the flow past a cylinder undergoing prescribed transverse oscillation in a free-stream. This flow has first been studied experimentally by Gu et al. [20]. Numerical computations have then been carried out by Lu and Dalton [37] and Guilmineau and Queutey [22] with a body-conformal method in the inertial frame of reference. Subsequently, this flow has become a popular benchmark for both IBM [65,31] and cut-cell methods [61,9].

The computational domain and boundary conditions are the same as for the stationary cylinder flow of Section 4.2. The flow parameters match the ones used in the above references. The transverse motion of the cylinder center is given by:

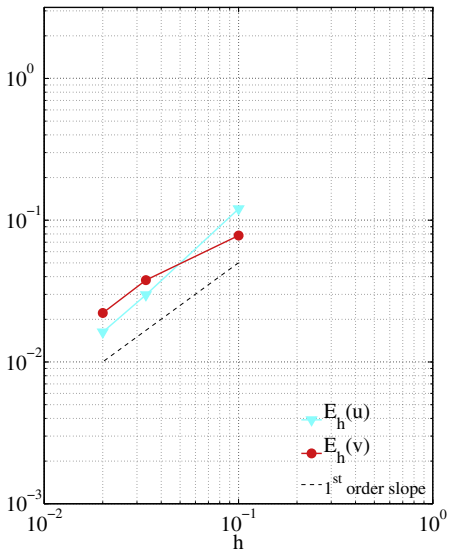


Fig. 15. L_∞ norm of the error for the horizontally oscillating cylinder test problem.

$$y_c(t) = A, \quad t < 10\frac{D}{U_\infty}, \tag{87a}$$

$$y_c(t) = A\cos\left(2\pi S_e\left[t\frac{U_\infty}{D} - 10\right]\right), \quad t \geqslant 10\frac{D}{U_\infty}, \tag{87b}$$

with the oscillations amplitude $A = 0.2D$ and S_e is the exciting Strouhal number. We considered a series of computations on the M4 mesh (see Table 3) at $Re = 185$, for $A = 0.2D$ and $S_e/S_0 \in [0.8, 1.2]$ where S_0 is the natural frequency of the vortex shedding, which has been computed as $S_0 = 0.201$ with the ER1 time-stepping. The value of the time step is $\Delta t = 5 \times 10^{-3} D/U_\infty$.

When the immersed body is moving, the CFL number (77) has to be corrected with the grid velocity as done in Eq. (81) for the ALE convective fluxes:

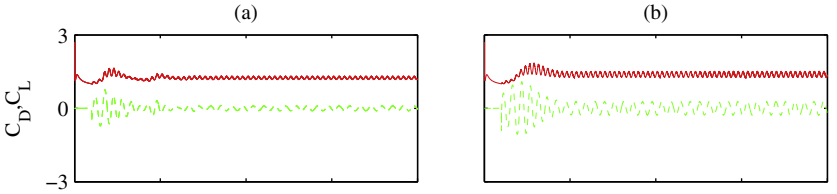
$$\mathrm{CFL}_{i,j} = \left[(\bar{u}_{i,j})^+ + (-\bar{u}_{i-1,j})^+ + (\bar{v}_{i,j})^+ + (-\bar{v}_{i,j-1})^+ + \left(\overline{U}^{\mathrm{ib}}_{i,j} - \overline{U}^{\mathrm{g}}_{i,j}\right)^+\right]\frac{\Delta t}{V_{i,j}}. \tag{88}$$

In addition, we mention that the fluid velocities in the cut-cells take values close to the velocity of the immersed boundary. Hence, in order to avoid numerical instabilities caused by the motion of the immersed body, a CFL number based on the kinematics of the IB has to be constructed, for example [19]:

$$\mathrm{CFL}^{\mathrm{ib}}_{i,j} = \frac{\left|u^{\mathrm{ib}}_{i,j}\right|\Delta t}{\Delta x_i} + \frac{\left|v^{\mathrm{ib}}_{i,j}\right|\Delta t}{\Delta y_j}. \tag{89}$$

The condition $\max_{i,j}\mathrm{CFL}^{\mathrm{ib}}_{i,j} < 1$ ensures that the body never traverses an entire computational cell within one time step. Since the motion of the immersed body is prescribed in the present work, this condition can be verified prior to the computations. We have verified that the kinematics CFL condition based on (89) is always much less restrictive than the fluid dynamics condition (88), and thus does not hamper the stability of the method: for example, for the case $S_e/S_0 = 1.2$ we observed that $\max_{i,j}\mathrm{CFL}^{\mathrm{ib}}_{i,j} \simeq 0.04$ while $\max_{i,j}\mathrm{CFL}_{i,j} \simeq 0.3$.

Fig. 16 shows the time evolution of the force coefficients for increasing values of the forcing frequencies. In particular, we observe a frequency "lock on" at all values of the forcing frequencies, and the presence of a higher harmonic when $S_e/S_0 \geqslant 1$. These results agree very well with previous studies [37,22,65]. To give a more quantitative evaluation of this series of simulations, Fig. 17 shows that the mean force coefficients are in very good agreement with both body-conformal (in the non-inertial frame of the cylinder) [22] and IBM [65] computations. Finally, for the case $S_e/S_0 = 1.2$ that shows the most complex dynamics, Table 10 displays a grid refinement study for the computation of the force coefficients. In order to keep the time



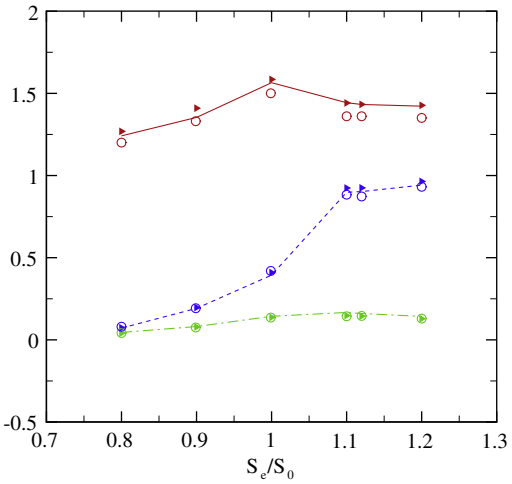


Fig. 17. Time averaged force coefficients for the flow past an oscillating cylinder in a free-stream: $\overline{C}_D$ (solid line), C_D^{rms} (dash-dot line) and C_L^{rms} (dot line). The symbols 'filled right triangle' and 'open circle' represent data obtained by Yang and Balaras [65] and Guilmineau and Queutey [22], respectively.

Table 10
Time averaged force coefficients for the case $S_e/S_0 = 1.2$ obtained on the series of meshes described in Table 3.

	$\overline{C}_D$	C_D^{rms}	C_L^{rms}
M1	1.309	0.366	0.884
M2	1.202	0.101	0.754
M3	1.386	0.124	0.889
M4	1.422	0.142	0.941
M5	1.387	0.132	0.975
IBM, 800×640 cells, $h = 0.005D$ [65]	1.426	0.128	0.964
O-type curvilinear grid, 180×120 cells, $h = 0.001D$ [22]	1.35	0.129	0.931

discretization error (which is of first-order for moving boundaries problems) comparable for all the meshes, all simulations have been performed with a CFL number equal to 0.3. We observes that on the 2 finest meshes, the discrepancy of the results is inferior to 3% for the mean drag coefficient, and 4% for the root-mean-square lift coefficient. This level of grid sensitivity is comparable to the one observed in [22].

Finally, we have examined the quality of our computations of the local forces on the moving body's surface. In effect, the accuracy and smoothness of their computation is of great importance for the application of the LS-STAG method to fluid–structure interaction problems. Fig. 18 displays the comparison of the local pressure and vorticity distributions with the reference data of Guilmineau and Queutey [22] obtained on a body-fitted grid. As Yang and Balaras [65] observed with their IBM computations, it is the local shear that is the most sensitive to grid resolution, most notably in the vicinity of local extremas. These discrepancies however diminish with mesh refinement, giving a smooth and accurate distribution on the M5 mesh. We also mention the body-fitted grid of Ref. [22] has a local grid spacing normal to the boundary equal to $h = 0.001D$, which is 10 times smaller than the resolution of our finest mesh (M5, see Table 3).

4.5. Evaluation of the computational efficiency of the LS-STAG method

There are essentially two features that make immersed boundary methods potentially more attractive than traditional body fitted methods: the quick design of quality meshes for highly complex geometries [36] and the lower computational cost of the flow simulations. This last feature is well documented for simple geometries, for which Cartesian flow solvers takes only a fraction of the CPU time of the more versatile body fitted solvers [38]. In contrast, the CPU cost of IBM simulations with a Cartesian solver is not so well documented. The additional overhead for handling complex geometries depends greatly on the IBM treatment and the cleverness of the code implementation, and in some cases may take a substantial part of the solution procedure. For the LS-STAG method, we observed that this overhead is associated to the computation of the geometric parameters of the fluid cells when constructing the discrete systems, and to the additional stiffness of the systems caused by the presence of small cut-cells.

This part is devoted to evaluating the current performance of the LS-STAG code used for all the computations presented in this article, and compare them with a widely available commercial code: *FLUENT*. We have chosen a well established benchmark problem: the unsteady flow past a square cylinder at $Re = 100$. For this geometry, both codes are able to use the same

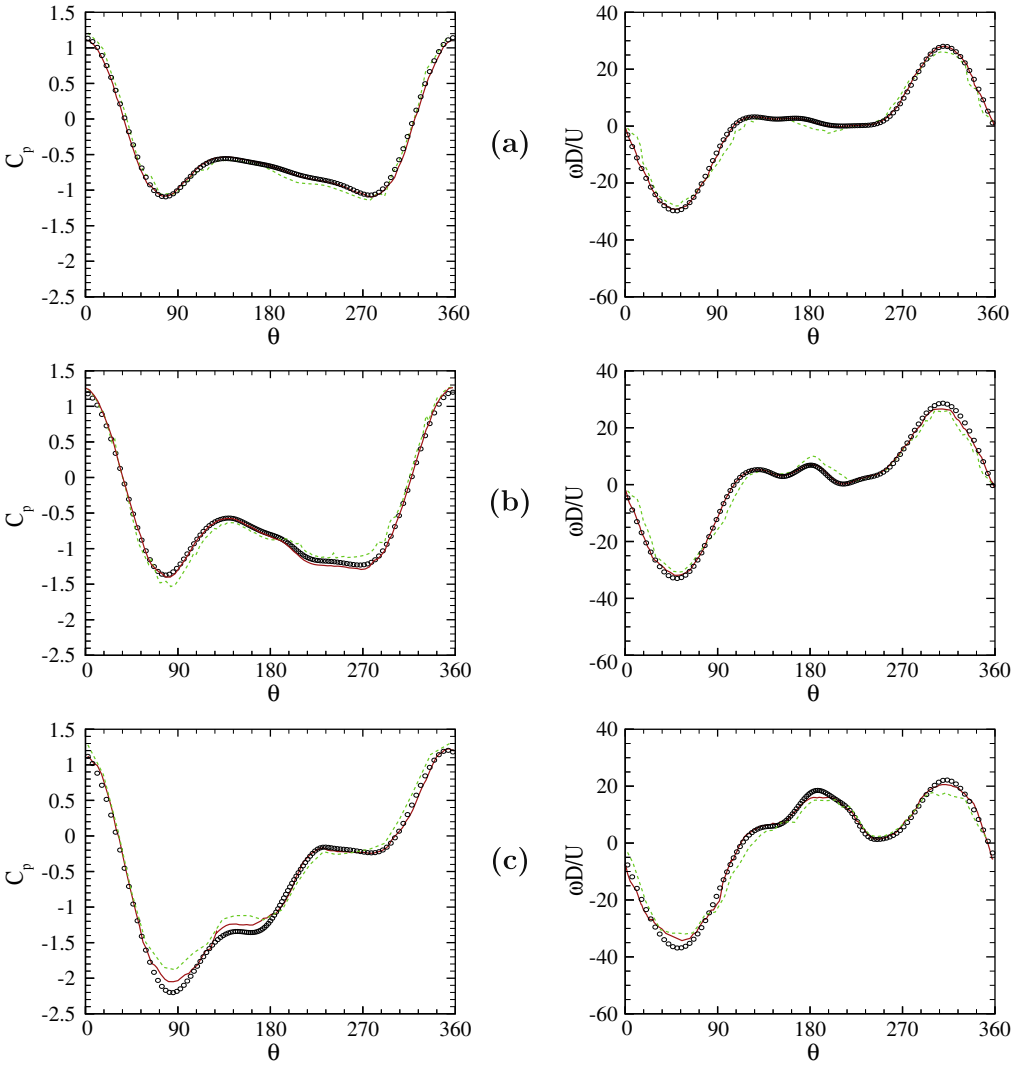


Fig. 18. Distribution of the pressure coefficient and vorticity on the cylinder's surface when it is located at its extreme upper position for S_e/S_0 equals to: (a) 0.8; (b) 1.0; (c) 1.2. The results have been obtained on two meshes: (- - -) M4 mesh; (——) M5 mesh. The symbols ∘ represent reference data obtained on a body fitted grid by Guilmineau and Queutey [22].

computational parameters and boundary-fitted Cartesian mesh (note that in the LS-STAG code the fluid cells adjacent to the cylinder boundary are treated as cut-cells). The computational domain and boundary conditions are the same as the circular cylinder computations of Section 4.2, with $X_u = 8.5D$, $X_d = 20D$ and $D/A = 1/20$. The mesh has 390×170 rectangular cells and its grid refinement near the cylinder body is based on the recommendations of Sohankar et al. [54], in particular $h/D = 3.9 \times 10^{-3}$ near the body. The time step is $\Delta t = 5 \times 10^{-3} D/U_\infty$. The computations have been performed on a *Pentium IV* desktop with a 3 GHz processor and 1 GB of RAM, equipped with the *WINDOWS XP* operating system.

The LS-STAG code is written in *FORTRAN 90/95* and compiled with doubled precision digits with the *INTEL FORTRAN 9.1*. A relevant parameter of the computation is the residual tolerance for the linear systems, which is set at $\epsilon = 10^{-8}$. After the initial transients the linear systems of both the prediction and projection steps are solved in only one iteration.

The commercial code *FLUENT* is an unstructured finite-volume code which uses a collocated arrangement of the flow variables. The *6.0* version in 2D with single precision digits has been employed for the computation. We have used the recommended settings for unsteady laminar flows: second-order segregated *PISO* time advancement, *QUICK* scheme for convective discretization, and the *PRESTO!* method for pressure interpolation at cell faces. All the corresponding computational parameters are set at their default values, except the residual tolerances which are set at $\epsilon = 10^{-4}$ only. For this value, the iterative algorithm takes only one iteration per time step to converge in average, but we could not get a converged solution for a tolerance as low as the one used in the LS-STAG computation.

Table 11
Results for the square cylinder flow at $Re = 100$ and comparison with established results from the literature. The CPU time is measured for the time period $tD/U_\infty \in [0, 50]$.

	St	$\overline{C}_D$	C_L^{rms}	CPU time (min)
FLUENT, $\epsilon = 10^{-4}$	$0.145 \pm 2.5 \times 10^{-3}$	1.510	0.185	199
LS-STAG, $\epsilon = 10^{-6}$	$0.150 \pm 2.5 \times 10^{-3}$	1.497	0.196	48
Okajima [45]	0.141–0.145	–	–	–
Davis et al. [10]	0.154	1.55	–	–
Sohankar et al. [54]	0.144	1.44	0.141	–
Saha et al. [51]	0.159	1.51	–	–

Salient results of the simulations are presented in Table 11. Both codes give accurate results which show good agreement with the literature. On average, the *FLUENT* simulation required 1.19 CPU seconds per time step while the LS-STAG simulation took only 0.28 CPU seconds, which makes the latter more than four times faster. Even greater savings can be expected for 3D computations (note that in [38], the Cartesian code *MGLET* is 10 times faster than the curvilinear code *DeFT* for turbulent channel flow computations).

5. Concluding remarks and implementation issues

In this paper, we have developed and analyzed a new IB/cut-cell method for incompressible viscous flows. For building the LS-STAG discretization in 2D, the methodology we have followed can be roughly summarized as: enumerating the various types of cut-cells, and then discretizing the fluxes in each cut-cell such that the global conservation properties of the Navier–Stokes equations are satisfied at the discrete level.

If we adopt the terminology used for computational methods for multiphase flows (see e.g. Tezduyar [57]), the LS-STAG method can be viewed as an interface-capturing method for representing the immersed boundary, as opposed to most of the IB methods that uses markers for the immersed boundary and thus belongs to the class of interface-tracking methods. Although each class of methods has its own merits and shortcomings (e.g. for interface-capturing methods, the interface is represented within the resolution of the finite-volume mesh, while for interface-tracking methods a second mesh is needed for "tracking" the interface), the use of the level-function in the LS-STAG method provided an invaluable tool for handling the immersed boundary and computing the geometry parameters of the cut-cells.

As for any interface-tracking or -capturing method, a satisfying implementation of the method in 3D (in terms of computational efficiency, stability and accuracy) is an important issue. To fully evaluate the complexity that the method would take in 3D, one has to understand first the actual algorithmic implantation of the LS-STAG method in 2D, and in particular the construction of the discrete system for the u-momentum equation. Only the "cheap" discretization of the shear stresses will be discussed, since our computations with the "cheap" method showed an accuracy comparable to the "complete" discretization.

The discrete system for the u-momentum equation is typically built in a loop over the control volumes for $u_{i,j}$, for which we determine first the type of the east and west half cut-cells. Depending on their types, the geometry parameters of the cut-cells (cell volume $V_{i,j}$, cell-face fraction ratios $\theta^u_{i,j}$, velocity boundary data $\boldsymbol{v}^{ib}$, integration areas $\Delta x^{ib,w}_{i,j}$ and $\Delta x^{ib,e}_{i,j}$) can then be computed. In the framework of a Fortran code, these computations are efficiently performed inside a case environment. We are then able to compute the various fluxes of the momentum equation: the easiest are the normal stresses and the pressure gradient since a unique formula (Eqs. (50) and (30), respectively) is valid for all cut- and fluid cells. The shear stresses are constructed with a boolean test on the cell-face ratios (see Eqs. (54) or (56) depending on the result of the test). In contrast the convective fluxes of Appendix A have to be constructed inside the case environment, since they are built independently for each type of half cut-cells such as any of their combination gives a skew-symmetric discretization.

For building the case environment, we had to enumerate the admissible types of cut-cells by counting the values that takes the level-set function $\phi_{i,j}$ at the corners of a 2D cell: $2^4 = 16$, from which we have to subtract 2 non-admissible types that correspond to checkerboard oscillations of $\phi_{i,j}$. Out of these 14 types of admissible cut-cells, we have sorted out six basic types of cut-cells where distinct skew-symmetric discretizations of the convective fluxes had to be constructed: these six basics types are the ones shown in Fig. 3. In addition, we remind that we have shown in Fig. 6 the three basic types where distinct integration areas for the shear stress have been defined.

In 3D, the formulas for the mass matrix, pressure and normal stresses are direct extensions of the 2D case, and only the discretization of the convective and shear stress fluxes has to be constructed for each type of cut-cells. We observe that there is a maximum of 2^8 types of cut-cells, out of which 108 are admissibles. For the convective fluxes, we have sorted out 16 distinct cases of cut-cells (instead of 6 in 2D) where the skew-symmetric discretization has to be constructed (see Fig. 19). For building the diffusive fluxes discretization, 7 distinct cases (instead of 3 in 2D) have to be examined. As a conclusion, the number of different cases of cut-cells to be considered is higher than in 2D, but not *substantially* higher, and the

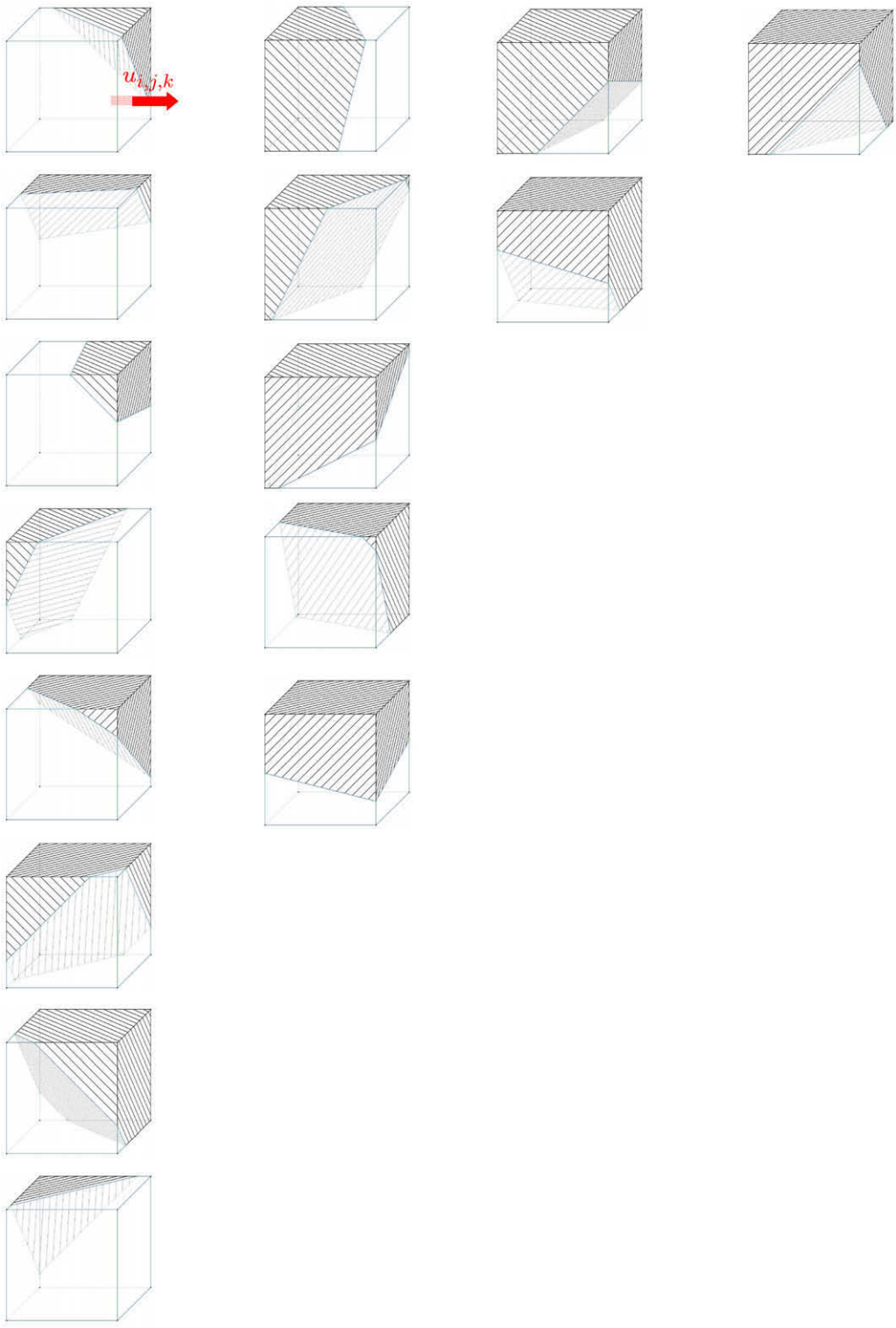


Fig. 19. Basic types of cut-cells for the skew-symmetric discretization of the convective terms for $u_{i,j,k}$. In the first row, we have grouped the cut-cells having 6 velocity unknowns (the maximum possible in 3D), then 5 unknowns, 4 unknowns, and finally 3 unknowns (the minimum).

methodology we have followed in this paper for building the LS-STAG method remains valid in 3D. The results will be reported in a future paper.

Appendix A. Complements on the discretization of the convective terms

In this appendix, we give the discretization of the convective fluxes $\int_{\Gamma_{i,j}^{u}}(\boldsymbol{v}\cdot\boldsymbol{n})u\,\mathrm{d}S$ in the half control volumes of Fig. 3 such that the skew-symmetry condition (27) is verified for any combinations of these half control volumes.

In the case of the northwest pentagonal cell (a) of Fig. 3, the boundary of the half control volume for $u_{i,j}$ is decomposed as $\Gamma_{i,j}^{u,\mathrm{w}}\cup\Gamma_{i,j}^{\mathrm{s,e}}\cup\Gamma_{i,j}^{\mathrm{n,e}}\cup\Gamma_{i,j}^{\mathrm{ib,e}}$, and on each of these faces the skew-symmetric discretization is:

$$\int_{\Gamma_{i,j}^{u,\mathrm{w}}}(\boldsymbol{v}\cdot\boldsymbol{n})u\,\mathrm{d}S=-\frac{\bar{u}_{i-1,j}+\bar{u}_{i,j}}{2}u_{\mathrm{w}},\tag{A.1a}$$

$$\int_{\Gamma_{i,j}^{\mathrm{s,e}}}(\boldsymbol{v}\cdot\boldsymbol{n})u\,\mathrm{d}S=-\frac{1}{2}\bar{v}_{i,j-1}u_{\mathrm{s}},\tag{A.1b}$$

$$\int_{\Gamma_{i,j}^{\mathrm{n,e}}}(\boldsymbol{v}\cdot\boldsymbol{n})u\,\mathrm{d}S=\frac{1}{2}\bar{v}_{i,j+1}u_{\mathrm{n}},\tag{A.1c}$$

$$\int_{\Gamma_{i,j}^{\mathrm{ib,e}}}(\boldsymbol{v}\cdot\boldsymbol{n})u\,\mathrm{d}S=\frac{1}{2}\overline{U}_{i,j}^{\mathrm{ib}}\left(\frac{1}{2}u_{i,j}+\frac{1}{2}u\left(x_{i,j}^{\mathrm{ib}},y_j\right)\right),\tag{A.1d}$$

with the following central interpolations:

$$u_{\mathrm{w}}=\frac{u_{i-1,j}+u_{i,j}}{2},\quad u_{\mathrm{s}}=\frac{u_{i,j-1}+u_{i,j}}{2},\quad u_{\mathrm{n}}=\frac{u_{i,j}+u_{i,j+1}}{2}.\tag{A.2}$$

For the case of the west trapezoidal cell (b), the boundary is $\Gamma_{i,j}^{u,\mathrm{w}}\cup\Gamma_{i,j}^{\mathrm{s,e}}\cup\Gamma_{i,j}^{\mathrm{n,e}}$, where the discretization on the last two faces is given by (A.1b) and (A.1c), respectively, and:

$$\int_{\Gamma_{i,j}^{u,\mathrm{w}}}(\boldsymbol{v}\cdot\boldsymbol{n})u\,\mathrm{d}S=\frac{1}{2}\left(\overline{U}_{i,j}^{\mathrm{ib}}-\bar{u}_{i,j}\right)\left(\frac{1}{2}u_{i,j}+\frac{1}{2}u_{i,j}^{\mathrm{ib}}\right).\tag{A.3}$$

For the case of northeast pentagonal cell (c), the boundary is $\Gamma_{i,j}^{u,\mathrm{w}}\cup\Gamma_{i,j}^{\mathrm{s,e}}\cup\Gamma_{i,j}^{\mathrm{ib,e}}$, where the discretization on the first two faces is given by (A.1a) and (A.1b), respectively, and:

$$\int_{\Gamma_{i,j}^{\mathrm{ib,e}}}(\boldsymbol{v}\cdot\boldsymbol{n})u\,\mathrm{d}S=\frac{1}{2}\left(\overline{U}_{i,j}^{\mathrm{ib}}+\bar{v}_{i,j}\right)\left(\frac{1}{2}u_{i,j}+\frac{1}{2}u\left(x_i,y_{i,j}^{\mathrm{ib}}\right)\right).\tag{A.4}$$

For the case of north trapezoidal cell (d), the boundary is $\Gamma_{i,j}^{u,\mathrm{w}}\cup\Gamma_{i,j}^{\mathrm{s,e}}\cup\Gamma_{i,j}^{\mathrm{ib,e}}$, where the discretization on the first two faces is given by (A.1a) and (A.1b), respectively, and:

$$\int_{\Gamma_{i,j}^{\mathrm{ib,e}}}(\boldsymbol{v}\cdot\boldsymbol{n})u\,\mathrm{d}S=\frac{1}{2}\overline{U}_{i,j}^{\mathrm{ib}}\left(\frac{1}{2}u_{i,j}+\frac{1}{2}u\left(x_i,y_{i,j}^{\mathrm{ib}}\right)\right).\tag{A.5}$$

For the case of northwest triangular cell (e) and northwest salient corner (f), the boundary is $\Gamma_{i,j}^{u,\mathrm{w}}\cup\Gamma_{i,j}^{\mathrm{s,e}}\cup\Gamma_{i,j}^{\mathrm{ib,e}}$, where the discretization on face $\Gamma_{i,j}^{\mathrm{s,e}}$ is given by (A.1b), and on the other two faces one has:

$$\int_{\Gamma_{i,j}^{u,\mathrm{w}}}(\boldsymbol{v}\cdot\boldsymbol{n})u\,\mathrm{d}S=\frac{1}{2}\left(u_{i,j}^{\mathrm{ib}}[n_x\Delta S]_{i,j}^{\mathrm{ib}}-\bar{u}_{i,j}\right)\left(\frac{1}{2}u_{i,j}+\frac{1}{2}u_{i,j}^{\mathrm{ib}}\right),\tag{A.6a}$$

$$\int_{\Gamma_{i,j}^{\mathrm{ib,e}}}(\boldsymbol{v}\cdot\boldsymbol{n})u\,\mathrm{d}S=\frac{1}{2}v_{i,j}^{\mathrm{ib}}[n_y\Delta S]_{i,j}^{\mathrm{ib}}\left(\frac{1}{2}u_{i,j}+\frac{1}{2}u\left(x_i,y_{i,j}^{\mathrm{ib}}\right)\right).\tag{A.6b}$$

Appendix B. Complements on the shear stress force acting on the immersed boundary

In this appendix, we describe for each cut-cell of Fig. 6 the quadrature of the shear stress used for discretizing the hydrodynamic force (61) acting on the immersed boundary. The quadratures used on the cut-cells of Fig. 3 will be easily deduced from the expressions given below.

The quadratures used for the drag (61a) and lift (61b) forces acting on the northeast pentagonal cell (a) of Fig. 6 are, respectively:

$$\mathrm{Quad}_{i,j}^{\mathrm{ib}}\left(\frac{\partial u}{\partial y}\boldsymbol{e}_y\cdot\boldsymbol{n}\right)=[n_y\Delta S]_{i,j}^{\mathrm{ib}}\left.\frac{\partial u}{\partial y}\right|_{i,j},\tag{B.1a}$$

$$\mathrm{Quad}_{i,j}^{\mathrm{ib}}\left(\frac{\partial v}{\partial x}\boldsymbol{e}_x\cdot\boldsymbol{n}\right)=[n_x\Delta S]_{i,j}^{\mathrm{ib}}\left.\frac{\partial v}{\partial x}\right|_{i,j},\tag{B.1b}$$

with $\partial u/\partial y|_{i,j}$ given by the one-sided finite-difference formula (56) and analogously:

$$\left.\frac{\partial v}{\partial x}\right|_{i,j} = \frac{v\left(x^{\mathrm{ib}}_{i,j}, y_j\right) - v_{i,j}}{\frac{1}{2}\theta^{v}_{i,j}\Delta x_i}. \tag{B.2}$$

The projected areas $[n_x \Delta S]^{\mathrm{ib}}_{i,j}$ and $[n_y \Delta S]^{\mathrm{ib}}_{i,j}$ are defined by (17), and for cut-cell (a) their expression reduces to:

$$[n_x \Delta S]^{\mathrm{ib}}_{i,j} = \left(1 - \theta^{u}_{i,j}\right)\Delta y_j, \quad [n_x \Delta S]^{\mathrm{ib}}_{i,j} = \left(1 - \theta^{v}_{i,j}\right)\Delta x_i.$$

For the north trapezoidal cell (b), it is more natural to use a trapezoidal quadrature:

$$\mathrm{Quad}^{\mathrm{ib}}_{i,j}\left(\frac{\partial u}{\partial y}\boldsymbol{e}_y \cdot \boldsymbol{n}\right) = \Delta x_i \left(\frac{1}{2}\left.\frac{\partial u}{\partial y}\right|_{i-1,j} + \frac{1}{2}\left.\frac{\partial u}{\partial y}\right|_{i,j}\right), \tag{B.3a}$$

$$\mathrm{Quad}^{\mathrm{ib}}_{i,j}\left(\frac{\partial v}{\partial x}\boldsymbol{e}_x \cdot \boldsymbol{n}\right) = \left(\theta^{u}_{i-1,j} - \theta^{u}_{i,j}\right)\Delta y_j \left(\frac{1}{2}\left.\frac{\partial v}{\partial x}\right|_{i-1,j} + \frac{1}{2}\left.\frac{\partial v}{\partial x}\right|_{i,j}\right). \tag{B.3b}$$

For the case of northwest triangular cell (c), one has:

$$\mathrm{Quad}^{\mathrm{ib}}_{i,j}\left(\frac{\partial u}{\partial y}\boldsymbol{e}_y \cdot \boldsymbol{n}\right) = \theta^{v}_{i,j-1}\Delta x_i \left(\frac{1}{2}\left.\frac{\partial u}{\partial y}\right|_{i-1,j-1} + \frac{1}{2}\left.\frac{\partial u}{\partial y}\right|_{i,j}\right), \tag{B.4a}$$

$$\mathrm{Quad}^{\mathrm{ib}}_{i,j}\left(\frac{\partial v}{\partial x}\boldsymbol{e}_x \cdot \boldsymbol{n}\right) = -\theta^{u}_{i,j}\Delta y_j \left(\frac{1}{2}\left.\frac{\partial v}{\partial x}\right|_{i-1,j-1} + \frac{1}{2}\left.\frac{\partial v}{\partial x}\right|_{i,j}\right). \tag{B.4b}$$

References

[1] A. Arakawa, Computational design for long-term numerical integration of the equations of fluid motion: two-dimensional incompressible flow, part I, J. Comput. Phys. 1 (1966) 119–143.
[2] U.M. Ascher, L.R. Petzold, Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations, SIAM, Philadelphia, 1998.
[3] E. Balaras, Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations, Comput. Fluid 33 (2004) 375–404.
[4] T.J. Barth, D.C. Jespersen, The design and application of upwind schemes on unstructured meshes, AIAA Paper, 89-0366, 1989.
[5] M. Bergmann, L. Cordier, J.P. Brancher, Drag minimization of the cylinder wake by trust-region proper orthogonal decomposition, in: R. King (Ed.), Notes on Numerical Fluid Mechanics and Multidisciplinary Design, 2007, pp. 309–324.
[6] O. Botella, On a collocation B-spline method for the solution of the Navier–Stokes equations, Comput. Fluid 31 (2002) 397–420.
[7] O. Botella, Y. Cheny, On the treatment of complex geometries in a Cartesian grid flow solver with the level set method, in: European Conference on Computational Fluid Dynamics ECCCOMAS CFD 06, 2006.
[8] Y. Cheny, O. Botella, La méthode LS-STAG: une nouvelle approche de type frontière immergée/"Level-Set" pour le calcul d'écoulements visqueux incompressibles en géométries complexes, Grenoble, 27-31 Aovt 2007, 18Fme CongrFs Frantais de MTcanique.
[9] M.-H. Chung, Cartesian cut cell approach for simulating incompressible flows with rigid bodies of arbitrary shape, Comput. Fluid 35 (2006) 607–623.
[10] R.W. Davis, E.F. Moore, L.P. Purtell, A numerical-experimental study of confined flow around rectangular cylinders, Phys. Fluid 27 (1) (1984) 46–59.
[11] J. Donea, A. Huerta, J.-Ph. Ponthot, A. Rodrìguez-Ferran, Arbitrary Lagrangian–Eulerian methods, in: E. Stein, R. de Borst, T.J.R. Hughes (Eds.), Encyclopedia of Computational Mechanics. Fundamentals, vol. 1, 2004.
[12] M. Dröge, R. Verstappen, A new symmetry-preserving Cartesian-grid method for computing flow past arbitrarily shaped objects, Int. J. Numer. Method Fluid 47 (2005) 979–985.
[13] R. Eymard, T. Gallouet, R. Herbin, Finite volume methods, in: P. Ciarlet, J.L. Lions (Eds.), Handbook for Numerical Analysis, 2000, pp. 715–1022.
[14] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, J. Comput. Phys. 161 (2000) 35–60.
[15] C. Farhat, P. Geuzaine, Design and analysis of robust ALE time-integrators for the solution of unsteady flow problems on moving grids, Comput. Method Appl. Mech. Eng. 193 (39–41) (2004) 4073–4095.
[16] C. Farhat, P. Geuzaine, C. Grandmont, The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids, J. Comput. Phys. 174 (2) (2001) 669–694.
[17] J.H. Ferziger, M. Perìc, Computational Methods for Fluid Dynamics, Springer, Berlin, 1996.
[18] F. Gibou, R.P. Fedkiw, Li-Tien Cheng, M. Kang, A second-order-accurate symmetric discretization of the Poisson equation on irregular domains, J. Comput. Phys. 176 (2002) 205–227.
[19] A. Gilmanov, F. Sotiropoulos, A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies, J. Comput. Phys. 207 (2) (2005) 457–492.
[20] W. Gu, C. Chyu, D. Rockwell, Timing of vortex formation from an oscillating cylinder, Phys. Fluid 6 (11) (1994) 3677–3682.
[21] H. Guillard, C. Farhat, On the significance of the geometric conservation law for flow computations on moving meshes, Comput. Method Appl. Mech. Eng. 190 (11–12) (2000) 1467–1482.
[22] E. Guilmineau, P. Queutey, A numerical simulation of vortex shedding from an oscillating circular cylinder, J. Fluid Struct. 16 (2002) 773–794.
[23] E. Guyon, J.P. Hulin, L. Petit, Physical Hydrodynamics, Oxford University Press, Oxford, 2001.
[24] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surfaces, Phys. Fluid 8 (1965) 2181–2189.
[25] J.C. Hart, Sphere tracing: a geometric method for the antialiased ray tracing of implicit surfaces, Visual Comput. 12 (1996) 527–545.
[26] J.W. He, R. Glowinski, R. Metcalfe, A. Nordlander, J. PTriaux, Active control and drag optimization for flow past a circular cylinder. Part 1: Oscillatory cylinder rotation, J. Comput. Phys. 163 (2000) 87–117.
[27] R.D. Henderson, Nonlinear dynamics and pattern formation in turbulent wake transition, J. Fluid Mech. 352 (1997) 65–112.
[28] G. Iaccarino, R. Verzicco, Immersed boundary technique for turbulent flow simulations, Appl. Mech. Rev. 56 (2003) 331–347.
[29] A. Jameson, The construction of discretely conservative finite volume schemes that also globally conserve energy or entropy, J. Sci. Comput. 34 (2008) 152–187.
[30] S. Kang, G. Iaccarino, P. Moin, Accurate and efficient immersed-boundary interpolations for viscous flows, in: Center for Turbulence Research Briefs, NASA Ames/Stanford University, 2004, pp. 31–43.
[31] D. Kim, H. Choi, Immersed boundary method for flow around an arbitrarily moving body, J. Comput. Phys. 212 (2) (2006) 662–680.

[32] M.P. Kirkpatrick, S.W. Armfield, J.H. Kent, A representation of curved boundaries for the solution of the Navier–Stokes equations on a staggered three-dimensional Cartesian grid, J. Comput. Phys. 184 (1) (2003) 1–36.
[33] M. Lesoinne, C. Farhat, Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations, Comput. Method Appl. Mech. Eng. 134 (1–2) (1996) 71–90.
[34] D.K. Lilly, Introduction to "Computational design for long-term numerical integration of the equations of fluid motion: two-dimensional incompressible flow, part I", J. Comput. Phys. 135 (1966) 101–102.
[35] M.N. Linnick, H.F. Fasel, A high order immersed interface method for simulating unsteady incompressible flows on irregular domains, J. Comput. Phys. 204 (2005) 157–192.
[36] R. Löhner, S. Appanaboyina, J.R. Cebral, Comparison of body-fitted, embedded and immersed solutions of low Reynolds-number 3-D incompressible flows, Int. J. Numer. Method Fluid 57 (1) (2008) 13–30.
[37] X. Lu, C. Dalton, Calculation of the timing of vortex formation from an oscillating cylinder, J. Fluid Struct. 10 (5) (1996) 527–541.
[38] M. Manhart, G.B. Deng, T.J. Hüttl, F. Tremblay, A. Segal, R. Friedrich, J. Piquet, P. Wesseling, The minimal turbulent flow unit as test case for three different computer codes, in: E.H. Hrischel (Ed.), Numerical Flow Simulation I, Vieweg, Braunschweig, 1998, pp. 365–381.
[39] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239–261.
[40] R. Mittal, H. Dong, M. Bozkurttas, F.M. Najjar, A. Vargas, A.v. Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, J. Comput. Phys. 227 (2008) 4825–4852.
[41] J. Mohd-Yusof, Combined immersed boundaries/B-splines methods for simulations of flows in complex geometries, in: Center for Turbulence Research Briefs, NASA Ames/Stanford University, 1997.
[42] Y. Morinishi, T.S. Lund, O.V. Vasilyev, P. Moin, Fully conservative higher order finite difference schemes for incompressible flow, J. Comput. Phys. 143 (1998) 90–124.
[43] F. Muldoon, S. Acharya, A divergence-free interpolation scheme for the immersed boundary method, Int. J. Numer. Method Fluid 56 (2008) 1845–1884.
[44] C. Norberg, Fluctuating lift on a circular cylinder: review and new measurements, J. Fluid Struct. 17 (2003) 57–96.
[45] Atsushi Okajima, Strouhal numbers of rectangular cylinders, J. Fluid Mech. 123 (1982) 379–398.
[46] S. Osher, R.P. Fedkiw, Level set methods: an overview and some recent results, J. Comput. Phys. 169 (2001) 463–502.
[47] S. Osher, R.P. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, New-York, 2003.
[48] S. Osher, J.A. Sethian, Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[49] N. Peller, A. Le Duc, F. Tremblay, M. Manhart, High-order stable interpolations for immersed boundary methods, Int. J. Numer. Method Fluid 52 (2006) 1175–1193.
[50] O. Pironneau, Finite Element Method for Fluids, Wiley, 1989.
[51] A.K. Saha, K. Muralidhar, G. Biswas, Transition and chaos in two-dimensional flow past a square cylinder, J. Eng. Mech. 126 (5) (2000) 523–532.
[52] P. Saramito, Numerical simulation of viscoelastic fluid flows using incompressible finite element method and a θ-method, Math. Model. Numer. Anal. 28 (1994) 1–35.
[53] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annu. Rev. Fluid Mech. 31 (1999) 567–603.
[54] A. Sohankar, C. Norberg, L. Davidson, Low-Reynolds-number flow around a square cylinder at incidence: study of blockage, onset of vortex shedding and outlet boundary condition, Int. J. Numer. Method Fluid 26 (1) (1998) 39–56.
[55] F. Sotiropoulos, S. Abdallah, The discrete continuity equation in primitive variable solutions of incompressible flow, J. Comput. Phys. 95 (1991) 212–227.
[56] M. Sussman, S. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146–159.
[57] T.E. Tezduyar, Finite element methods for fluid dynamics with moving boundaries and interfaces, in: E. Stein, R. de Borst, T.J.R. Hughes (Eds.), Encyclopedia of Computational Mechanics. Fluids, vol. 3, 2004.
[58] P.T. Tokumaru, P.E. Dimotakis, Rotary oscillation control of a cylinder wake, J. Fluid Mech. 1224 (1991) 77–90.
[59] P.G. Tucker, Z. Pan, A Cartesian cut cell method for incompressible viscous flow, Appl. Math. Model. 24 (2000) 591–606.
[60] H.S. Udaykumar, R. Mittal, W. Shyy, Computation of solid–liquid phase fronts in the sharp interface limit on fixed grids, J. Comput. Phys. 153 (2) (1999) 535–574.
[61] H.S. Udaykumar, R. Mittal, P. Rampunggoon, A. Khanna, A sharp interface Cartesian grid method for simulating flows with complex moving boundaries, J. Comput. Phys. 174 (2001) 345–380.
[62] J. van Kan, C. Vuik, P. Wesseling, Fast pressure calculation for 2D and 3D time dependent incompressible flow, Numer. Linear Algebra Appl. 7 (2000) 429–447.
[63] R.W.C.P. Verstappen, A.E.P. Veldman, Symmetry-preserving discretization of turbulent flow, J. Comput. Phys. 187 (2003) 343–368.
[64] M.C. Wendl, R.K. Agarwal, Mass conservation and the accuracy of non-staggered grid incompressible flow schemes, in: M.M. Hafez (Ed.), Numerical Simulations of Incompressible Flows, World Scientific, Singapore, 2003, pp. 277–287.
[65] J. Yang, E. Balaras, An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries, J. Comput. Phys. 215 (2006) 12–40.
[66] T. Ye, R. Mittal, H.S. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, J. Comput. Phys. 156 (1999) 209–240.
[67] M.M. Zdravkovich, Flow Around Circular Cylinders, Fundamentals, vol. 1, Oxford University Press, Oxford, 2003.