Agenzia nazionale per le nuove tecnologie,
l'energia e lo sviluppo economico sostenibile

Ministero dello Sviluppo Economico

RICERCA DI SISTEMA ELETTRICO

A numerical thecnique for treating complex geometries in compressible staggered cartesian code

D. Cecere, E. Giacomazzi, F.R. Picchia, N.M.S. Arcidiacono, F. Donato

Report RdS/2012/197

A NUMERICAL TECHNIQUE FOR TREATING COMPLEX GEOMETRIES IN COMPRESSIBLE STAGGERED CARTESIAN CODES

D. Cecere, E. Giacomazzi, F.R.Picchia, N.M.S. Arcidiacono, F. Donato (ENEA)

Settembre 2012

Report Ricerca di Sistema Elettrico
Accordo di Programma MSE-ENEA sulla Ricerca di Sistema Elettrico
Area: Produzione di energia elettrica e protezione dell'ambiente
Progetto: Studi sull'utilizzo pulito dei combustibili fossili, la cattura ed il sequestro della CO_2

Responsabile del Progetto: Stefano Giammartini, ENEA

Index

Abstract

In the present report, within the program agreement MSE-ENEA, an efficient Immersed Volume Method (IVM) for the computation of compressible viscous flows with complex stationary geometries in a staggered non-uniform cartesian finite difference code (HeaRT code) is presented.

A background Cartesian mesh is generated for each staggered variable and a finite volume approach is adopted in the layer near the immersed boundaries by means of cut cell method. Accurate description of the real three-dimensional geometry inside the cut cell volume is preserved by means of triangulated surface description instead of approximating it by a plane. The overall accuracy of the base solver (2nd order in this article) is preserved by means of high order flux reconstruction in the cut cells.

The Large Eddy Simulation (LES) solver is parallelized using domain decomposition and message passing interface. The robustness and accuracy of the method is proved simulating a laminar flow past a cube at Re= 215, a turbulent flow past a sphere with a sting at Re = 51500 and a turbulent premixed, stoichiometric CH4/Air bluff body flame at Re = 3200, both adopting the LES approach.

Introduction

The development and diffusion of LES as a more and more common methodology applied to a wide variety of turbulent flows has been possible due to the rapid increase in computational power. In particular, accuracy, robustness, efficiency and handling of complex three-dimensional geometries are key requirements for LES engineering applications. Numerical codes based on structured grids and finite difference scheme with a staggered formulation of the transported variables fulfill all these requirements. While complex geometries are more naturally treated with unstructured grids, structured grids let a more simplified management of data coding and thus higher computational efficiency. Finite difference schemes, both explicit ([1],[2]) and compact [3], are characterized by an aliasing error lower than other methods due to their enhanced damping at high wavenumbers ([4],[5]) and thanks to the possible modified forms of the non-linear advection terms, e.g., the skew-symmetric form [4]. The dispersive properties (and robustness) of finite difference schemes are improved with respect to collocated schemes by adopting a staggered formulation of variables ([3],[6]). Besides, staggering naturally provides a fully conservative form of equations, and in particular guarantees the conservation of total energy. Hence, within this outlined framework, an efficient, robust and accurate technique to simulate complex geometries in structured grids is required, especially for compressible and reacting turbulent flows.

Many flow physical problems involve geometrical complexities with irregular boundaries that usually are not aligned with the grid. If that is the case, the solid boundary will cut through this grid. Because the grid does not conform to the solid boundary, imposing the boundary conditions will require to modify the governing equations in the vicinity of the boundary.

Two major classes of methods are suitable for treating arbitrarily complex geometries with cartesian grids. They distinguish on the basis of their approach to impose boundary conditions in the cells cut by the solid interface. The first includes the classical Immersed Bounday (IB) methods where governing equations are modified adding forcing terms to account for the embedded complex surface ([9],[10]). These methods are attractive because of their simplicity, but their major drowbacks are the occurrence of non-divergence free velocities in incompressible flows and spurious unphysical pressure oscillations in compressible ones due to not satisfying strictly conservation of mass, momentum and kinetic energy near the irregular boundaries ([11],[12]). The second class includes methods based on the so called cut-cell method (also called Cartesian grid methods) introduced first by Clarke [13]. This approach requires truncating the Cartesian cells at the immersed boundary to create new cells which conform to the shape of the complex surface. In this way, the advantage of a Cartesian grid is retained for the standard, non-boundary cells and a more complex treatment is necessary only for the boundary cells.

The cut-cell method is based on a finite-volume discretization of the flow equations in the cells cut by the immersed boundary surface; the local mass and momentum conservation are satisfied. On a non-staggered grid, the velocity and density (or equivalently pressure, energy or scalars) are collocated at the same nodes and the geometry of the associated control volumes is also identical. With a staggered grid, the scalars' cell and the cells associated to each of the three velocity components are at different locations and have different shapes when they are cut by an embedded boundary. A cut cell scheme for a staggered grid must deal with this extra complexity in a consistent manner. Another complication of the staggered approach involves the calculation of convective fluxes, because different interpolation stencils are used for velocities and scalars.

For sharp-interface cartesian grid methods, the so called "small cell problem" causing numerical instability [14] arises when finite difference or finite volume methods are applied to small-sized irregular cut grid cells. The new volume elements created by the cutting procedure can be many times smaller than the original uncut cartesian cells. Their small volume can seriously increase the stiffness of the governing equations and can lead to problems of numerical stability. Johansen and Colella [15] adopted a flux redistribution procedure. Most notable is the cell merging technique used by Chung ([16],[17],[18]) that links small cells and adjacent fluid cells to form master-slave pair.

Very few cut cell methods for staggered grids have been reported in literature and implementations of cut-cell based Cartesian grid methods for the compressible 3D Navier-Stokes equations on staggered grid have never been presented to the best of authors' knowledge. Kirkpatrick et al. [17] proposed a method for representing curved boundaries as quadratic surfaces for the solution of the incompressible Navier-Stokes equations on non-uniform staggered, three-dimensional cartesian grid. In the same article, it was also developed a cell-linkig method to overcome problems associated with the creation of small cells without adding the complexities of cell-merging techniques. Meyer et al. [19] proposed a conservative, second-order accurate Cartesian cut-cell method for incompressible Navier-Stokes equations in three-dimensional non uniform staggered grids suitable for finite-volume discretization. To ensure numerical stability for small cells they followed the conservative mixing procedure by Hu et al. [20]. Cheny et al. [21] proposed a new IB method for incompressible viscous flows, based on the MAC method [23] for staggered not-uniform Cartesian grids where the irregular boundary is sharply represented by its level-set function and flow variables are computed in the cut cells and not interpolated. Their method, called the LS-STAG method, is based on the symmetry preserving finite-volume method by Verstappen and Veldman [24], which has the ability to preserve the conservation properties (for total mass, momentum and kinetic energy). Seo et al. [22] proposed a method for reducing spurious pressure oscillations in simulations of moving boundary problems with sharp-interface immersed boundary methods, applying a cut-cell method to the solution of the Poisson equation.

The purpose of this work is to present a new efficient, conservative, high-order accurate (up to third order) Cartesian cut-cell method, called Immersed Volume Method (IVM) for the compressible Navier-Stokes equations solved by finite difference method on three-dimensional non-uniform staggered grids. This method is suitable for the extension to solid/fluid heat conduction and to moving boundaries. The immersed boundary is represented by means of triangulated surfaces (STL representation). In literature, the arbitrary curved or irregular boundary is approximated (except for some cases in collocated grids ([25],[26]) by means of a plane in each cut cell. Unfortunately for LES, unless a very fine grid is employed, the representation of an irregular boundary (or shape) using a planar approximation is too crude, with the deleterious result that additional errors are introduced into the flow solution in the vicinity of the irregular boundary. The full geometrical characteristics of the cut cells, with the intersecting surface described by polygons with different normals, are identified in a preprocessor procedure and retained in flow solver calculations.

The IVM method solves exactly, by means of finite volume method, the flow variables in the cut cells and links the velocities and energy fluxes to the thermodynamic variable changes overcoming in this way the drawbacks of classical IB methods. The high-order fluxes calculated at the cut cells' faces are also adopted in the finite difference Navier-Stokes discretization of the Cartesian adjacent cells layer (with at least one face in contact with a cut cell) to match IVM method with the general finite difference code. In fact, since the values ââof the fluid dynamic variables are stored at the centroid of the cut cell volume of fluid, by directly applying a finite difference method to this second layer, would lead to an incorrect fluxes evaluation (different from that calculated by IVM method) and thendue to disagreement with the conservativity property to numerical instabilities.

The flow variables are stored at the cut-cell volume centroid and, to ensure numerical stability for small cells, a cell-merging/cell-linking method is adopted to form a master/slaves pair. The basic idea is to combine several neighbouring cells together and to shift the original cut cells volume centroids to that of the merged new cut cell [18, 27, 28].

The authors are currently working on an extension of the method to finite difference codes with order of accuracy higher than two (explicit or implicit compact scheme) and heat conduction inside solid boundaries. A general formulation to impose boundary conditions and calculate viscous and diffusive fluxes on the complex cutting surface is presented without introducing ghost cells.

The paper is organized as follows. In Section 2 the governing equations within the LES framework are presented. In section 3 the procedures adopted for determining the geometrical characteristics of the cut cells are prescribed. Section 4 presents the IVM discretization for the calculation of continuity, momentum and energy convective and diffusive fluxes. Section 5 is devoted to some numerical tests on non-reactive/reactive flows at low and high Reynolds numbers for assessing the accuracy and robustness of the IVM method.

Governing Equations

In LES each turbulent field variable is decomposed into a resolved and a subgrid-scale part. In this work, the spatial filtering operation is implicitly defined by the local grid cell size. Variables per unit volume are treated using Reynolds decomposition, while Favre (density weighted) decomposition is used to describe quantities per mass unit. The instantaneous small-scale fluctuations are removed by the filter, but their statistical effects remain inside the unclosed terms representing the influence of the subgrid scales on the resolved ones. In this article, a test deals with combustion to show the robustness of the suggested technique. Gaseous combustion is governed by a set of transport equations expressing the conservation of mass, momentum and energy, and by a thermodynamic equation of state describing the gas behaviour. For a mixture of N_s ideal gases in local thermodynamic equilibrium but chemical nonequilibrium, the corresponding filtered field equations (extended Navier−Stokes equations) are:

- Transport Equation of Mass

$$\frac{\partial \overline{\rho}}{\partial t} + \frac{\partial \overline{\rho}\widetilde{u}_i}{\partial x_i} = 0. \tag{1}$$

- Transport Equation of Momentum

$$\frac{\partial(\overline{\rho}\widetilde{u}_j)}{\partial t} + \frac{\partial(\overline{\rho}\widetilde{u}_i\widetilde{u}_j + \overline{p}\delta_{ij})}{\partial x_i} = \frac{\partial\widetilde{\tau}_{ij}}{\partial x_i} + \frac{\partial\tau_{ij}^{sgs}}{\partial x_i} \tag{2}$$

- Transport Equation of Total Energy (internal + mechanical, $\mathcal{E} + \mathcal{K}$)

$$\frac{\partial(\overline{\rho}\widetilde{\mathcal{U}})}{\partial t} + \frac{\partial(\overline{\rho}\widetilde{u}_i\widetilde{\mathcal{U}} + \overline{p}\widetilde{u}_i)}{\partial x_i} = -\frac{\partial(\overline{q_i} - \widetilde{u}_j\overline{\tau_{ij}} + H_i^{sgs} + q_i^{sgs})}{\partial x_i} \tag{3}$$

- Transport Equation of the N_s Species Mass Fractions

$$\frac{\partial(\overline{\rho}\widetilde{Y}_n)}{\partial t} + \frac{\partial(\overline{\rho}\widetilde{u}_j\widetilde{Y}_n)}{\partial x_i} = \frac{\partial}{\partial x_i}(\widetilde{J}_{n,i} + \widetilde{J}_{n,i}^{sgs}) + \overline{\rho}\widetilde{\dot{\omega}}_n \tag{4}$$

- Thermodynamic Equation of State

$$\overline{p} = \overline{\rho}\sum_{n=1}^{N_s}\frac{\widetilde{Y_n}}{W_n}\mathcal{R}_u\widetilde{T} \tag{5}$$

These equations must be coupled with the constitutive equations which describe the molecular transport. In the above equations, t is the time variable, ρ the density, u_j the velocities, τ_{ij} the viscous stress tensor, $\widetilde{\mathcal{U}}$ the total filtered energy per unit of mass, that is the sum of the filtered internal energy, $\widetilde{e}$, and the resolved kinetic energy, $1/2\,\widetilde{u}_i\widetilde{u}_i$, q_i is the heat flux, p the pressure, T the temperature, $\mathcal{R}_u$ is the universal gas constant, W_n the nth-species molecular weight, $\dot{\omega}_n$ is the production/destruction rate of species n, diffusing at velocity $V_{i,n}$ and resulting in a diffusive mass flux $\boldsymbol{J_n} = \rho Y_n\boldsymbol{V_n}$. The stress tensor and the heat flux are respectively:

$$\overline{\tau_{ij}} = 2\mu\,(\widetilde{S_{ij}} - \frac{1}{3}\widetilde{S_{kk}}\delta_{ij}) \tag{6}$$

$$\overline{q_i} = -k\frac{\partial\widetilde{T}}{\partial x_i} + \overline{\rho}\sum_{n=1}^{N_s}\widetilde{h_n}\widetilde{Y_n}\widetilde{V_{i,n}}. \tag{7}$$

In Eqn. 7, the first term is the heat transfer by conduction, modeled by Fourier's law, the second is the heat transport due to molecular diffusion acting in multicomponent mixtures and driven by concentration gradients. The Hirschfelder and Curtiss approximate formula for mass diffusion V_n in a multicomponent mixture is adopted, i.e.,

$$\boldsymbol{J_n} = \rho Y_n\boldsymbol{V_n} = -\rho Y_n D_n\frac{\nabla X_n}{X_n} = -\rho\frac{W_n}{W_{mix}}D_n\nabla X_n\ . \tag{8}$$

where $X_n = Y_n W_{mix}/W_n$ and the D_n is

$$D_n = \frac{1 - Y_n}{\sum_{j=1,\,j\neq n}^{N_s}\frac{X_j}{\mathcal{D}_{jn}}}\ , \tag{9}$$

$\mathcal{D}_{jn}$ being the binary diffusion coefficient. When inexact expressions for diffusion velocities are used (as when using Hirschfelder's law), and in general when differential diffusion effects are considered, the constrain $\sum_{i=1}^{N_s} \boldsymbol{J_i} = \sum_{i=1}^{N_s} \rho Y_i \boldsymbol{V_i} = 0$ is not necessarily satisfied. In this paper, to impose mass conservation, an artificial diffusion velocity $\boldsymbol{V^c}$ is subtracted from the flow velocity in the species transport equations [31]. This velocity, assuming Hirschfelder's law holds, becomes :

$$\boldsymbol{V^c} = -\sum_{n=1}^{N_s} \frac{W_n}{W_{mix}} D_n \nabla X_n \ . \tag{10}$$

In Eqn. 2, the subgrid stress tensor, τ_{ij}^{sgs}, is expressed through a Smagorinsky model:

$$\tau_{ij}^{sgs} = -\bar{\rho}\big(\widetilde{u_i u_j} - \widetilde{u}_i \widetilde{u}_j\big) \simeq 2C_R(\boldsymbol{x},t)\bar{\rho}\Delta^2 \Pi^{\frac{1}{2}} \big(\widetilde{S}_{ij} - \frac{1}{3}\bar{\rho} q^2 \delta_{ij}\big), \tag{11}$$

where $1/3\,\bar{\rho}q^2$ is the subgrid kinetic energy, $\widetilde{S}_{ij} = 1/2\ (\partial\widetilde{u}_i/\partial x_j + \partial\widetilde{u}_j/\partial x_i)$ is the filtered strain rate tensor, $\Pi^{1/2} = \sqrt{2\widetilde{S}_{ij}\widetilde{S}_{ij}}$ is its module, $\Delta = \sqrt[3]{Volume}$ is the grid filter width, and C_R is the "constant" of the subgrid stress model, here dynamically computed. The unclosed subgrid reaction rates in the Eqn. (4), are modeled using the Fractal Model FM, details of which can be found in previous works [32]. In Eqn. 3, the subgrid energy flux H^{sgs} is modelled as $\mu_t/Pr_t\frac{\partial\widetilde{H}}{x_i}$, Pr_t being the turbulent Prandtl number here assumed 0.9, while the subgrid heat transfer q_i^{sgs} as $-\mu_t/(\mu_t+\mu_l)k\,\partial\widetilde{T}/\partial x_i$.

In the transport equation of the N_s Species Mass Fraction (Eqn. 4), the subgrid mass flux $\widetilde{J}_{n,i}^{sgs}$ is modelled using a gradient assumption as $\mu_t/Sc_t\partial\widetilde{Y_n}/\partial x_i$, Sc_t being the turbulent Schmidt number, here assumed 0.7. Kinetic theory is used to calculate dynamic viscosity and thermal conductivity of individual species [33]. The mixture-average properties are estimated by means of Wilke's formula with Bird's correction for viscosity ([34],[35]), and Mathur's expression for thermal conductivity [36].

The finite difference code is second-order accurate. In the case of premixed reactive flows the convective species and energy fluxes are computed adopting a third-order modified version of the advection upstream splitting method (AUSM) to reduce spurious oscillations due to strong unresolved density gradients in the flame front. Time-integration of Navier-Stokes equations (1-4) is performed by means of the fully explicit third-order accurate TVD Runge-Kutta scheme of Shu and Osher [38].

IVM method

Because of the staggering arrangement, momenta are located half a cell width from thermodynamic variables and consequently four control volumes are defined and associated to the three momenta and scalars (density, pressure, total energy, chemical species). Rather than storing the flow variables at the original Cartesian cell center, the variables are collocated at the true cut-cell volume centroid (that not always lies inside the fluid region) and the fluxes of these variables are estimated at the area's centroids of the fluid faces bounding the cut-cell. For each of the four field variable type, the relevant geometric characteristics of the resulting cut volume of fluid polyhedron, resulting from the difference of the original structured cell and the intersecting volume of the solid, has to be derived. The mass fluid volume centroid, the fluid volume fraction, the wetted surface areas and centroids are then used to interpolate variables and to calculate the fluxes required to solve the Navier-Stokes equations in the general finite volume approach.

Cut cell geometric properties evaluation

A triangulated surface mesh is used to represent the solid boundary surface for three dimensional problems (see Fig. 1). The vertices and the positive normals (towards the fluid region) of these triangles are stored in a StereoLithography file (STL). Computational cells are divided in three types: solid cells that are inside the solid volume, fluid cells that lie completely in the fluid and cut cells that are intersected by the immersed boundary surface (see Fig. 1).

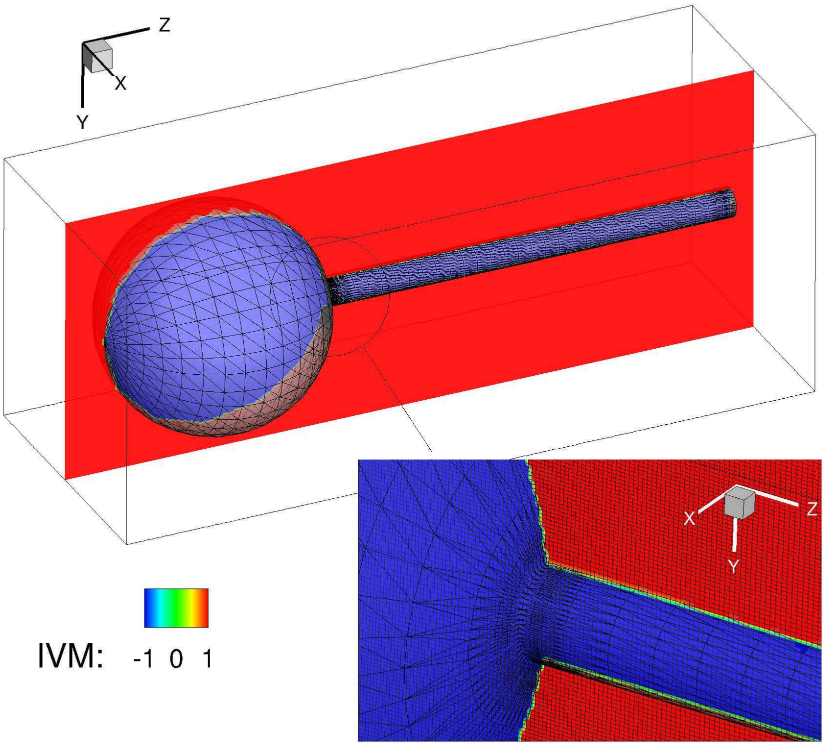

Figure 1: Top: A three-dimensional Cartesian grid illustrating the three types of cells in the cut-cell approach. The red region ($IVM = 1$) denotes the fluid cells which lie entirely outside the solid boundary, whereas the blue cells ($IVM = -1$) denote the solid cells which lie entirely inside the solid boundary. The green cells ($IVM = 0$) correspond to the cut-cells which are intersected by the internal boundary. Bottom: zoomed-in-view of the part of the immersed boundary region showing the computational cartesian grid.

In a first stage, after the production of the cartesian structured computational grid, a marker is assigned to each vertex of the cartesian cell (the grid may not be uniform) that determines whether the vertex is inside or outside the solid. A ray tracing procedure is applied in order to do this [29]. Referring to Fig. 2 a ray is traced from a point **A** and the number of intersections with the solid triangulated surface is counted. The point **A** lies inside the solid if the number of intersections is even, outside otherwise. Each computational cell's face is divided into two triangles whose vertices may be fluid or solid points (see the black bullets in Fig. 2). For

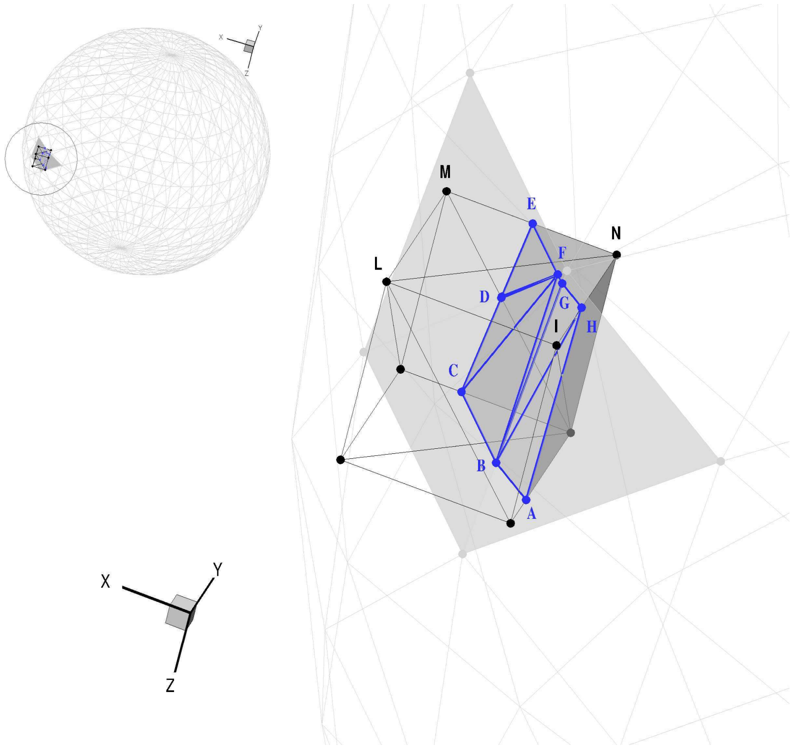

Figure 2: Example of STL boundary surface representation and cut cell. Black line: cut structured cell; Blu line: immersed boundary surface/cut cell intersection; gray line: immersed boundary surface represented by triangulation, gray solid: internal solid part of the immersed boundary.

each cartesian cell face, the intersection points of the triangulated solid boundary surface (A-B-C-D-E-F-G-H-A) with the two triangles of each face are evaluated by a fast triangle/triangle intersection routine [30] and stored in a linking list associated to the computational cell face. The intersection points are ordered to form a polyline (e.g., the connected blue points E-F-G-H in Fig. 2) that divides the face (I-L-M-N) into two polygons respectively in the fluid (E-F-G-H-I-L-M polygon of Fig. 2) and the solid region (E-F-G-H-N polygon). The wet polygon of each face (if exists) is triangularized by a two-dimensional Delaunay triangulation, where the intersection polyline of each face is adopted as a constraint. These boolean operations are performed for all the faces of the original Cartesian cells with at least one internal vertex. The faces' wet polygons form a polyhedron that is closed by the surface of the immersed solid boundary $\partial\Gamma$ internal to the computational cartesian cell. In order to characterize this surface, for each intersecting STL triangle, its intersection points with the cartesian face and its internal vertices (if they exist) are stored in a second ordered list associated to the cell (e.g. the three polygons E-F-D, D-F-G-B-C and G-H-A-B in Fig. 2 belonging to different planes). Once the volume of fluid polyhedron is identified by its set of polygons, applying Gauss's divergence theorem, the wet volume may be calculated for example as:

$$V = \int_V dV = \int_{\partial B} x\hat{\mathbf{i}} \cdot \hat{\mathbf{n}}\, dS\,, \tag{12}$$

$\hat{\mathbf{i}}$ and $\hat{\mathbf{n}}$ being the versor of the $i-th$ direction and the normal surface versor, respectively, and

∂B the polyhedron surface. Furthermore, the polyhedron fluid volume centroid coordinates x_i^V, which usually does not coincide with the volumetric center of the original square grid cell, can be calculated as

$$x_i^V = \frac{1}{V}\int_V x_i\, dV = \frac{1}{2\,V}\int_{\partial B} x_i^2 dA = \frac{1}{2\,V}\sum_{j=1}^{N_f}\int_{\partial B_j} x_i^2\, dS, \tag{13}$$

$$\tag{14}$$

where N_f is the number of polyhedron's polygonal faces. The centroid coordinates $x_{i,j}^A$ of the $j - th$ face area (“wetted“ or solid) are given by

$$x_{i,j}^A = \frac{1}{A_j}\int_{A_j} x_i\, dA = \frac{1}{\sum_{n=1}^{N_{t,j}} A_{n,j}}\sum_{n=1}^{N_{t,j}} x_{i,n}A_n \tag{15}$$

$$\tag{16}$$

$N_{t,j}$ being the number of triangles and $A_{n,j}$ the area of the $n - th$ triangle of the Delaunay triangulation.

The “wetted“ and solid polyhedron's areas must be calculated with high precision since in the case of uniform pressure p no source terms related to the pressure gradient are present in the three momentum equations and then the following equation must be satisfied in the $i - th$ coordinate direction:

$$\int_{\partial B} p\hat{\mathbf{i}}\cdot\hat{\mathbf{dA}} = p\int_{\partial B}\hat{\mathbf{i}}\cdot\hat{\mathbf{dA}} = 0 \tag{17}$$

$$\tag{18}$$

This means that, geometrically, the projection along a fixed direction of the signed wetted and solid surfaces must be zero (in this work is at least $\sim\ 10^{-17}\mathrm{m}^2$). In classical cut cell methods, where the internal cutting surface is approximated by a plane, this requirement is naturally fulfilled since the normal versor $\hat{\mathbf{n}}$ and the area of the solid boundary surface S are calculated as:

$$\hat{\mathbf{n}} = (\frac{A_x^+ - A_x^-}{|S|}, \frac{A_y^+ - A_y^-}{|S|}, \frac{A_z^+ - A_z^-}{|S|}) \tag{19}$$

$$\tag{20}$$

where $|S| = \sqrt{(A_x^+ - A_x^-)^2 + (A_y^+ - A_y^-)^2 + (A_z^+ - A_z^-)^2}$ and $A_i^{\pm}$ being the wetted areas on the positive/negative faces in the $i - th$ direction.

Figure 3 shows the volume of fluid centroid ($\mathrm{c_{vof}}$) where field variables are collocated, the six wetted areas' centroids, the internal boundary surfaces and their normals.

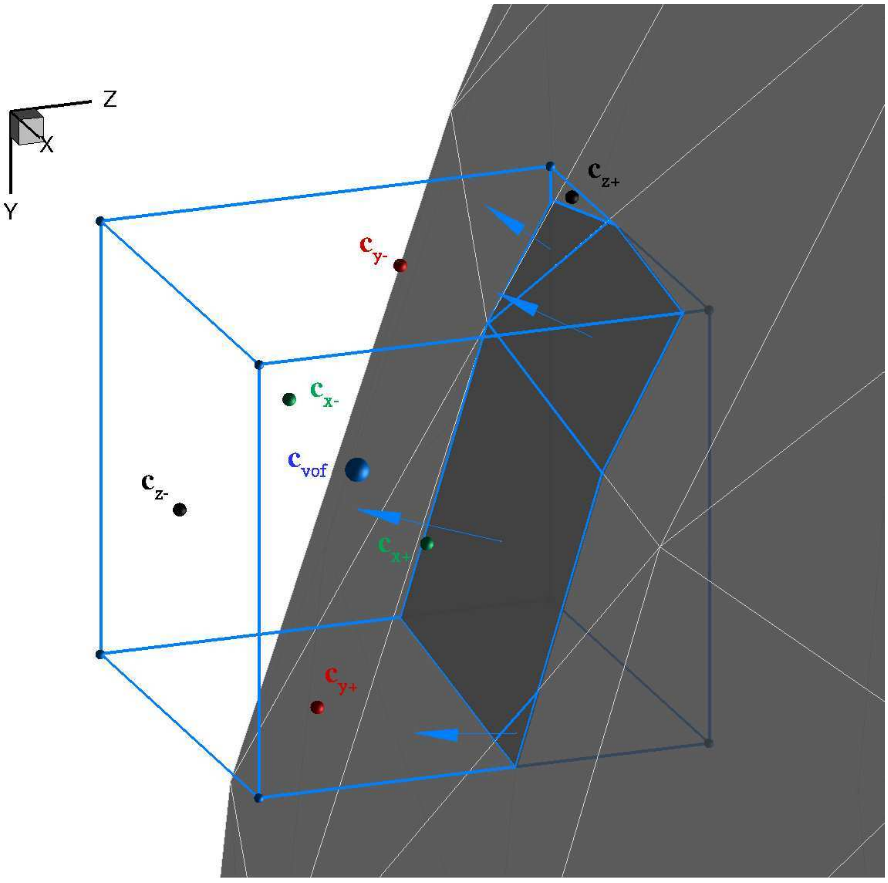

Figure 3: Example of cut cell geometric characteristics. Blue bullet: cut cell volume of fluid centroid; black bullets: z-normal faces's centroids; green bullets: x-normal faces's centroids; red bullets: y-normal faces's centroid; blue edges with bullets: original cartesian cell; grey surface with blue edges: surface intersection of the immersed body with the cartesian cell; blue arrows: the normals of the immersed surfaces.

Small cell treatment

After calculating all geometric properties of the cut cells, the problem of removing small cut cells has to be taken into account. In fact, the volume of fluid fraction V can be arbitrarily small compared with that of the original cartesian grid cell. These small cell volumes increase the stiffness of the system of equations and restrict the maximum time step that can be used in an explicit time stepping procedure. Special treatment of such cells is necessary for numerical stability and several approaches have been proposed in literature: the application of cell linking [17], cell merging [22], the adoption of a redistribution technique for the cell's volumes ([20],[19]) and mixed approaches [39]

In this work, the solution of a combined cell-merging/cell-linking approach proposed by Hartmann et al. [39] is adopted. The basic idea is to combine several neighbouring cells together in a newly combined larger cell. To implement this technique, we need first to determine which cells should be merged. A cut cell is considered a slave cell when its volume fraction is less then one half of the original Cartesian cell. Let $\mathbf{n}_s$ be the mean normal versor of the solid boundary surface, characterized by N_p polygonal surfaces (see Fig. 3):

$$\mathbf{n}_s = \frac{\sum_{k=1}^{N_p} A_k \mathbf{n}_k}{\sum_{k=1}^{N_p} A_k} \tag{21}$$

A_k being the $k-th$ surface area with normal $\mathbf{n}_k$. The master cell is chosen in $i-th$ coordinate direction maximizing the dot product $\hat{\mathbf{n}}_\mathbf{s}\cdot(\pm\hat{\mathbf{n}}_\mathbf{i})$, $i\ =\ 1,2,3\,,$ $\hat{\mathbf{n}}_\mathbf{i}$ being the versor of $i-th$ coordinate direction. When multiple master cells are identified, the greatest is chosen. In the case that the master cell m is a slave cell of another cell $m^\star$, both s and m become slave cells of $m^\star$. The cell volume $V_{m^\star}$ and the centroid $\mathbf{x}^{V_{m^\star}}$ of the combined master-slave(s) cluster $m^\star$ are computed as:

$$V_{m^\star} = V_m + \sum_{k=1}^{N_s} V_{s_k} \tag{22}$$

$$\mathbf{x}^{V_{m^\star}} = \frac{\mathbf{x}^{V_m} V_m + \sum_{k=1}^{N_s} \mathbf{x}^{V_{s_k}} V_{s_k}}{V_{m^\star}} \tag{23}$$

$$A^j_{m^\star} = \sum_{k=1}^{N_s} (A^j_{s_k} |\mathbf{n}^k_i \times \mathbf{n}^j|) \tag{24}$$

$$\mathbf{x}^{A^j_{m^\star}} = \frac{\mathbf{x}^{A^j_m} A^j_m + \sum_{k=1}^{N_s} (\mathbf{x}^{A^j_{s_k}} A^j_{s_k} |\mathbf{n}^k_i \times \mathbf{n}^j|)}{A^j_{m^\star}} \tag{25}$$

N_s being the number of slave cells associated to the master cell m, V_{s_k} and $\mathbf{x}^{V_{s_k}}$ the volume and the volume centroid of the slave cell s_k respectively, $\mathbf{n}^k_i$ the versor of the $i-th$ direction of connection between the slave cell s_k and the master cell m, $A^j_{s_k}$ the slave face area with normal in the $j-th$ direction. The data are copied to the slave cell(s) $s_k \in S$ and the master cell according to

$$\begin{aligned}
&\mathbf{x}^{V_m} \leftarrow \mathbf{x}^{V_{m+S}} \\
&\mathbf{x}^{V_{s_k}} \leftarrow \mathbf{x}^{V_{m+S}}, \forall s_k \in S \\
&\mathbf{x}^{A^j_m} \leftarrow \mathbf{x}^{A^j_{m^\star}} \\
&\partial B^m \leftarrow \partial B^{m+S} \\
&V^m \leftarrow V^{m+S} \\
&A^j_m \leftarrow A^j_{m^\star}.
\end{aligned} \tag{26}$$

In this way, all small cells, linked and merged with a suitable master cell m, is treated in the numerical method as passive cells contributing to the balance equations only through fluxes exchange across its surface.

IVM method discretization

In this section, first we describe how performing high order least-squares reconstruction in each cut cell, next we discuss boundary conditions enforcement and advective and diffusive flux calculation.

Least-Squares interface reconstruction

While fluxes evaluation is straightforward in structured grids, it becomes more difficult at cells cutted by solid boundaries. In the proposed formulation, the least-squares method is adopted to obtain a discretization scheme which is flexible in terms of the local cut cell volume topology and the shape of embedded boundaries and conserves mean value in the control volumes. The cell interface values of the solution variables are found using Taylor series expansion about the cell centroid c_i of the volume i:

$$\begin{aligned}\phi_{c_i}^{INT}(\mathbf{x}) = \phi_{c_i} + \left.\frac{\partial\phi}{\partial x}\right|_{c_i}\Delta x + \left.\frac{\partial\phi}{\partial y}\right|_{c_i}\Delta y + \left.\frac{\partial\phi}{\partial z}\right|_{c_i}\Delta z + \left.\frac{\partial^2\phi}{\partial x^2}\right|_{c_i}\Delta x^2 + \left.\frac{\partial^2\phi}{\partial y^2}\right|_{c_i}\Delta y^2 + \\ + \left.\frac{\partial^2\phi}{\partial z^2}\right|_{c_i}\Delta z^2 + \left.\frac{\partial^2\phi}{\partial x\partial y}\right|_{c_i}\Delta x\Delta y + \left.\frac{\partial^2\phi}{\partial y\partial z}\right|_{c_i}\Delta y\Delta z + \left.\frac{\partial^2\phi}{\partial x\partial z}\right|_{c_i}\Delta x\Delta z\end{aligned} \tag{27}$$

with $\Delta x = x - x_{c_i}, \Delta y = y - y_{c_i}, \Delta z = z - z_{c_i}$ being the distances, along the three cartesian coordinates, between the reconstruction point and the centroid c_i where derivatives in Eqn. (27) are calculated. Obviously, if a first order reconstruction is required, only the first four terms must be retained in Eqn. (27). The computational stencil of the least squares system, is constructed looking at the neighbouring control volumes ensemble $\{S_j\}_i$ $(j = 1, ..., N_i)$ of the cut/cartesian cells (the small cut cells are excluded). The control volume ensemble $\{S_j\}_i$ must include a sufficient number of control volumes for the determination of the derivatives in Eqn. (27). The minimum number of unknowns for the linear and quadratic reconstruction in 3D are 4 and 10 leading to a 2nd and 3rd order accuracy respectively. In practice, in this work, a maximum number of 20 points are used to construct the stencil of the least square method, imposing the conservation of mean in the control volumes and boundary conditions. The conservation of the mean value within the control volume V_i of the interpolating function $\phi_{c_i}^{INT}(\mathbf{x})$ requires that the following equation must be satisfied:

$$\overline{\phi}_i = \frac{1}{V_i}\int_{V_i}\phi_{c_i}^{INT}(\mathbf{x})\,dV. \tag{28}$$

Substituting the Taylor series, Eqn. (27) in Eqn. (28), collocating the mean value at the volume centroid, gives [40]

$$\begin{aligned}0 = \left.\frac{\partial\phi}{\partial x}\right|_{c_i}\overline{x} + \left.\frac{\partial\phi}{\partial y}\right|_{c_i}\overline{y} + \left.\frac{\partial\phi}{\partial z}\right|_{c_i}\overline{z} + \left.\frac{\partial^2\phi}{\partial x^2}\right|_{c_i}\frac{\overline{x}^2}{2} + \left.\frac{\partial^2\phi}{\partial y^2}\right|_{c_i}\frac{\overline{y}^2}{2} + \\ \left.\frac{\partial^2\phi}{\partial z^2}\right|_{c_i}\frac{\overline{z}^2}{2} + \left.\frac{\partial^2\phi}{\partial x\partial y}\right|_{c_i}\overline{xy} + \left.\frac{\partial^2\phi}{\partial y\partial z}\right|_{c_i}\overline{yz} + \left.\frac{\partial^2\phi}{\partial x\partial z}\right|_{c_i}\overline{xz}\,.\end{aligned} \tag{29}$$

with

$$\overline{x^m y^n z^p}_i = \frac{1}{V_i}\int_{V_i} (x - x_{c_i})^m (y - y_{c_i})^n (z - z_{c_i})^p \, dV. \tag{30}$$

that considering the expression of the centroid in Eqn. (13) reduces to calculate moments of the volume V_i respect to c_i (using Gauss' theorem). If a first order reconstruction is adopted, and the mean value is collocated at the volume of fluid centroid, the conservation of the mean condition is automatically satisfied [18].

Computing the mean value of the reconstruction $\phi_{c_i}^{INT}(\mathbf{x})$ in a volume V_j of the ensemble $\{S_j\}_i$, that forms the compact stencil of the least square method, implies that:

$$\begin{aligned}
\overline{\phi}_j &= \frac{1}{V_j}\int_{V_j} \phi_{c_i}^{INT}(\mathbf{x})\, dV = \overline{\phi}_i + \left.\frac{\partial \phi}{\partial x}\right|_{c_i} \left(\frac{1}{V_j}\int_{V_j} (x - x_{c_i})\, dV \right) + \\
&+ \left.\frac{\partial \phi}{\partial y}\right|_{c_i} \left(\frac{1}{V_j}\int_{V_j} (y - y_{c_i})\, dV \right) + \left.\frac{\partial \phi}{\partial z}\right|_{c_i} \left(\frac{1}{V_j}\int_{V_j} (z - z_{c_i})\, dV \right) \\
&+ \left.\frac{\partial^2 \phi}{\partial x^2}\right|_{c_i} \left(\frac{1}{V_j}\int_{V_j} (x - x_{c_i})^2\, dV \right) + \left.\frac{\partial^2 \phi}{\partial y^2}\right|_{c_i} \left(\frac{1}{V_j}\int_{V_j} (y - y_{c_i})^2\, dV \right) \\
&+ \left.\frac{\partial^2 \phi}{\partial z^2}\right|_{c_i} \left(\frac{1}{V_j}\int_{V_j} (z - z_{c_i})^2\, dV \right) + \left.\frac{\partial^2 \phi}{\partial x \partial y}\right|_{c_i} \left(\frac{1}{V_j}\int_{V_j} (x - x_{c_i})(y - y_{c_i})\, dV \right) \\
&+ \left.\frac{\partial^2 \phi}{\partial y \partial z}\right|_{c_i} \left(\frac{1}{V_j}\int_{V_j} (y - y_{c_i})(z - z_{c_i})\, dV \right) + \left.\frac{\partial^2 \phi}{\partial x \partial z}\right|_{c_i} \left(\frac{1}{V_j}\int_{V_j} (x - x_{c_i})(z - z_{c_i})\, dV \right)
\end{aligned} \tag{31}$$

Following the work of Gooch [40] in order to avoid the calculation of moments of each control volume V_j with respect to c_i, in Eqn. (31) $x - x_{c_i}$,$y - y_{c_i}$,$z - z_{c_i}$ are replaced with $(x - x_{c_j}) + (x_{c_j} - x_{c_i})$, $(y - y_{c_j}) + (y_{c_j} - y_{c_i})$, $(z - z_{c_j}) + (z_{c_j} - z_{c_i})$ respectively:

$$\begin{aligned}
\overline{\phi}_j = \overline{\phi}_i + \left.\frac{\partial \phi}{\partial x}\right|_{c_i} \hat{x} + \left.\frac{\partial \phi}{\partial y}\right|_{c_i} \hat{y} + \left.\frac{\partial \phi}{\partial z}\right|_{c_i} \hat{z} + \left.\frac{\partial^2 \phi}{\partial x^2}\right|_{c_i} \hat{x^2} + \left.\frac{\partial^2 \phi}{\partial y^2}\right|_{c_i} \hat{y^2} \\
+ \left.\frac{\partial^2 \phi}{\partial z^2}\right|_{c_i} \hat{z^2} + \left.\frac{\partial^2 \phi}{\partial x \partial y}\right|_{c_i} \widehat{xy} + \left.\frac{\partial^2 \phi}{\partial y \partial z}\right|_{c_i} \widehat{yz} + \left.\frac{\partial^2 \phi}{\partial x \partial z}\right|_{c_i} \widehat{xz}
\end{aligned} \tag{32}$$

with

$$\begin{aligned}
\widehat{x^n y^m z^p} &= \frac{1}{V_j}\int_{V_j} \Big((x - x_{c_j}) + (x_{c_j} - x_{c_i})\Big)^n \Big(((y - y_{c_j}) + (y_{c_j} - y_{c_i})\Big)^m \Big(((z - z_{c_j}) + (z_{c_j} - z_{c_i})\Big)^p \, dV (33) \\
&= \sum_{r=0}^{p} \left\{ \frac{p!}{r!(p-r)!} (z_{c_j} - z_{c_i})^r \sum_{l=0}^{m} \left\{ \frac{m!}{l!(m-l)!} (y_{c_j} - y_{c_i})^l \sum_{l=0}^{m} \left[\frac{n!}{k!(n-k)!} (x_{c_j} - x_{c_i})^k \overline{x^{n-k} y^{m-l} z^{p-r}} \right] \right\} \right\}
\end{aligned}$$

The overdetermined system of equations (29,32) can be written in matrix form as

$$\mathbf{\Delta}\phi = \mathbf{S}\,d\phi \tag{34}$$

where

$$\mathbf{\Delta}\phi = \begin{bmatrix} 0 \\ \overline{\phi}_1 - \overline{\phi}_{c_i} \\ \overline{\phi}_2 - \overline{\phi}_{c_i} \\ \cdots \\ \cdots \\ \overline{\phi}_{N_i} - \overline{\phi}_{c_i} \end{bmatrix} \tag{35}$$

$$S = \begin{bmatrix} \overline{x}_i & \overline{y}_i & \overline{z}_i & \overline{x^2}_i & \overline{y^2}_i & \overline{z^2}_i & \overline{xy}_i & \overline{yz}_i & \overline{xz}_i \\ \widehat{x}_1 & \widehat{y}_1 & \widehat{z}_1 & \widehat{x^2}_1 & \widehat{y^2}_1 & \widehat{z^2}_1 & \widehat{xy}_1 & \widehat{yz}_1 & \widehat{xz}_1 \\ \widehat{x}_2 & \widehat{y}_2 & \widehat{z}_2 & \widehat{x^2}_2 & \widehat{y^2}_2 & \widehat{z^2}_2 & \widehat{xy}_2 & \widehat{yz}_2 & \widehat{xz}_2 \\ & & & & \cdots & & & & \\ & & & & \cdots & & & & \\ \widehat{x}_{N_i} & \widehat{y}_{N_i} & \widehat{z}_{N_i} & \widehat{x^2}_{N_i} & \widehat{y^2}_{N_i} & \widehat{z^2}_{N_i} & \widehat{xy}_{N_i} & \widehat{yz}_{N_i} & \widehat{xz}_{N_i} \end{bmatrix} \tag{36}$$

$$d\phi = \begin{bmatrix} \frac{\partial\phi}{\partial x} & \frac{\partial\phi}{\partial y} & \frac{\partial\phi}{\partial z} & \frac{\partial^2\phi}{\partial x^2} & \frac{\partial^2\phi}{\partial y^2} & \frac{\partial^2\phi}{\partial z^2} & \frac{\partial^2\phi}{\partial x\partial y} & \frac{\partial^2\phi}{\partial y\partial z} & \frac{\partial^2\phi}{\partial z\partial x} \end{bmatrix} \tag{37}$$

The system of equation (34) becomes:

$$(\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T\mathbf{\Delta}\phi = \mathbf{C}\,\mathbf{\Delta}\phi = d\phi \tag{38}$$

where the **C** matrix has dimension $(\mathrm{N_i}+1)\,\mathrm{x}\,(\mathrm{N_i}+1)$, it contains only geometric constants, and it may be computed and stored in a preprocessing stage. Figure 4 shows, for a scalar centroid, the interpolation stencil used for the determination of the least square system (34).

Application of boundary conditions

Boundary conditions are imposed in Eqn. (34) by prescribing the values of a variable or its derivatives on specific auxiliary points of the boundary surface. Each point is created for the cut cell, by finding the intersection point $\mathbf{x}_{i_b}$ of the line passing from the volume of fluid centroid having the direction of the mean normal to the solid surface ∂B. Given the mean normal of the boundary cutting surfaces Eqn. (21), if one of the normals of the triangulated boundary surface (e.g., the edge of a cube intersecting a cartesian grid), forms an angle greater then a fixed value (30° in this work), other boundary points are added as boundary constraints (surface centroids). For example, the velocities no-slip condition for a non-moving body (a Dirichlet boundary condition) is imposed at these auxiliary points $\mathbf{x}_{i_b}$ by means of $\phi^{INT}(\mathbf{x}_{i_b}) = \phi(\mathbf{x}_{i_b}) = 0$, $i_b = 1..N_b$ (N_b is the number of boundary points in the interpolation cloud) in the vector $\Delta\phi$ of Eqn. (34). Since

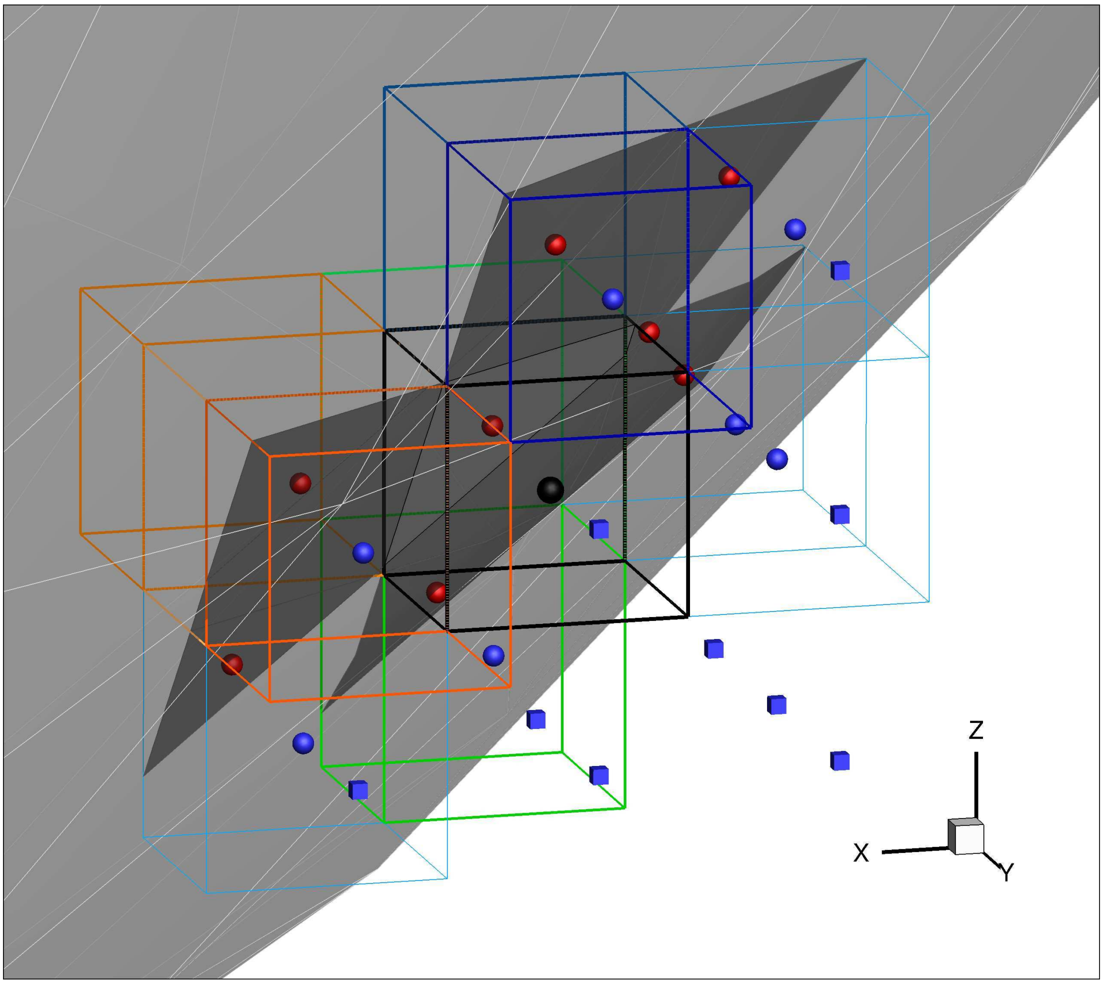

Figure 4: Example of points cloud used for the interpolation of scalar variables. White line: stl surface triangulation; black line: original cartesian volume of the cut cell and its boundary cutting surface; red bullets: surface boundary points; blue bullets: the serrounding volume of fluid centroids; blue cube: centroid of the second layer cells; cartesian edges: different colour shades indicate cut cells forming master-slave pair (green and orange).

$$\begin{aligned}\phi_{c_i}^{INT}(\mathbf{x}_{i_b}) = \phi_{c_i} + \left.\frac{\partial\phi}{\partial x}\right|_{c_i}\Delta x_{i_b} + \left.\frac{\partial\phi}{\partial y}\right|_{c_i}\Delta y_{i_b} + \left.\frac{\partial\phi}{\partial z}\right|_{c_i}\Delta z_{i_b} + \left.\frac{\partial^2\phi}{\partial x^2}\right|_{c_i}{\Delta x_{i_b}}^2 + \left.\frac{\partial^2\phi}{\partial y^2}\right|_{c_i}{\Delta y_{i_b}}^2 + \\ \left.\frac{\partial^2\phi}{\partial z^2}\right|_{c_i}{\Delta z_{i_b}}^2 + \left.\frac{\partial^2\phi}{\partial x\partial y}\right|_{c_i}\Delta x_{i_b}\Delta y_{i_b} + \left.\frac{\partial^2\phi}{\partial y\partial z}\right|_{c_i}\Delta y_{i_b}\Delta z_{i_b} + \left.\frac{\partial^2\phi}{\partial x\partial z}\right|_{c_i}\Delta x_{i_b}\Delta z_{i_b},\end{aligned} \tag{39}$$

with $\Delta x_{i_b} = x_{i_b} - x_{c_i}, \Delta y_{i_b} = y_{i_b} - y_{c_i}, \Delta z_{i_b} = z_{i_b} - z_{c_i}$, the system (34) becomes

$$
\mathbf{\Delta}\phi = \begin{bmatrix} 0 \\ \phi(\mathbf{x}_{1_b}) - \overline{\phi}_{c_i} \\ \phi(\mathbf{x}_{2_b}) - \overline{\phi}_{c_i} \\ \dots \\ \phi(\mathbf{x}_{N_b}) - \overline{\phi}_{c_i} \\ \overline{\phi}_{N_b+1} - \overline{\phi}_{c_i} \\ \dots \\ \dots \\ \overline{\phi}_{N_i} - \overline{\phi}_{c_i} \end{bmatrix} \tag{40}
$$

$$
S = \begin{bmatrix}
\overline{x}_i & \overline{y}_i & \overline{z}_i & \overline{x^2}_i & \overline{y^2}_i & \overline{z^2}_i & \overline{xy}_i & \overline{yz}_i & \overline{xz}_i \\
\Delta x_{1_b} & \Delta y_{1_b} & \Delta z_{1_b} & \Delta x^2_{1_b} & \Delta y^2_{1_b} & \Delta z^2_{1_b} & \Delta x_{1_b}\Delta y_{1_b} & \Delta y_{1_b}\Delta z_{1_b} & \Delta x_{1_b}\Delta z_{1_b} \\
\Delta x_{2_b} & \Delta y_{2_b} & \Delta z_{2_b} & \Delta x^2_{2_b} & \Delta y^2_{2_b} & \Delta z^2_{2_b} & \Delta x_{2_b}\Delta y_{2_b} & \Delta y_{2_b}\Delta z_{2_b} & \Delta x_{2_b}\Delta z_{2_b} \\
 & & & & \dots & & & & \\
\Delta x_{N_b} & \Delta y_{N_b} & \Delta z_{N_b} & \Delta x^2_{N_b} & \Delta y^2_{N_b} & \Delta z^2_{N_b} & \Delta x_{N_b}\Delta y_{N_b} & \Delta y_{N_b}\Delta z_{N_b} & \Delta x_{N_b}\Delta z_{N_b} \\
\widehat{x}_1 & \widehat{y}_1 & \widehat{z}_1 & \widehat{x^2}_1 & \widehat{y^2}_1 & \widehat{z^2}_1 & \widehat{xy}_1 & \widehat{yz}_1 & \widehat{xz}_1 \\
\widehat{x}_2 & \widehat{y}_2 & \widehat{z}_2 & \widehat{x^2}_2 & \widehat{y^2}_2 & \widehat{z^2}_2 & \widehat{xy}_2 & \widehat{yz}_2 & \widehat{xz}_2 \\
 & & & & \dots & & & & \\
 & & & & \dots & & & & \\
\widehat{x}_{N_i} & \widehat{y}_{N_i} & \widehat{z}_{N_i} & \widehat{x^2}_{N_i} & \widehat{y^2}_{N_i} & \widehat{z^2}_{N_i} & \widehat{xy}_{N_i} & \widehat{yz}_{N_i} & \widehat{xz}_{N_i}
\end{bmatrix} \tag{41}
$$

The red and blue colors of the matrix coefficients in Eqn. (40) and (41) refers to the points showed in Fig. 4 representing boundary points and volume of fluid centroids respectively.

After calculating the gradient of Eqn. 27 at a boundary point $\mathbf{x}_{i_b}$, multiplying it by the component of the local normal direction $\mathbf{n}_{i_b}$, it is possible to evaluate the normal derivative and impose a Neumann boundary condition $\frac{\partial \phi(\mathbf{x}_{i_b})}{\partial n} = g_n(\mathbf{x}_{i_b})$:

$$
\begin{aligned}
\nabla \phi^{INT}_{c_i}(\mathbf{x}_{i_b}) \cdot \mathbf{n_{i_b}} = {} & (\frac{\partial \phi}{\partial x}\bigg|_{c_i} + \frac{\partial^2 \phi}{\partial x^2}\bigg|_{c_i} \Delta x_{i_b} + \frac{\partial^2 \phi}{\partial x \partial y}\bigg|_{c_i} \Delta y_{i_b} + \frac{\partial^2 \phi}{\partial x \partial z}\bigg|_{c_i} \Delta x_{i_b})\, n_{i_{bx}} + \\
& +(\frac{\partial \phi}{\partial y}\bigg|_{c_i} + \frac{\partial^2 \phi}{\partial y^2}\bigg|_{c_i} \Delta y_{i_b} + \frac{\partial^2 \phi}{\partial x \partial y}\bigg|_{c_i} \Delta x_{i_b} + \frac{\partial^2 \phi}{\partial y \partial z}\bigg|_{c_i} \Delta z_{i_b})\, n_{i_{by}} + \\
& +(\frac{\partial \phi}{\partial z}\bigg|_{c_i} + \frac{\partial^2 \phi}{\partial z^2}\bigg|_{c_i} \Delta z_{i_b} + \frac{\partial^2 \phi}{\partial y \partial z}\bigg|_{c_i} \Delta y_{i_b} + \frac{\partial^2 \phi}{\partial x \partial z}\bigg|_{c_i} \Delta x_{i_b})\, n_{i_{bz}} = g_n(\mathbf{x}_{i_b})
\end{aligned} \tag{42}
$$

In this case, the Least Square Matrix of the metric coefficients $\mathbf{S}$ becomes:

$$S = \begin{bmatrix} \overline{x}_i & \overline{y}_i & \overline{z}_i & \overline{x^2}_i & \overline{y^2}_i & \overline{z^2}_i & \overline{xy}_i & \overline{yz}_i & \overline{xz}_i \\ n_{x1_b} & n_{y1_b} & n_{z1_b} & \Delta x_{1_b} n_x & \Delta y_{1_b} n_y & \Delta z_{1_b} n_z & \Delta y_{1_b} n_x + \Delta x_{1_b} n_y & \Delta y_{1_b} n_y + \Delta z_{1_b} n_y & \Delta x_{1_b} n_z + \Delta z_{1_b} n_x \\ n_{x2_b} & n_{y2_b} & n_{z2_b} & \Delta x_{2_b} n_x & \Delta y_{2_b} n_y & \Delta z_{2_b} n_z & \Delta y_{2_b} n_x + \Delta x_{2_b} n_y & \Delta y_{2_b} n_y + \Delta z_{2_b} n_y & \Delta x_{2_b} n_z + \Delta z_{2_b} n_x \\ & & & & \dots & \dots & \dots & & \\ n_{xN_b} & n_{yN_b} & n_{zN_b} & \Delta x_{N_b} n_x & \Delta y_{N_b} n_y & \Delta z_{N_b} n_z & \Delta y_{N_b} n_x + \Delta x_{N_b} n_y & \Delta y_{N_b} n_y + \Delta z_{N_b} n_y & \Delta x_{N_b} n_z + \Delta z_{N_b} n_x \\ \widehat{x}_1 & \widehat{y}_1 & \widehat{z}_1 & \widehat{x^2}_1 & \widehat{y^2}_1 & \widehat{z^2}_1 & \widehat{xy}_1 & \widehat{yz}_1 & \widehat{xz}_1 \\ \hat{}x_2 & \widehat{y}_2 & \widehat{z}_2 & \widehat{x^2}_2 & \widehat{y^2}_2 & \widehat{z^2}_2 & \widehat{xy}_2 & \widehat{yz}_2 & \widehat{xz}_2 \\ & & & & \dots & \dots & \dots & & \\ & & & & \dots & \dots & \dots & & \\ \widehat{x}_{N_i} & \widehat{y}_{N_i} & \widehat{z}_{N_i} & \widehat{x^2}_{N_i} & \widehat{y^2}_{N_i} & \widehat{z^2}_{N_i} & \widehat{xy}_{N_i} & \widehat{yz}_{N_i} & \widehat{xz}_{N_i} \end{bmatrix} \tag{43}$$

while the vector $\Delta\phi$ is:

$$\boldsymbol{\Delta}\phi = \begin{bmatrix} 0 \\ g(\mathbf{x}_{1_b}) \\ g(\mathbf{x}_{2_b}) \\ \dots \\ g(\mathbf{x}_{N_b}) \\ \overline{\phi}_{N_b+1} - \overline{\phi}_{c_i} \\ \dots \\ \dots \\ \overline{\phi}_{N_i} - \overline{\phi}_{c_i} \end{bmatrix} \tag{44}$$

In this work, no slip boundary conditions for all the three velocity components and Neumann boundary conditions $\frac{\partial\phi(\mathbf{x}_{i_b})}{\partial n} = 0$) for density, energy and species mass fraction, are applied at boundary points (determined as described in the initial section of this paragraph) , while Neumann boundary conditions are enforced at Gauss boundary integration points for pressure.

Fluxes calculation

Integrating Eqns. (1-4) over the volume V_i of the cut cell and using the divergence theorem:

$$\frac{\partial\overline{\mathbf{Q}}}{\partial t} = \frac{1}{V_i}\int_{\partial B_i} \mathbf{F}\cdot\mathbf{n}dA + \mathbf{S} \tag{45}$$

$$\tag{46}$$

where $\overline{\mathbf{Q}} = [\rho, \rho\mathbf{u}, \rho\mathcal{U}, \rho Y_i]^T$, is the vector of conserved variables, $\mathbf{S} = [0, 0, 0, \rho\dot{\omega}]^T$, $\mathbf{F} = \mathbf{F}^{inv} + \mathbf{F}^v$ being the flux vector containing an inviscid part $\mathbf{F}^{inv}$ and a viscous part $\mathbf{F}^v$, and $\mathbf{n}$ the outward unit normal vector to the cut cell surface dA.

The inviscid surface integral in Eqn. (45) is approximated as:

$$\frac{1}{V_i}\int_{\partial B_i} \mathbf{F}^{inv}\cdot\mathbf{n}dA = \frac{1}{V_i}\sum_{j=1}^{N_f}\int_{\partial A_j} \mathbf{F}^{inv}_{n_j}(\mathbf{Q}^L, \mathbf{Q}^R)\, dA_j \tag{47}$$

$$\tag{48}$$

Here, $\mathbf{F}^{inv}_{n_j}(\mathbf{Q}^L, \mathbf{Q}^R)$ represents the numerical convective flux in the direction normal to the face A_j, ($\mathbf{n_j}$ being the outword versor normal to A_j face) as a function of the reconstructed solution $\mathbf{U}^{\mathbf{L/R}}$ on both sides of A_j. The superscripts "R" and "L" refer to the spatial limit respectively on the outside and inside of the cut cell V_i with respect to its face A_j. In particular $\mathbf{Q}^L$ represents the solution calculated on the face A_j using the interpolation function ϕ_i^{INT} in V_i, while $\mathbf{Q}^R$ represents the reconstructed solution calculated on A_j using the interpolation function ϕ_j^{INT} in the neighbouring cell V_j (V_j cell may be a cut cell or a second layer cell).

Since the flux $\mathbf{F}_{nj}^{inv}(\mathbf{Q}^L, \mathbf{Q}^R)$ varies along the face A_j (quadratically in this work), it must be evaluated at each Gauss point $\mathbf{x}_g$ (or at the face centroid when linear reconstruction is adopted). The flux is formulated using a modified version of the advection upstream splitting method (AUSM) [41]. In this method, the inviscid flux is split into a convective component and a pressure term involving the Mach number $M_g = v_g/a_g$, such that the numerical inviscid flux $\mathbf{F}_{\mathrm{j}}(\mathbf{x}_{\mathrm{g}})$, can be computed as

$$\mathbf{F}_j(\mathbf{x}_{g_j}) = \frac{1}{2}\left\{M_j^m[(\mathbf{f}_j)^L + (\mathbf{f}_j)^R] + |M_j^m|[(\mathbf{f}_j)^L - (\mathbf{f}_j)^R]\right\} + p^m \tag{49}$$

where

$$M_j^m = 0.5((\mathbf{f}_j)^L + (\mathbf{f}_j)^R) \tag{50}$$

and

$$\mathbf{f}_j = \begin{pmatrix} \rho c \\ \rho c\mathbf{u} \\ c(\rho\mathcal{U} + p) \\ c\rho Y_n \end{pmatrix}.$$

$\rho, c, \mathbf{u}, \mathcal{U}, Y_i$ being density, sound velocity, velocity vector, total energy, species mass fraction respectively. The pressure term p^m is computed such that

$$p^m = \left\{(p^m)^L\left[\frac{1}{2} + \chi\left(M_j^m\right)^L\right] + (p^m)^R\left[\frac{1}{2} - \chi\left(M_j^m\right)^R\right]\right\}\begin{pmatrix} 0 \\ \mathbf{u}^m \\ 0 \\ 0 \end{pmatrix}. \tag{51}$$

where a dissipative splitting at $\chi = 0.5$ is used to dump spurious oscillations.

With the details of the gradient expression in mind (eq.42), viscous flux $\mathbf{F}^v$ in Eqn. 45 are now derived. First, it is calculated the gradient at the centroid $\mathbf{x}^{A_j}$ of the face A_j (the gradient varies linearly inside each cut cell) from the reconstruction functions ϕ_i^{INT} in V_i and ϕ_j^{INT} in the neighbouring cell V_j. The surface gradient is computed as a distance weighted convex combination of cell center gradients:

$$\nabla\boldsymbol{\phi}(\mathbf{x}_{\boldsymbol{f}}) = w^i\nabla\boldsymbol{\phi}_i^{INT}(\mathbf{x}_f) + w^j\nabla\boldsymbol{\phi}_j^{INT}(\mathbf{x}_f) \tag{52}$$

with

$$\begin{cases} w^i = \frac{|\mathbf{x}_{c_i} - \mathbf{x}_f|}{|\mathbf{x}_{c_i} - \mathbf{x}_f| + |\mathbf{x}_{c_j} - \mathbf{x}_f|} \\ w^j = 1 - w^i \end{cases} \tag{53}$$

Due to the staggering, for each type of cut cell V_i (velocity or scalars) and for each of the 6 possible wetted polyhedron' faces, the index of the corresponding neighbouring cell V_j is required to reconstruct the left and right solutions. All the required indexes are stored in a pre-computed list (see Fig. 5), readily available at runtime calculation.

Figure 5 shows the neighbouring centroids (red bullets) of the U_z interpolation function used to calculate the left and right states on the positive (A,B centroids) and negative (B,C centroids) faces with normal versor parallel to the z direction of a ρ cut cell (the centroid D, indicated with a blue open circle).

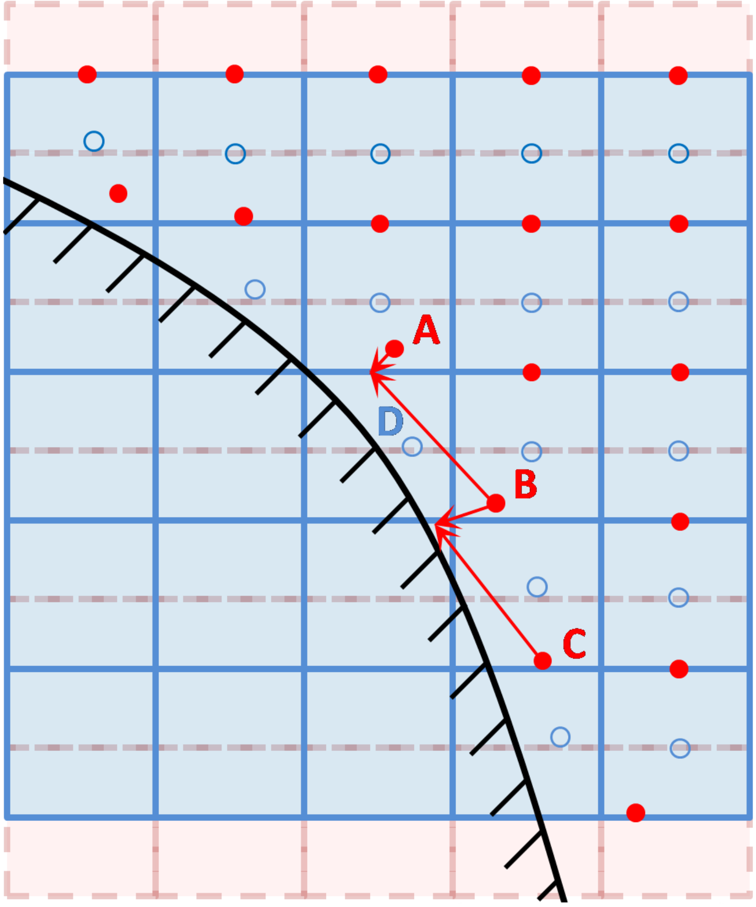

Figure 5: Interpolation functions of ρU_z used to calculate left and right states on density cut cell. Blue open circles: density centroids; red circles: ρU_z centroids; red dot-line grid: original cartesian ρU_z grid; blue line grid: original cartesian ρ grid; black line: boundary surface.

To compute wall shear stress on the boundary surfaces of no-slip walls of a cut cell V_i, the components of the stress tensor $\tau(\mathbf{x}_{b_i})$ $(i = 1..N_p)$ must be calculated at the cut cell's boundary face centroids. In the general case of a boundary surface constituted by N_p polygons with different normals (this is the case of high curvature boundary surfaces immersed in coarse cartesian grid), the expression of the viscous flux integral related to the boundary surface in Eqns. (45) is approximated as

$$\frac{1}{V_i}\int_{\partial B_w} \mathbf{F}^v \cdot \mathbf{n} dA = \frac{1}{V_i}\sum_{k=1}^{N_p} \tau(\mathbf{x}_{b_k}) \cdot \mathbf{n}_k \, dA_k \tag{54}$$

$$\tag{55}$$

with ∂B_w the wall boundary surface of the cut cell V_i, $\mathbf{n}_k$ the versor of $k-th$ boundary polygonal surface, τ the stress tensor calculated specifing velocity gradients $\nabla u_x^{INT}(x,y,z)$, $\nabla u_y^{INT}(x,y,z)$, $\nabla u_z^{INT}(x,y,z)$ at the $k-th$ boundary surface centroids $\mathbf{x}_{b_k}$.

In order to couple the finite volume method of the Immersed Volume method and the finite difference general code, the convective or diffusive term $\mathbf{F}$ in the transport equation for a second layer cell is calculated as:

$$F_i = \frac{f_{i,p}^{IVM} c_{i,p} + f_{i,n}^{IVM} c_{i,n} + f_{i,p}^{fd}(1-c_{i,p}) + f_{i,n}^{fd}(1-c_{i,n})}{Vol_{2l}} \tag{56}$$

where Vol_{2l} is the volume of the second layer cell, $f_{i,p}^{IVM}$ is the flux, on the positive face p (n, stands for the negative one) with normal i, calculated by the finite volume solver of the IVM method, $f_{i,p}^{fd}$ that calculated by the general finite difference code and the coefficient $c_{i,p(n)}$ is 1 if the positive (negative) face is in contact with a cut cell, 0 otherwise.

Numerical results and validation

The accuracy and robustness of the Immersed Volume Method is validated by computing two three-dimensional test cases. The solver has been fully parallelized using the Message Passing Interface (MPI) libraries such that parallel computations on shared and distributed memory systems are possible.

Flow past a cube at $Re = 215$

The flow past a cube is an appropriate validation test case, because of the presence of sharp edges boundaries, cut cells with internal volume and faces centroid. The Reynolds number based on the freestream velocity is defined as $Re_{edge} = \frac{\rho u L}{\mu_\infty} = 215$, with L being the edge lenght of the cube. For the three-dimensional simulation of a uniform flow past a cube, a computational domain Ω: [-2.5L, 2.5L]x[-2.5L, 2.5L]x[-2.5L, 7.5L] is used, with the midpoint of the cube located at the system origyn. The cartesian domain contains approximately 3.0 million of cells. The grid is refined near the cube edges in all directions. Computations for this Reynolds number found a symmetric and steady flow as shown in Fig. 6 where the instantaneous pressure and axial velocity contour around the cube are reported. The recirculating lenght is 2.73 ([42]), while the drug coefficient is 0.864 (0.866 from the classical fit Drug coefficient curve[42]).

Is clearly visible the expansion that occurs near the edge of the cube and the consequent speed increase (see Fig. 6). Despite the presence of cube's edges, and so of strong gradients of velocities, the streamtraces are very regular, (see Fig. 6) this confirm the stability of the new proposed method.

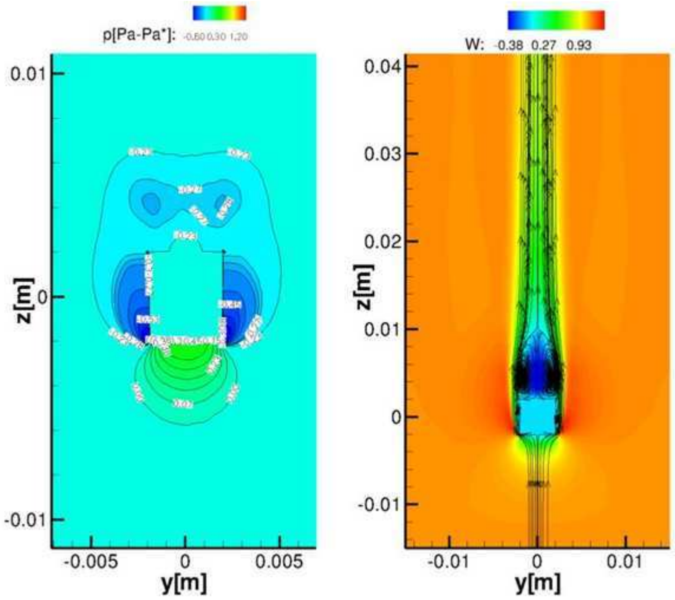

Figure 6: Flow past a cube: left) color map of pressure at the plane x-z, and y=0; right) snapshot of the axial velocity at the plane x-z, and y=0.

Flow past a sphere at $Re = 51500$

In this section the LES of the Bakic experiment [46] of the flow past a sphere with sting at $Re = 51500$ is presented. Several simulations of the flow past a sphere in the sub-critical regime have been carried out, contributing to a better understanding of fluid and vortex-shedding dynamics. Tomboulides et al. [47] performed time-accurate direct numerical simulation up to $Re = 1000$. Costantinescu et al. carried DES for studying the flow behind a sphere for the sub-critical and supercritical regimes at Reynolds numbers in the range of $10^4 - 10^6$ [48]. More recently Rodriguez et al. [49] performed DNS of the flow over a sphere in the subcritical regime at $Re = 3700$, determining the separation point and vortex shedding characteristics frequencies. The Reynolds number based on the freestream velocity is defined as $Re_D = \frac{\rho_\infty u_\infty D}{\mu_\infty} = 51500$, with D being the sphere diameter (0.0614 m). For the three-dimensional simulation of the flow past a sphere, a cartesian computational domain Ω: $[-1.7D, 5D]$x$[-4.88D, 4.88D]$x$[-4.88D, 4.88D]$ is used with [280]x[140]x[140] points in the z (streamwise), x and y directions respectively. The midpoint of the sphere is located at the system origyn. The grid is locally refined near the surface of the sphere, along the stick (that has a diamater d of $0.13D$) and near the separation regions. The free stream air velocity U_∞ is 12.6 m/s and the corresponding Mach number $Ma_\infty = U_\infty/c_\infty = 0.037$. The inlet turbulence level is 0.56% and these turbulent inflow boundary conditions are artificial prescribed by means of the Klein procedure [50].

Premixed CH_4/Air past a cube at $Re = 3200$

The flow past a cube with combustion is chosen as an appropriate validation test case, because of its sharp edge boundaries, cut cells with internal volume and faces' centroids and sharp density

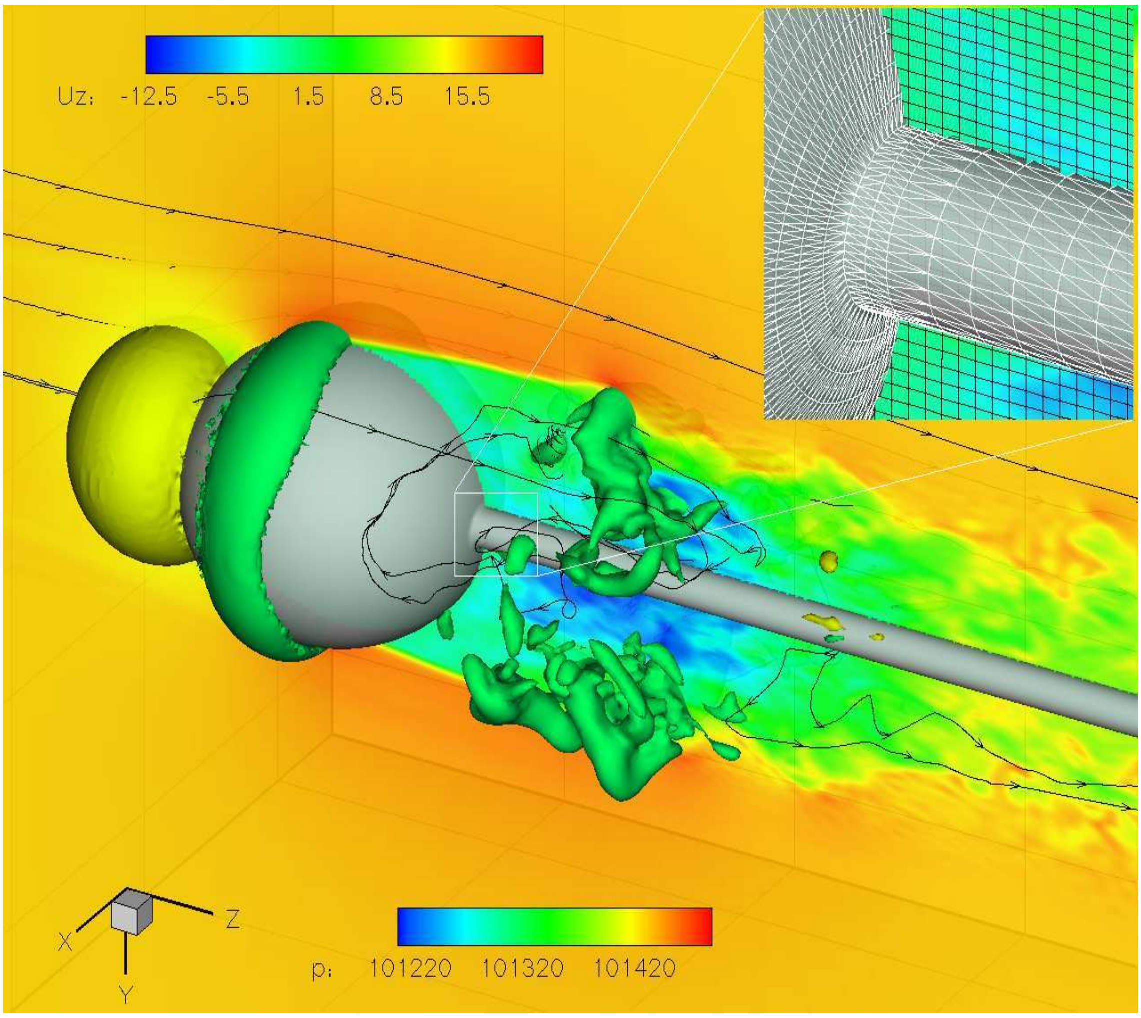

Figure 7: Instantaneous axial velocity; pressure isosurfaces at 101317 Pa and 101409 Pa. Zoomed in view: STL of the solid surface (white lines) and simulation grid (black lines).

and velocities gradients. The Reynolds number based on bulk quantities at the duct section (assuming its half width as reference length) crossing the cube leading edge is $Re_{LED} = 3200$ and the maximum Mach number is ~ 0.05. For the three-dimensional simulation of a uniform flow past a cube, a computational domain Ω:$[-2L, 3.5L]$x$[-2L, 2L]$x$[-2L, 2L]$ is used, L being the cube size.

A reduced kinetic mechanisms for the CH_4 with 5 chemical reacting species and 3 reactions is adopted. The premixed stoichiometric mixture is preheated up to $650K$ since with an inlet temperature of $300K$ the flame is stretched by high velocity gradients. The cartesian not uniform domain has 120x80x80 grid points in the axial and spanwise directions. It is clearly visible the separation region that occurs near the lower edge of the cube and the consequent formation of a lateral first recirculation region at the side walls immediately after the separations. The flame is attached in the second recirculation region downstream of the cube bluff-body and because the first lateral vortex is able to go up stream up to the leading edge of the cube (see Fig. 8).

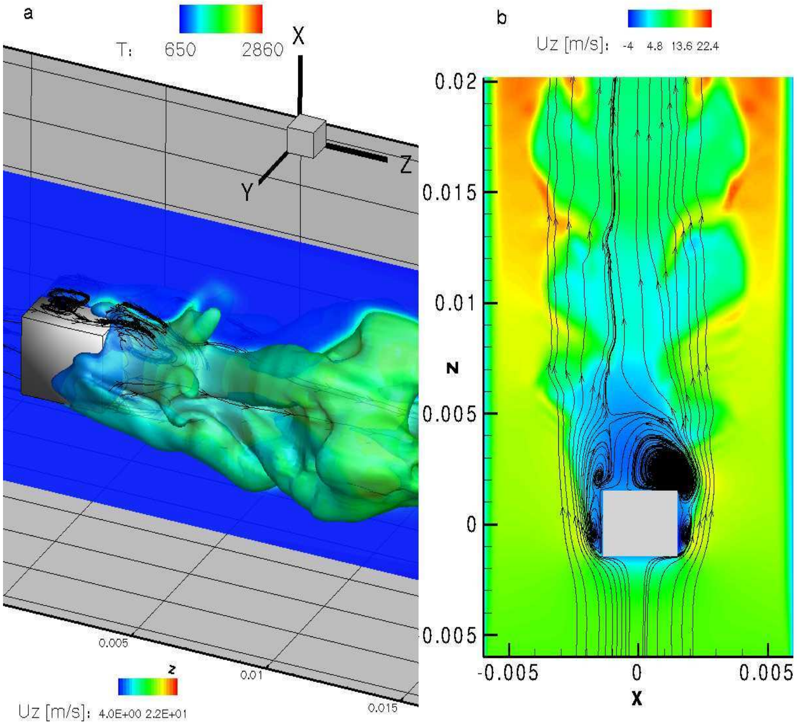

Figure 8: Premixed flame past a cube: a) Temperature isosurface (1500K) coloured by axial velocity $\mathrm{U_z}$ and Temperature slice; b) x-z plane of instantaneous U_z with streamlines.

0.1 Conclusions

A cut-cell based Cartesian grid method for three-dimensional compressible flows and not-uniform staggered grid is presented. The small-cell problem inherent in Cartesian cut-cell methods is solved using a cell-merging/cell-linking technique. The effective treatment of the small cells enabled the use of rather large CFL numbers in simulations. The accuracy of the viscous fluxes is second order. Along with this extension, the solver is currently being extended for combustion problems and heat transfer between the solid and the fluid flow.

Bibliography

[1] K. Akselvoll, P. Moin, Large-eddy simulation of turbulent confined coannular jets, J. Fluid Mech. 315 (1996) 387.

[2] C.D. Pierce, Progress-variable approach for large eddy simulation of turbulent combustion, Ph.D. thesis, Stanford University, 2001.

[3] S. Nagarajan, S.K. Lele, J.H. Ferziger, A robust high-order compact method for large eddy simulation, Journal of Computational Physics 191, (2003) 563-582.

[4] F.K. Chow, P. Moin, A further study of numerical errors in Large Eddy Simulations, Journal of Computational Physics 184, (2003) 366-380.

[5] S. Ghosal, An analysis of numerical errors in Large Eddy Simulation of turbulence, Journal of Computational Physics 125, (1996) 187-206.

[6] F.E.Ham,F.S. Lien, A.B. Strong, A fully conservative second-order finite difference scheme for incompressible flow on non unifrom grids, Journal of Computational Physics 191, (2002) 117-133.

[7] D.A. Kopriva, A staggered-grid multidomain spectral method for the compressible NavierâStokes equations, Journal of Computational Physics 143 (1998) 125.

[8] G.S. Djambazov, C.H. Lai, K.A. Pericleous, Staggered-mesh computation for aerodynamic sound, AIAA J. 38 (2000) 16.

[9] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, Journal of Computational Physics 161 (2000) 35-60.

[10] J. Mohd-Yusof, Combined immersed boundary/B-splines methods for simulations of flows in complex geometries, in: Center for Turbulence Research Briefs, NASA Ames/Stanford University, 1997.

[11] F. Muldoon, S. Acharya, A divergence free interpolation scheme for the immersed boundary method, Int. J. Numer. Method Fluid 56 (2008) 1845-1884.

[12] S. Kang, G. Iaccarino, P. Moin, Accurate immersed boundary reconstractions for viscous flow simulations, AIAA J. 47 (7) (2009) 1750-1760.

[13] D. Clarke, M. Salas, H. Hassan, Euler calculations for multi-elements airfoils using Cartesian grids, AIAA J. 24 (3) (1986) 353-358.

[14] M.J. Berger, R.J. Leveque, Stable boundary condition for cartesian grid calculations, Computer System in Engineering 1 (1990) 305-311.

[15] H. Johansen, P. Colella, A cartesian grid embedded boundary method for Poisson's equation on irregular domains, Journal of Computational Physics 147 (1998) 60-85.

[16] M.H. Chung, Cartesian cut cell approach for simulating incompressible flows with rigid bodies of arbitrary shape, Comput. Fluid 35 (2006) 607-623.

[17] M.P. Kirkpatrick, S.W. Armfield, J.H. Kent, A representaion of curved boundaries for the solution of the Navier-Stpkes equations on a staggered three-dimensional Cartesian grid, Journal of Computational Physics 184 (2003) 1-36.

[18] D. Heartmann, M. Meinke, W. Schroderm, A strictly conservative Cartesian cut-cell method for compressible viscous flows on adaptive grids, Comp. Methods Appl. Mech. Engrg. 200 (2011) 1038-1052.

[19] M. Mayer, A. Devesa, X.Y. Hu, N.A. Adams, A conservative immersed interface method for Large-Eddy Simulation of incompressible flows, Journal of Computational Physics 229 (2010) 6300-6317.

[20] X.Y. Hu, B.C. Khoo, N.A. Adams, F.L. Huang, A conservative interface method for compressible flows, Journal of Computational Physics 219 (2006) 553-578.

[21] Y. Cheny, O. Botella, The LS-STAG method: A new immersed/level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties, Journal of Computational Physics 229 (2010) 1043-1076.

[22] J.H. Sao, R. Mittal, A sharp-interface immersed boundary method with improved mass conservation and reduced spuroius pressure oscillations, Journal of Computational Physics 230 (2011) 7347-7363.

[23] 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.

[24] R.W.C.P. Verstappen, A.E.P. Veldman, Symmetry-preserving discretization of turbulent flow, Journal of Computational Physics 187 (2003) 343-368.

[25] C. Günther, D. Hartmann, M. Meinke, W. Schroder, A level-set based cut-cell method for flows with complex moving boundaies,V European Conference on Computational Fluid Dynamic, Lisbon, Portugal,14-17 June 2010.

[26] H. Ji, F.S. Lien, E. Yee, NUerical simulation of detonation using an adaptive Cartesian method combined with a cell-merging technique, Computers and Fluids, 39,6, (2010), 1041-1057.

[27] Yang G., Causon D.M., Ingram D.M., Saunders R., Batten P., A Cartesian cut cell method for compressible flows â part B: moving body problems. Aeronautical Journal 101 (1997) 57-65.

[28] Chiang Y., Van Leer B., Powell K.G., Simulation of unsteady inviscid flow on an adaptively refined Cartesian grid, AIAA Paper (1992) 92-0443-CP.

[29] M.J. Aftosmis, M.J. Berger, J.E. Melton, Robust and efficient cartesian mesh generation for component based geometry, Tech. Report AIAA-97-0196, US Air Force Wright Laboratory, (1997).

[30] T. Moller, A Fast Triangle-Triangle Intersection Test, Journal of Graphics Tools, 2(2), 1997.

[31] T. Poinsot, D. Vaynante, Theoretical and numerical combustion, 2012.

[32] E. Giacomazzi, V. Battaglia and C. Bruno, The Coupling of Turbulence and Chemistry in a Premixed Bluff-Body Flame as Studied by LES, Combustion and Flame, 138 (2004) 320-335.

[33] R.B. Bird, W.E. Stewart, E.N. Lightfoot, Transport Phenomena, Wiley International Edition, (2002).

[34] C.R. Wilke, J. Chem. Phys., 18, (1950), 517-9.

[35] R.J. Kee, G. Dixon-Lewis, J. Warnatz, M.E. Coltrin, Miller JA, Moffat HK, The CHEMKIN Collection III: Transport, San Diego, Reaction Design, (1998).

[36] S. Mathur, P.K. Tondon, S.C.Saxena, Molecular Physics, 12:569, (1967).

[37] E. Giacomazzi, F.R. Picchia, N.M. Arcidiacono, A Review on Chemical Diffusion, Criticism and Limits of Simplified Methods for Diffusion Coefficients Calculation, Combustion Theory and Modelling, (2008).

[38] C.W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, Journal of Computational Physics, 77, 439-471 (1988).

[39] D. Hartmann, M. Meinke, W. SchrÂ¨der, An adaptive multilevel multigrid formulation for Cartesian hierarchical grid methods, Comput. Fluids 37 (2008).

[40] C. O. Gooch, M.Alten, A High order accurate unstructured mesh Finite Volume scheme for the advection diffusion equation, Journal of Computational Physics, 181, 729-752 (2002).

[41] M.S. Liou, C.J. Steffen Jr., A new flux splitting scheme, Journal of Computational Physics 107 (1993) 23-39.

[42] A.K. Saha, Three dimensional numerical simulations of the transition of flow past a cube, Journal of Computational Physics 16, 2004.

[43] J.C. Mandal, S.P. Rao, High resolution finite volume computations on unstructured grids using solution dependent weighted least square gradients, Comput. Fluids, 2010.

[44] J.C. Mandal, J. Subramanian, On the link between weighted least-squares and limiters used in higher-order reconstructions for finite volume computations of hyperbolic equations, Appl Numer Math, 2008;58:705-25.

[45] D. Levin, The approximation power of moving least-squares, Math. Comput. 224 (1998), 1517-1531.

[46] V. Bakic, M. Schmid, B. Stankovic, Experimental Investigaion of Turbulent Structures of Flow Around a Sphere, Thermal Scinces, 10 (2006), 97-112.

[47] A. Tomboulides, S. Orszag, Numerical Investigation of Transitional and Weak Turbulent Flow past a Sphere, Journal of Fluid Mechanics, 416, (2000), 45-73.

[48] G. Costantinescu, K. Squires, Numerical Investigations of flow over a sphere in the subcritical and supercritical regimes, Phys. Fluids, 16, (2004), 1449-1466.

[49] I. Rodriguez, R. Borell, O. Lehmkuhl, C.D. Perez Segarra, A. Oliva, Direct Numerical Simulation of the flow over a sphere at $Re = 3700$, Journal of Fluid Mechanics, 679, (2011), 263-287.

[50] M. Klein, A. Sadiki, J. Janicka, A digital filter based generation of inflow data for spatially developing direct numerical or large eddy simulations, Journal of Computational Physics 186, (2003), 652-665.