Agenzia Nazionale per le Nuove Tecnologie,
l'Energia e lo Sviluppo Economico Sostenibile

Ministero dello Sviluppo Economico

RICERCA DI SISTEMA ELETTRICO

An Immersed Volume Method for Large Eddy Simulation of compressible flows

G. Rossi, N.M. Arcidiacono, F.R. Picchia, B. Favini

Report RdS/2011/211

AN IMMERSED VOLUME METHOD FOR LARGE EDDY SIMULATION OF COMPRESSIBLE FLOWS

D. Cecere, E. Giacomazzi, N.M. Arcidiacono, F.R. Picchia, F.Donato
ENEA
Settembre 2011

Report Ricerca di Sistema Elettrico
Accordo di Programma Ministero dello Sviluppo Economico – ENEA
Area: Produzione di energia elettrica e protezione dell'ambiente
Progetto: 2.2 – Studi sull'utilizzo pulito dei combustibili fossili, cattura e sequestro della CO_2

Responsabile Progetto: Antonio Calabrò, ENEA

ENEA		Pag. di Copia di

Unità UTTEI - COMSO	Classificazione UTTEI-COMSO **COMSO/2011/009IL /**	Distribuzione: Interna/Libera

Progetto

CERSE- Ricerca di Sistema Elettrico

AdP Ministero dello Sviluppo Economico – ENEA
Area: Produzione e fonti energetiche

Tema: *5.2.2.2 Tecnologie innovative per migliorare i rendimenti di conversione delle centrali a polverino di carbone*

Attività
Aumento dell'efficienza delle tecnologie di cattura della CO2 con produzione di elettricità "zero emission"

Parole chiave
- MILD Combustion
- Turbine a gas
- Trapped vortex
- Simulazione numerica

Titolo
An Immersed Volume Method for Large Eddy Simulation of compressible flows

Autori
D. Cecere, E. Giacomazzi, N.M. Arcidiacono, F.R. Picchia, F. Donato
ENEA UTTEI/COMSO

Sommario
Il documento descrive l'applicazione di una nuova tecnica , denominata Immersed Volume Method (IVM), per consentire il trattamento di geometrie complesse 3D da parte di un codice CFD di tipo compressibile alle differenze finite basato su griglie strutturate non uniformi e 'staggered'. La tecnica è stata implementata nel codice HeaRT e validata con un caso test significativo.

4		D.Cecere				S.Giammartini	
3		E.Giacomazzi					
2		N.M.Arcidiacono					
1		F.R. Picchia					
0		F. Donato					
Rev	Descrizione	Redazione	Data	Convalida	Data	Approvazione	Data

An Immersed Volume Method for Large Eddy Simulation of compressible flows

D. Cecere,[a], E. Giacomazzi[a], N.M. Arcidiacono[a], F.R. Picchia[a], F. Donato[a]

[a]*Sustainable Combustion Laboratories, ENEA, Rome, Italy.*

Abstract

In this work a technique for treating complex geometries in a compressible code using staggered non uniform cartesian grids and finite difference method is developed. The new method developed is called Immersed Volume Method (IVM) and consists in the application (to each scalar of the Navier Stokes equations) of a finite volume method in the cartesian cells cutted by the complex surface geometry. Accurate description of the real three dimensional geometry inside the cell volume is preserved by means of triangulated surface description (STL stereolithography) instead of approximating it by a plane. Since in the finite volume method, the geometric quantities of the cutted cell like face's areas, volume, volume centroid, face's areas centroids are needed, all these properties are evaluated by means of a specific code developed for this goal. In the Navier-Stokes solver, the finite volume method is applied to the cutted cells. The uxes are formulated using a modified version of the advection upstream splitting method (AUSM) originally proposed by Liou and Steffen. The recontruction of the variables at the cell interfaces, is done by means of a third order TVD (Total Variation Diminishing) interpolator. This choice is related to the future application of the method to combustors with complex geometries, and so the limitation of numerical wiggles for the chemical species is a must. In this way, the overall second-order accuracy of the base solver is preserved. The Large Eddy Simulation (LES) solver is parallelized using domain decomposition and message passing interface. The robustness and accuracy of the method is validated using LES of a flow past a cube.

Key words:
Complex Geometries, Cut-Cell Method, Finite Difference Staggered
Approach, Large Eddy Simulation.

1. Introduction

The development of Large Eddy Simulation (LES) as a methodology applied to a wide variety of turbulent flows, ranging from problems of scientific interest to those with engineering applications has been possible due to a rapid increase in computational power. In all engineering computational fluid dynamic problems in order to obtain satisfactory results the primary goals of a successful LES are accuracy, robustness and handling of complex three-dimensional geometry. In fact many flow physical problem involve geometrical complexities with irregular boundaries that usually are not alligned with the grid. Unstructured grids doesn't present constraint on the cell size and aspect ratio, but they are known to be not well suited for time-resolving turbulent flow computations as LES. The numerical method should not be sensitive to aliasing errors. Finite difference schemes provide good resolution while staggering of variables leads to improved robustness respect to a collocated approach without sacrificing conservation [1]. Total energy conservation, in fact, is guaranteed by the full conservative form of the governing equations. Staggered grids have been proposed earlier for solution of the compressible Navier-Stokes equations, recent research includes that of Kopriva [2] who uses a spectral method along with staggering of flux location with respect to the conserved variables, Djambazov et al. [3] that used a higher-order staggered (in space and time) method to solve the linearized Euler equations for computational aeroacoustics, Nagarajan et al. [1] that solved Navier-Stokes equations adopting high order compact schemes on staggered arrengements of the conserved variables. Two major class of methods are suitable for treating arbitrarily complex geometries with cartesian grids and distinguished on the basis of their approach to impose boundary conditions in the cells cut by the solid interface. The first is the classical Immersed Bounday (IB) methods where special interpolations are adopted to set the value of dependent variables in the cut cells [4; 5]. These methods are attractive because of their semplicity, but their major drowbacks are the occurrence of non-divergence free velocities in incompressible flows and spurius non physical pressure oscillations in compressible ones due to not observation of strict conservation of quantities as mass, momentum and kinetic energy near the irregular boundaries [6; 7]. A second class of IB methods is the cut-cell method (also called Cartesian grid methods) introduced first by Clarke [8]. The cut-cell method is based on a finite-volume discretization of the flow equations in the cells cut by the immersed interface and the discrete conservation is verified. For sharp-interface cartesian grid methods, the ”small cell problem” [9] of numerical instability would arise when finite difference or finite volume methods are applied to small-sized irregular cut grid cells. This may significantly restrict the time steps in temporal integration. Johansen and Colella [10] adopted a flux redistribution procedure. Most notable is the cell merging technique used by Chung [13] that link small cells and adjacent fluid cells to form a master-slave pair. Very few cut cell method for staggered grids have been reported in literature: Meyer et al. [11] proposed a conservative, second-order accurate Cartesian cut-cell for incompressible Navier-Stokes equations on three-dimensional non uniform staggered grids applicable to finite-volume discretization. To ensure numerical stability for small cells they follow the conservative mixing procedure by Hu et al. [12]. Cheny et al. [14] proposed a new IB method, based on the MAC method [15] for staggered 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. On a non-staggered grid, not only the velocity and pressure are colocated at nodes , but the position and geometry of the associated cells is also identical. With a staggered grid, the density cell and the cells associated with each of the three velocity components are at a different location and will generally have a different shape 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. The purpose of this work is to present a new efficient, conservative, second-order accurate Cartesian cut-cell method, called Immersed Volume Method (IVM) for the compressible Navier-Stokes equations solved by finite difference method on three-dimensional not 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 a triangulated surface (STL representation). The full geometrical characteristics of the cut cells are identified in a preprocessor procedure. Flow variables fluxes are computed in the cut cells and in a second adjacent layer to couple finite volume with finite difference method. The IVM method, by means of finite volume method, solves exactly the flow variables in the cut cells and links the velocities and energy fluxes to the thermodynamic variable changes and overcoming in this way the drowbacks of classical IB methods. The finite volume method is then applied also to a second layer to couple IVM to the finite difference method of the general code. The flow variables are stored at the cut-cell centroid and, to ensure numerical stability for small cells, the basic idea is to combine several neighbouring cells together so that the interfaces between merged cells are ignored and waves can travel in a newly combined larger cell without reducing the global time step [16; 17]. The paper is organized as follows. In section 2 we describe the procedures adopted for determining the geometrical characteristics of the cut cells. Section 3 presents the IVM discretization for the calculation of continuity, momentum and energy convective and diffusive fluxes. Section 4 is devoted to a numerical test on canonical non reactive flows at low Reynolds number for assessing the accuracy and robustness of the IVM method.

2. Cut cell characteristics evaluation

In the Immersed Volume Method a finite volume approach is applied to the two cell's layers adjacent to the solid surface. The first layer consists of cells that are directly cut from the solid surface, while the latter is formed by fluid cells that are in contact with cut cells. The presence of this second layer is needed to connect the finite volume method with the finite difference one.

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).

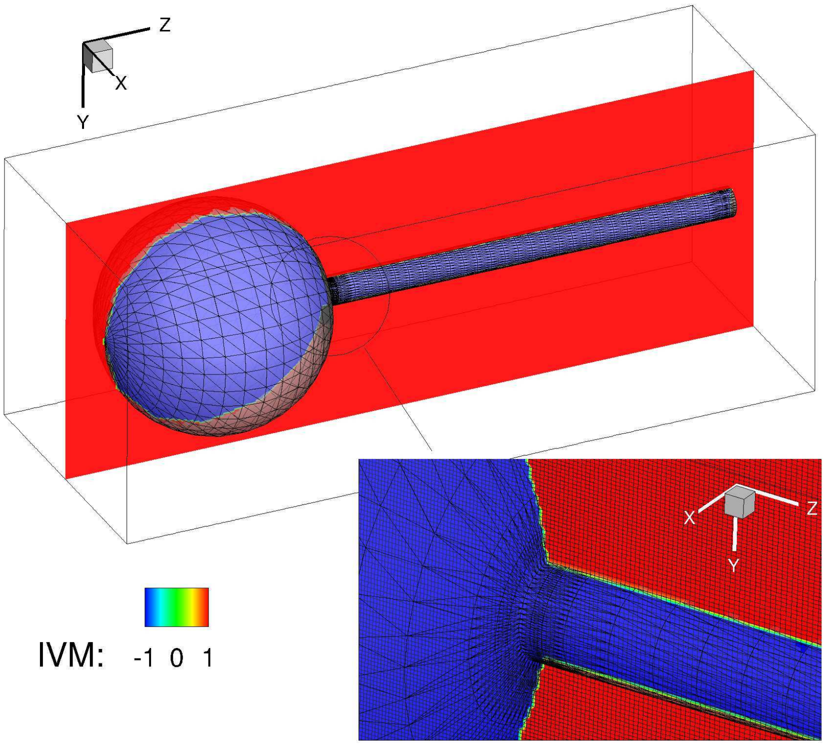

Figure 1: Top: A three-dimensional Cartesian grid illustrating the three types of cells in the cut-cell approach. The red region ($IVM = 1$) denotes the fluid cells which lie entirely outside the internal boundary, whereas the blue cells ($IVM = -1$) denote the solid cells which lie entirely inside the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

Because of the staggering arrangement, momenta are located half a cell width from thermodynamic variables and consequently four control volumes are defined associated to the three momenta and scalars (density, total energy, chemical species). Rather than storing the flow variables at a Cartesian cell center, the variables are collocated at the true cut-cell volume centroid (which always lies inside the fluid region) and the fluxes of these variables are estimated at the centroids of the faces bounding the cut-cell. In order to do this, for each field variable, the geometric characteristics of the cut control volume, obtained by the difference between the structured cell and the solid volume, have to be known. The geometric features of a 3D cartesian cut cell as the mass volume centroid, the volume of fluid, the polyhedron faces's areas and centroids are then used to evaluate interpolation of variables and to calculate the fluxes in the solution of the Navier-Stokes equations. A triangulated surface mesh is used to represent the boundary surface (see gray surface in Fig. 2). The vertices and the outside solid normal (outwards in the fluid region) of these triangles are stored in a StereoLithography file (STL).

In a first stage, after the production of the cartesian structured computational grid, a marker is assigned to each vertex of the parallelepiped cell (the grid may be not uniform) that determines whether the vertex is inside or outside the solid. A ray tracing procedure is applied in order to do this [18]. A ray is casted from a point **A** and the number of intersections with the solid triangulated surface is counted. The point **A** lies in 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. For each cartesian cell face, the intersection points of the solid triangulated surface with this two triangles are evaluated by a fast triangle/triangle intersection routine [19] and stored in a linking list associeted to the computational cell

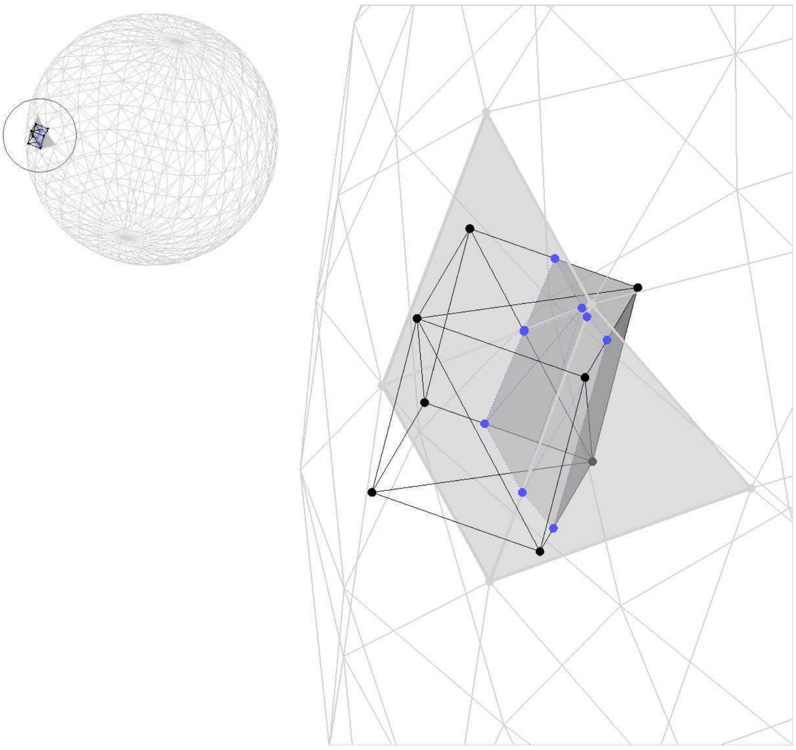

Figure 2: Example of STL boundary surface representation and cut cell. Black line: cut structured cell; Blu line: immersed boundary surface/cut cell intersection; gray line: immersed boundary surface represented by triangulation, gray solid: internal solid part of the immersed boundary.

face. For each cartesian cell face, the intersection points are ordered to form a polyline that divides the face into two polygons respectively in the fluid and solid region (remember that the vertices of the face are mapped as internal or external to the solid as shown for all points in Fig. 1). The wet polygons are triangularized by a Delaunay triangulation, where the intersection polyline of each face is adopted as a constrain. This boolean operations are performed for all the faces of the cell. The faces' wet polygons form a 3D polyhedron that is closed by the surface of the immersed solid boundary internal to the computational cell (see the blue lines and points in Fig. 2). This surface must be characterized and, in order to do this, for each triangle of the STL triangularization, (the STL triangles may lie in different planes), its intersection points and its internal vertices are stored in a second list associated to the cell. At this point the closed wet polyhedron is defined, with some faces lying in the computational faces' planes and others in contact with the solid surface. Applying the Gauss's divergence theorem, the wet volume may be calculated as:

$$V = \int_V dV = \int_S x \, dS \,, \tag{1}$$

Furthermore, the polyhedron mass wet centroid x_i^V $(i = 1\,, 3)$, which usually does not coincide with the volumetric center of the original square grid cell, can be calculated as:

$$x_i^V = \frac{1}{V}\int_V x_i \, dV = \frac{1}{2\,V}\int_S x_i^2 dS \tag{2}$$

$$= \frac{1}{2\,V}\sum_{k=1}^{N}\int_{S_k} x_i^2 \, dS = \frac{1}{2\,V}\sum_{k=1}^{N}\sum_{j=1}^{M_k} x_{i,c_j}^2 \, A_j \tag{3}$$

where N is the number of polyhedron's faces, M_k is the number of triangles of the polyhedron's face triangulation, x_{i,c_j} and A_j are respectevely the centroid and the area of the $j-th$ triangle. All these areas must be calculated with high precision, because these areas will be used in the calculation of the pressure momenta.

3. Governing Equations

Fluid dynamic 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. The derivation of these conservation equations from mass, species and energy balances may be found in standard books. For a mixture of N_s ideal gases in local thermodynamic equilibrium and chemical nonequilibrium, the corresponding field equations (extended Navier-Stokes equations) are:

- Transport Equation of Mass

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \tag{4}$$

- Transport Equation of Momentum

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}\boldsymbol{u}) = \nabla \cdot \boldsymbol{S} + \rho \sum_{i=1}^{N_s} Y_i \boldsymbol{f_i} \tag{5}$$

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

$$\frac{\partial \rho \mathcal{U}}{\partial t} + \nabla \cdot [\rho \boldsymbol{u} \mathcal{U}] = \nabla \cdot (\boldsymbol{S}\boldsymbol{u}) - \nabla \cdot \boldsymbol{q} + \rho \sum_{i=1}^{N_s} Y_i \boldsymbol{f_i} \cdot (\boldsymbol{u} + \boldsymbol{V_i}) \tag{6}$$

- Transport Equation of Species Mass Fraction

$$\frac{\partial \rho Y_i}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} Y_i) = -\nabla \cdot \boldsymbol{J_i} + \dot{\omega}_i \tag{7}$$

- Thermodynamic Equation of State

$$p = \rho \sum_{i=1}^{N_s} \frac{Y_i}{W_i} \mathcal{R}_u T \tag{8}$$

These equations must be coupled with the constitutive equations which describe the molecular transport properties of the flow.

In above equations, t is the time variable, ρ the density, $\boldsymbol{u}$ the velocity, $\boldsymbol{S}$ the stress tensor, $\mathcal{U}$ the total energy per unit of mass, $\mathcal{E}$ and $\mathcal{K}$ respectively the internal and mechanical (i.e., kinetic) energies per unit of mass, $\boldsymbol{q}$ is the heat flux, p the pressure, T the temperature. For what concerns chemical species, $\boldsymbol{f_i}$ is the body force per unit of mass acting on the species i, with molecular weight W_i and mass fraction Y_i, $\dot{\omega}_i$ is the production/destruction rate of species i, diffusing at velocity $\boldsymbol{V_i}$ and resulting in a diffusive mass flux $\boldsymbol{J_i}$. Finally, $\mathcal{R}_u$ is the universal gas constant.

Summation of all species transport equations (7) yields the total mass conservation equation (4). Therefore, the N_s species transport equations (7) and the mass conservation equation (4) are linearly dependent and one of them is redundant. Furthermore, to be consistent with mass conservation, the diffusion fluxes ($\boldsymbol{J_i} = \rho Y_i \boldsymbol{V_i}$) and chemical source terms must satisfy

$$\sum_{i=1}^{N_s} \boldsymbol{J_i} = 0 \qquad \textit{and} \qquad \sum_{i=1}^{N_s} \dot{\omega}_i = 0 \ . \tag{9}$$

In particular, the constraint on the summation of chemical source terms derives from mass conservation for each of the N_r chemical reactions of a chemical mechanism. With the tensor notation this mechanism can be written as

$$\nu^{'}_{ij} A_i = \nu^{''}_{ij} A_i \qquad \textit{with} \qquad i = 1, ..., N_s \qquad \textit{and} \qquad j = 1, ..., N_r \ , \tag{10}$$

where $\nu^{'}_{ij}$ and $\nu^{''}_{ij}$ are the stoichiometric coefficients of species i on the left ($^{'}$) and right side ($^{''}$) of the $j-th$ reaction. Since mass is given by the product of number of moles times molecular weight, mass conservation for each reaction, $m_j^{'} = m_j^{''}$, is written as

$$\sum_{i=1}^{N_s} \left(\nu_{ij}^{''} - \nu_{ij}^{'} \right) W_i = 0 \qquad with \qquad j = 1, ..., N_r \ . \tag{11}$$

The net source / sink term of the $i - th$ chemical species is

$$\dot{\omega}_i = \sum_{j=1}^{N_r} \dot{\omega}_{ij} = W_i \sum_{j=1}^{N_r} \left(\nu_{ij}^{''} - \nu_{ij}^{'} \right) \dot{\omega}_{Rj} \qquad with \qquad i = 1, ..., N_s \ , \tag{12}$$

$\dot{\omega}_{Rj}$ being the reaction rate associated to the $j - th$ reaction. Summing Eqn. (12) over the number of chemical species N_s,

$$\sum_{i=1}^{N_s} \dot{\omega}_i = \sum_{j=1}^{N_r} \dot{\omega}_{Rj} \sum_{i=1}^{N_s} W_i \left(\nu_{ij}^{''} - \nu_{ij}^{'} \right) = 0 \ , \tag{13}$$

after using Eqn. (11).

4. IVM method discretization

While the flux evaluation is straightforward in structured grid areas, it becomes more difficult on partial surfaces at fine-coarse cell interfaces and near boundaries. In the proposed formulation, the least-squares method is utilized to obtain a discretization scheme which is flexible in terms of the local mesh topology and the presence and shape of embedded boundaries.

4.1. Interface value calculation

The cell interface values of the solution variables are found using Taylor series expansion of solution variables about the cell center. The left and right state values of any scalar at the interface between two adjacent cells i and j is calculated as:

$$\phi_{L/R} = \phi_{i/j} + (\mathrm{r}_{int} - \mathrm{r}_{i/j})^T \nabla\phi_{i/j} + \frac{1}{2}(\mathrm{r}_{int} - \mathrm{r}_{i/j})^T H_{i/j}(\mathrm{r}_{int} - \mathrm{r}_{i/j}) + ... \quad (14)$$

where $\phi_{L/R}$ are the interpolated value of the scalar ϕ on the left and right interface's sides at position r_{int} from the cell centers ($\mathrm{r}_{i/j}$) values $\phi_{i/j}$ and $\nabla\phi_{i/j}$, $H_{i/j}$ are the gradients and the Hessian matrices of ϕ at the i and j cell centers respectively. In order to obtain a third order interpolation scheme the three first terms in the eqn. (29) must be retained and the calculation of nine derivatives is required up to mixed second derivatives. In this solution dependent weighted least square method (SDWLS) [22], the computational stencil is constructed looking at the 125 cell centers (the small cut cells are exluded) serrounding the point where the derivatives must be calculated, plus the intersections of the normal from the cut cell centroids with the solid surface. For these last points, the value of the scalar variable is determined by the choosen boundary condition. At this point, considering that a cell i contains n vertex neighbouring cells ($j = 1, ..., n$), the value of variable ϕ at these centroids can be expressed by means of Taylor's series expansion at the centroid i:

$$\begin{aligned}\phi_j = \phi_i + \frac{\partial\phi}{\partial x}\Delta x_j + \frac{\partial\phi}{\partial y}\Delta y_j + \frac{\partial\phi}{\partial z}\Delta z_j + \frac{\partial^2\phi}{\partial x^2}\frac{\Delta x_j}{2} + \frac{\partial^2\phi}{\partial y^2}\frac{\Delta y_j}{2} + \\ \frac{\partial^2\phi}{\partial z^2}\frac{\Delta z_j}{2} + \frac{\partial^2\phi}{\partial x\partial y}\Delta x_j\Delta y_j + \frac{\partial^2\phi}{\partial y\partial z}\Delta y_j\Delta z_j + \frac{\partial^2\phi}{\partial x\partial z}\Delta x_j\Delta z_j\end{aligned} \quad (15)$$

with $\Delta x_j, \Delta y_j, \Delta z_j$ the distances, along the three cartesian coordinates, between the j-th cell centroid and the centroid i where the nine derivatives are calculated. The overdetermined system of equations (15) can be written in matrix form as

$$\Delta\phi = Sdu \tag{16}$$

where

$$\Delta\phi = \begin{bmatrix} \phi_1 - \phi_i \\ \phi_2 - \phi_i \\ ... \\ ... \\ \phi_n - \phi_i \end{bmatrix} \tag{17}$$

$$S = \begin{bmatrix} \triangle x_1 & \triangle y_1 & \triangle z_1 & \frac{\triangle x_1^2}{2} & \frac{\triangle y_1^2}{2} & \frac{\triangle z_1^2}{2} & \triangle x_1 \triangle y_1 & \triangle y_1 \triangle z_1 & \triangle x_1 \triangle z_1 \\ \triangle x_2 & \triangle y_2 & \triangle z_2 & \frac{\triangle x_2^2}{2} & \frac{\triangle y_2^2}{2} & \frac{\triangle z_2^2}{2} & \triangle x_2 \triangle y_2 & \triangle y_2 \triangle z_2 & \triangle x_2 \triangle z_2 \\ & & & & ... & & & & \\ & & & & ... & & & & \\ \triangle x_n & \triangle y_n & \triangle z_n & \frac{\triangle x_n^2}{2} & \frac{\triangle y_n^2}{2} & \frac{\triangle z_n^2}{2} & \triangle x_n \triangle y_n & \triangle y_n \triangle z_n & \triangle x_n \triangle z_n \end{bmatrix} \tag{18}$$

$$d\phi = \begin{bmatrix} \frac{\partial\phi}{\partial x} & \frac{\partial\phi}{\partial y} & \frac{\partial\phi}{\partial z} & \frac{\partial^2\phi}{\partial x^2} & \frac{\partial^2\phi}{\partial y^2} & \frac{\partial^2\phi}{\partial z^2} & \frac{\partial^2\phi}{\partial x\partial y} & \frac{\partial^2\phi}{\partial y\partial z} & \frac{\partial^2\phi}{\partial z\partial x} \end{bmatrix} \tag{19}$$

A weighted least square formulation is adopted to calculate the nine derivative:

$$S^T W d\phi = S^T W \Delta\phi \tag{20}$$

where $W = diag(w_1, ...w_n)$ is the diagonal matrix of weights $w_i = 1/\Delta\phi_i^2$ [23]. Since, the equation (20) is not always well behaved, it is rewritten in equivalent normal form:

$$W^{1/2}\Delta\phi = (W^{1/2}S)d\phi \tag{21}$$

and solved by means of QR factorization. In Fig. 3 is represented, for a scalar centroid, the cloud interpolation points used in Eqn. (20) for the calculation of the gradient.

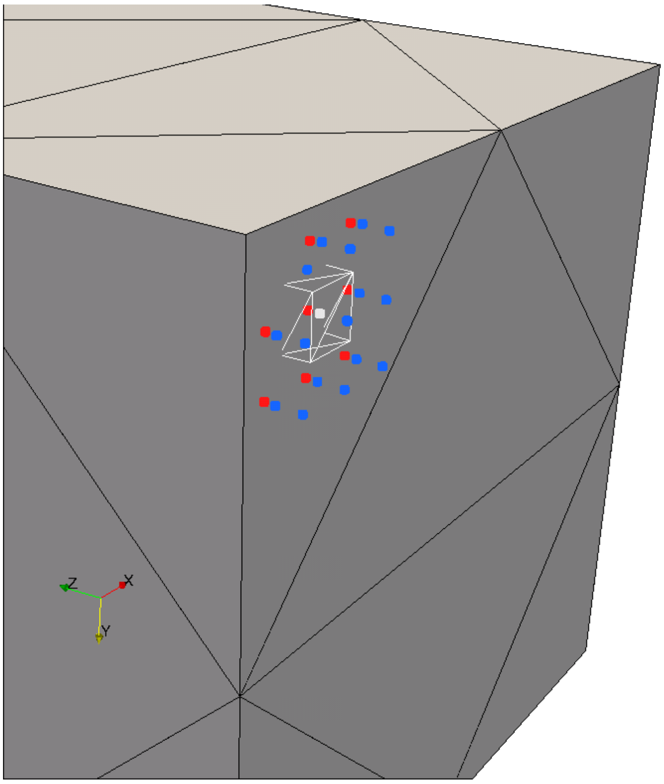

Figure 3: Example of points cloud used for the interpolation of scalar variables. White bullet: the centroid of the cutted cell; white line: volume of fluid of the cutted cell; red bullets: the interpolation points used for the boundary condition; blue bullets: the serrounding volume of fluid centroids.

4.2. Small cell treatment

After calculating all geometric properties of the cut cells, the problem of removing small cut cells must be solved. In fact, it can happen that the volume of some cut cell may be a very small fraction of that of the original uncut grid cell. Their small cell volume increases the stiffness of the system of equations and may result in numerical instabilities. In this work the solution proposed by Hartmann et al. [20] is adopted. Each small cell, linked and merged with a suitable master cell, is treated in the numerical method as passive cells. The algorithm to find the corresponding master cell is presented. Let $\mathbf{n}_s$, the mean normal versor of the solid surface, formed by $\mathrm{N_p}$ plane surfaces, that intersects the slave cell:

$$\mathbf{n}_s = \frac{\sum_{k=1}^{N_p} A_k \mathbf{n}_k}{\sum_{k=1}^{N_p} A_k} \tag{22}$$

where $\mathrm{A_k}$ is the $\mathrm{k-th}$ surface area with normal $\mathbf{n}_k$. The master cell is chosen in $i-th$ coordinate direction maximizing the dot product $\mathbf{n_s} \cdot (\pm \mathbf{n_i})$, $\mathbf{i} = 1 \dots 3$. In the case that the master cell m is a slave cell of another cell $m^{'}$, both s and m become slave cells of $m^{'}$.

After having established the connectivity between a master cell m and its slave cell(s) $\mathrm{N_s \epsilon S}$, the cell volume $V^{m'}$ and the centroid $\mathbf{x}^{m'}$ of the combined master-slave cell cluster $\mathrm{m'}$ are computed by:

$$V^{m'} = V^m + \sum_{k=1}^{N_s} V_k^s \tag{23}$$

$$\mathbf{x}^{m'} = \frac{\mathbf{x}^m V^m + \sum_{k=1}^{N_s} \mathbf{x}_k^s V_k^s}{V^{m'}} \ , \tag{24}$$

The data is copied to the slave cell(s) and the master cell according to

$$\mathbf{x}^{m} \leftarrow \mathbf{x}^{m+S} \tag{25}$$

$$\mathbf{x}^{s_i} \leftarrow \mathbf{x}^{m+S}, \forall s_i \epsilon S, \tag{26}$$

$$\sigma^{m} \leftarrow \sigma^{m+s_i}, \tag{27}$$

$$V^{m} \leftarrow V^{m+s_i} \tag{28}$$

where σ are the solid and fluid surface of the cuttes cell.

4.3. Application of boundary conditions

Boundary conditions are imposed by prescribing the primitive variables on specific auxiliary points each of which is created for the cut cell, by finding the intersection point $\mathbf{x}_{\Gamma}^{n}$ of the line passing from the volume of fluid centroid and the mean normal direction to the solid surface Γ as direction. For example, the velocities no-slip condition for a non-moving body is imposed at this auxiliary points by means of $\phi_n = 0$, $n = 1..N_b$ (N_b is the number of the boundary points in the interpolation cloud) in the vector $\Delta\phi$ of Eqn. 17. The value of $\Delta\phi_n = 0$, $n = 1..N_b$ is zero when a Neumann boundary condition must be implemented. In general, Neumann boundary conditions are used for the pressure, density and total Energy, and Dirichlet boundary conditions are used for the velocities as is the case for an adiabatic no-slip wall. Once all the scalar variable gradients are calculated, the inviscid and viscous fluxes may be evaluated on the cut cell face's centroids.

4.4. Inviscid flux calculation

The inviscid surface fluxes $\mathbf{F}_{\mathrm{j}}(\mathbf{x}_{\mathrm{m}})$ is computed based on the left and right interpolated states L and R in the surface centroid x_{m}. The left state L corresponds to information coming from the negative space direction, while the right state R corresponds to information coming from the positive space direction. The flux is formulated using a modified version of the advection upstream splitting method (AUSM) [21]. In this method, the inviscid flux is split into a convective component and a pressure term involving the Mach number $M_j = v_j/a$, such that the numerical inviscid flux $\mathbf{F}_{\mathrm{j}}(\mathbf{x}_{\mathrm{m}})$, $\mathrm{j} = 1, 2, 3$, on the surface m can be computed as:

$$\mathbf{F}_j(\mathbf{x}_m) = \frac{1}{2}\left\{M_j^m[(\mathbf{f}_j)^L + (\mathbf{f}_j)^R] + |M_j^m|[(\mathbf{f}_j)^L - (\mathbf{f}_j)^R]\right\} + p^m \tag{29}$$

where

$$M_j^m = 0.5((\mathbf{f}_j)^L + (\mathbf{f}_j)^R) \tag{30}$$

and

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

$\rho, \mathcal{U}, c, \mathbf{u}$ are density, total energy, sound velocity, and velocity vector respectively.

The pressure term p^m is computed such that

$$p^m = \left\{(p^m)^L\left[\frac{1}{2} + \chi\left(M_j^m\right)^L\right] + (p^m)^R\left[\frac{1}{2} - \chi\left(M_j^m\right)^R\right]\right\}\begin{pmatrix} 0 \\ \mathbf{u}^m \\ 0 \end{pmatrix}. \tag{31}$$

where a dissipative splitting at $\chi = 0.5$ is used to dump spurious oscillations. The left and right interpolated fluxes, the pressure and the field fariables are computed at third order accuracy using a monotone Total Variation Diminishing schemes for the primitive variables.

4.5. Viscous fluxes

The viscous flux F^v is computed by means of the cell center gradients available for the general interpolation routines. The surface gradient is computed as a distance weighted convex combination of cell center gradients:

$$\nabla \boldsymbol{\phi}^m = w^L \nabla \boldsymbol{\phi}^L + w^R \nabla \boldsymbol{\phi}^R \tag{32}$$

with

$$\begin{cases} w^L = \frac{|\mathbf{x}^L - \mathbf{x}^R|}{|\mathbf{x}^L - \mathbf{x}^R| + |\mathbf{x}^R - \mathbf{x}^L|} \\ w^R = 1 - w^L \end{cases} \tag{33}$$

This reconstruction is second order accurate on not uniform mesh.

5. Numerical results and validation

The accuracy of the presented Immersed Volume Method is validated by computing three-dimensional test case. 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. The three-dimensional simulation has been performed in parallel on up to 8 CPUs. A cube of edge 0.004 m is cut out of the domain to investigate the order of the discretization at complex boundaries. The cube is centered in the computational domain so that a constant-valued Dirichlet boundary condition can be specied on its boundary. The flow past a cube without visocity effect is an appropriate validation test case, because of the presence of edges, cut cells with internal volume and faces centroid and the absence of viscous stability effects. The Reynolds number based on the freestream velocity is dened as $Re_e dge = \frac{\rho u L}{\mu_\infty} = 40$. For the three-dimensional simulation of the ow past a cube, a computational domain Ω: [7.5L, 7.5L]x[-0.5L, 0.5L]x[-0.5L, 0.5L] is used. The cartesian domain contains approximately 2.0 million of cells. The grid is refined near the cube in all directions.

In Fig. 4 the contour of pressure around the cube is shown.

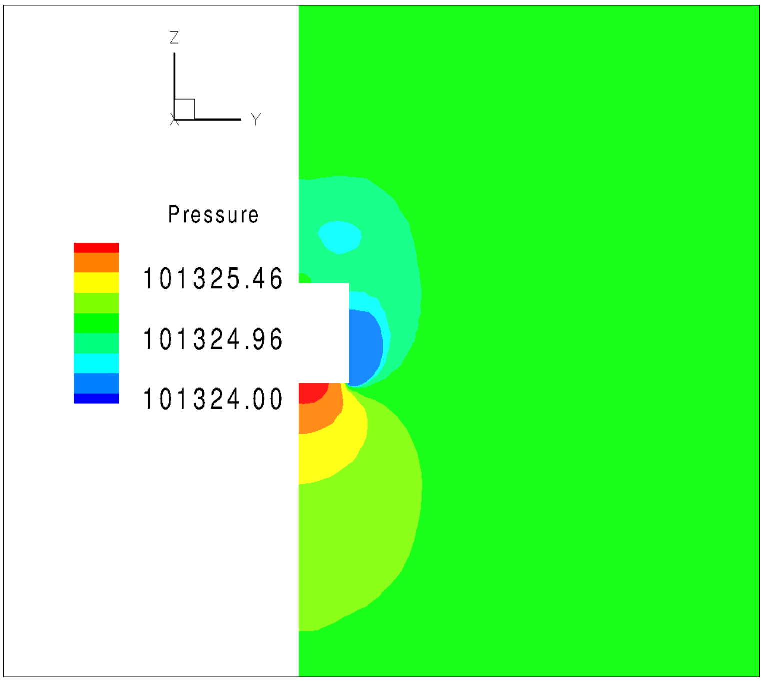

Figure 4: Flow past a cube: color map of pressure in a plane x-z, and y=0.

Is clearly visible the expansion that occurs near the edge of the cube and the conseguent speed increase (see Fig. 5). Despite the presence of cube's edges, and so of strong gradients of velocities, the streamtraces are very regular, (see Fig. 5) this confirm the stability of the new proposed method.

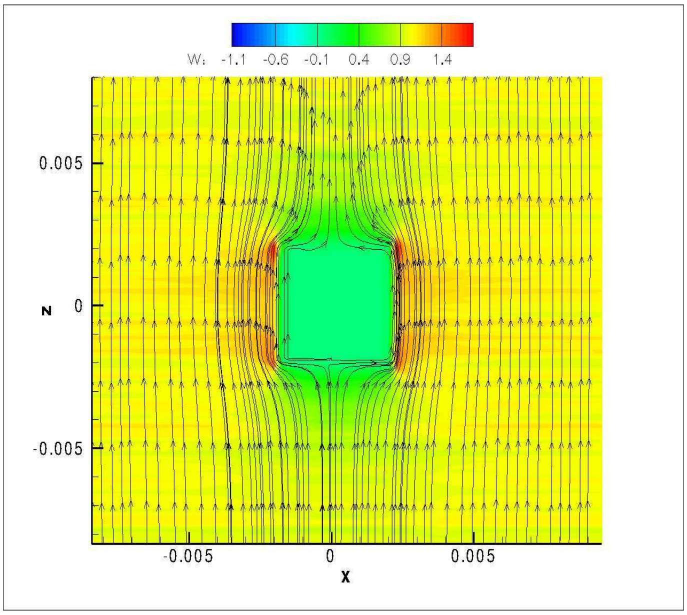

Figure 5: Flow past a cube: color map of axial velocity component and streamtraces in a plane x-z, and y=0.

6. 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.

References

[1] S. Nagarajan, S.K. Lele, J.H. Ferziger, A robust high-order compact method for large eddy simulation, J. Comput. Phys. 191, (2003) 563-582.

[2] D.A. Kopriva, A staggered-grid multidomain spectral method for the compressible NavierStokes equations, J. Comput. Phys. 143 (1998) 125.

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

[4] 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.

[5] 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.

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

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

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

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

[10] H. Johansen, P. Colella, A cartesian grid embedded boundary method for Poisson's equation on irregular domains, J. Comput. Phys. 147 (1998) 60-85.

[11] M. Mayer, A. Devesa, X.Y. Hu, N.A. Adams, A conservative immersed interface method for Large-Eddy Simulation of incompressible flows, J. Comput. Phys. 229 (2010) 6300-6317.

[12] X.Y. Hu, B.C. Khoo, N.A. Adams, F.L. Huang, A conservative interface method for compressible flows, J. Comput. Phys. 219 (2006) 553-578.

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

[14] 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, J. Comput. Phys. 229 (2010) 1043-1076.

[15] 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.

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

[17] 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.

[18] 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).

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

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

[21] M.S. Liou, C.J. Steen Jr., A new ux splitting scheme, J. Comput. Phys. 107 (1993) 2339.

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

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