INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A simple second order cartesian scheme for compressible Euler flows

Yannick Gorsse — Angelo Iollo — Haysam Telib — Lisl Weynans —

N° 7773

Octobre 2011

Computational models and simulation

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--7773--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
BORDEAUX - SUD OUEST

A simple second order cartesian scheme for compressible Euler flows

Yannick Gorsse* , Angelo Iollo*, Haysam Telib† , Lisl Weynans*,

Theme : Computational models and simulation
Applied Mathematics, Computation and Simulation
Équipe-Projet MC2

Rapport de recherche n° 7773 — Octobre 2011 — 19 pages

Abstract: We present a finite-volume scheme for compressible Euler flows where the grid is cartesian and it does not fit to the body. The scheme, based on the definition of an ad hoc Riemann problem at solid boundaries, is simple to implement and it is formally second order accurate. Results show that pressure is locally and globally second accurate, whereas the accuracy of other variables is between 1 and 2. Examples of flow solutions in two and three dimension are shown.

Key-words: Immersed boundary, compressible fluids, second order scheme.

* Univ. Bordeaux, IMB, UMR 5251, F-33400 Talence, France. CNRS, IMB, UMR 5251, F-33400 Talence, France. INRIA, F-33400 Talence, France.

† Dipartimento di Ingegneria Aeronautica e Spaziale, Politecnico di Torino, and Optimad Engineering, Via Giacinto Collegno 18, 10143 Turin, Italy.

Centre de recherche INRIA Bordeaux – Sud Ouest
Domaine Universitaire - 351, cours de la Libération 33405 Talence Cedex
Téléphone : +33 5 40 00 69 00

A simple second order cartesian scheme for compressible Euler flows

Résumé : Nous présentons dans ce rapport un schéma volumes finis pour les équations d'Euler compressible sur maillage cartésien. Notre méthode, basée sur la définition d'un problme de Riemann approprié aux interfaces proches du solide, est simple à implémenter et est d'ordre 2. Les tests de convergence montrent que la pression est d'ordre 2 localement et globalement, tandis que l'ordre de convergence en norme L_∞ des autres variables se situe entre 1 et 2. Nous présentons des simulations numériques d'écoulements 2D et 3D.

Mots-clés : Frontière immergé, fluides compressibles, schéma d'ordre deux.

1 Introduction

The computation of flows in complex geometries is a crucial issue to perform realistic simulations of physical or biological applications like for instance biolocomotion (fish swimming or insect flight), turbomachines, windmills... To this end several class of methods exist. Here we are concerned with immersed boundary methods, i.e., integration schemes where the grid does not fit the geometry. These methods have been widely developed in the last 15 years, though the first methods were designed earlier (see for example [31], [4], [5]). The general idea behind immersed boundary methods is to take into account the boundary conditions by a modification of the equations to solve, either at the continuous level or at the discrete one, rather than by the use of an adapted mesh. The main advantages of using these approaches, compared to methods using body-conforming grids, are that they are easily parallelizable and allow the use of powerful line-iterative techniques. They also avoid to deal with grid generation and grid adaptation, a prohibitive task when the boundaries are moving.

A recent through review of immersed boundary methods is provided by Mittal and Iaccarino [26]. They distinguish two different kinds of immersed boundary methods. In the first category are methods that handle the presence of boundaries by adding a continuous forcing directly to the flows equations, like in the original immersed boundary method ([31]). The boundary is represented by lagrangian markers exerting a forcing in the momentum equations. Also the penalty method ([1],[2]) and its recent developments ([22],[7],[6]), belong to this category: the solid is modeled as a porous medium with vanishing porosity, typically leading to a singular volume force in the momentum equations.

The methods in the second category are usually refered as "cartesian grid methods", where the forcing accounting for the presence of boundaries is performed at the discrete level. The ghost-cell approach, inspired by the ghost-fluid method developed by Fedkiw ([14], [13]) for multiphase flows belongs to this category. The first developments ([23], [12]) were followed by many other extensions [38], [16], [25]. Ghost cells are cells in the solid with at least one neighbour in the fluid. The values on these ghost cells are extrapolated from the values in the fluid in order to impose the appropriate boundary conditions at the interface. The sub-mesh penalty method introduced by Sarthou et al. [32] is also related to this family. The immersed interface method is another approach, developed by Leveque and Li [20] and extended to flow problems in [21]. It is based on Taylor expansions of the solution on each side of the interface. The "cut-cell" approach belongs also to this class, and was mainly developed for compressible flows, see for example [30], [40], [24], [10], [17]. These methods preserve conservation properties near the boundary.

Parallel to this classification, the issue of the accuracy of the immersed boundary methods, especially near the immersed interface, is worth being raised. For example, methods such as the original immersed boundary method from Peskin, or the penalty method, are known to be order one accurate in space. For Navier-Stokes equations, either compressible or incompressible, several types of methods provide an order two accuracy in space, global as well as local (near the interface): Tseng and Ferziger [38], testing their ghost-cell method on an uniform flow past a cylinder, report an order two accuracy in the L^∞ norm in the domain and on the boundary, provided that the values of the ghost cells are computed with a quadratic extrapolation. Mittal et al. [25] and Ghias et al. [16], using a ghost-cell method, also obtained an order two accuracy for the same kind of test case. Ye et al [40], using a cut-cell method, reported an order two accuracy for the L^1 and L^2 norms of the error on the velocity in the whole domain for a two-dimensional Stokes flow. Kim et al. [18], with a finite volume approach, for decaying vortices obtained an order two accuracy for the L^∞ norm of the velocity error on the immersed boundary.

However, in the case of the compressible Euler equations, order two accuracy near the boundary seems more difficult to get. Pember et al. [30] proposed a cut cell method based on a Godunov algorithm and a correction step for cells at the boundary inspired by volume of fluid approaches, with a redistribution step to maintain conservativity. They tested the accuracy of their method on a Prantl-Meyer expansion wave. They reported for the entropy an order two of convergence for the L^1 norm on the whole domain, and an order one for the cells at the interface (and smaller convergence rates for the stagnation enthalpy). Forrer and Jeltsch [15], with a ghost-cell type method reported similar results on the same test case. Coirier and Powell [9] had similar convergence results too for the Ringleb flow, with another cut cell method. Colella et al. [10] obtained also the same kind of accuracy results for the case of a wave propagating in a straight circular channel, with another cut-cell method generalizing the work of Pember et al [30]. Hu et al. [17] devised a cut cell method based on a level-set description of the interface and a mixing procedure to merge small fluid cells with their neighbours. For a moving wall problem, they obtained an order two accuracy for the global error, and a slightly superior to one error on the boundary. Sjogreen et al. [37] presented a ghost-cell method based on a second order interpolation in the normal direction. They studied the conservativity properties of their scheme and showed a 1.6 order for mass loss. Chaudhuri et al. [8] coupled a 5th order WENO scheme to an immersed boundary method where the fluid variables at the ghost cells are obtained using a local quadratic interpolation. They showed second order accuracy for shock/obstacle interaction problems. Dadone and Grossman [11] proposed a ghost-cell method based on an isoentropic pressure gradient correction at the boundary and obtained an order two accuracy for the normal velocity on the surface of a spheroid.

At this stage let us notice that ghost-cell methods used for compressible non-viscous flows are in fact more sophisticated than the ones developed for Navier-Stokes equations. For example, Forrer and Jeltsch [15] computed integrals of piecewise constant functions on polygons corresponding to the reflection of the computational fluid cells along the boundary. Dadone and Grossman [11] used a corrected pressure value accounting for the curvature of the interface. In the context of cut-cell methods, the shape of the cut cells must be carefully identified. This task is non-trivial in two dimensions, and can become very tedious in three dimensions. Small cell effects can deteriorate the stability of the method, and strategies developed to cope with this problem increase the complexity of the programming ([4], [5] and [3]).

In this paper we present a simple globally second order scheme inspired by ghost-cell approaches to solve compressible inviscid flows. In the fluid domain, away from the boundary, we use a classical finite-volume method based on an approximate Riemann solver. At the cells located on the boundary, we solve an ad hoc Riemann problem taking into account the relevant boundary condition for the convective fluxes by an appropriate definition of the contact discontinuity speed. To avoid pressure oscillations near the boundary, we weight the boundary condition with a fluid extrapolation, as a function of the angle between the normal to the boundary and the cartesian mesh. These ideas can be adapted to reach higher order accuracy. However, here we limit the discussion to second order accuracy as our objective is to device a method that can easily be implemented in existing codes and that is suitable for massive parallelization.

The paper is organized as follows: in section 2 we describe the finite volume scheme that we use to solve the flow equations in the fluid domain, away from the interface. In section 3 the scheme at the boundary is detailed. In section 4 we present numerical tests to validate the expected order of convergence.

2 Resolution in the fluid domain

We briefly describe how we solve the flow equations in the fluid domain, that is, on grid cells containing only fluid and whose neighbours contain only fluid too. The method consists in a classical finite-volume scheme based on a MUSCL ([19]) reconstruction with slope limiters.

2.1 Governing equations

The compressible Euler equations are:

$$
\begin{aligned}
&\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} = 0\\
&\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + p\mathbf{I}) = 0\\
&\frac{\partial E}{\partial t} + \nabla \cdot ((E+p)\mathbf{u}) = 0
\end{aligned}
$$

where the symbols have usual meaning. For a perfect gas

$$
E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho \mathbf{u}^2 \ \ , p = \rho R T \text{ and } c = \sqrt{\gamma R T}
$$

2.2 Discretization in space

We focus for the moment on a two-dimensional setting. The extension to three dimensions of the interior domain scheme is straightforward. Let i and j be integers and consider the rectangular lattice generated by i and j, with spacing Δ_x and Δ_y in the x and y direction, respectively.

Let W be the conservative variables and $\mathcal{F}^x(W)$, $\mathcal{F}^y(W)$ the convective flux vector in the x and y direction, respectively. By averaging the governing equations over any cell of the rectangular lattice we have

$$
\frac{dW_{i\,j}}{dt} + \frac{1}{\Delta_x}(\mathcal{F}^x{}_{i+1/2\,j} - \mathcal{F}^x{}_{i-1/2\,j}) + \frac{1}{\Delta_y}(\mathcal{F}^y{}_{i\,j+1/2} - \mathcal{F}^y{}_{i\,j-1/2}) = 0
$$

where $W_{i\,j}$ is the average value of the conservative variables on the cell considered, $\mathcal{F}^x_{i+1/2\,j}$ the average numerical flux in the x direction taken on the right cell side, and similarly for the other sides.

The average numerical convective fluxes at cell interfaces are approximated here using the Osher numerical flux function [29], so for example

$$\mathcal{F}^x_{i+1/2\,j} \approx \mathcal{F}^x_- + \int_{W_-}^{W_+} A^-_x(W)dW \tag{1}$$

where W_- and W_+ are the values of W at the left and the right of the appropriate interface retrieved by a MUSCL ([19]) reconstruction of the primitive variables, $\mathcal{F}^x_- = \mathcal{F}^x(W_-)$, $A^-_x(W)$ is the negative part of $A_x(W) = \partial\mathcal{F}^x/\partial W$ and the integral is taken on a specific path assuming a locally isoentropic Riemann problem. The choice of a specific Riemann solver to compute the numerical flux function is inessential to the following developments.

2.3 Time integration

A second order Runge-Kutta scheme is used for the time integration. We denote by $\mathcal{W}$ the cell averages of the conservative unknowns at a given time stage. The right-hand side of the semi-discrete equations is a non-linear vector function of the conservative variable averages, $-\mathcal{F}(\mathcal{W})$. According to this notation, given $\mathcal{W}^n$, the second order Runge-Kutta scheme that we employ in the following is:

$$\begin{array}{l} \mathcal{W}^{(1)} = \mathcal{W}^n - \Delta t \mathcal{F}(\mathcal{W}^n) \\ \mathcal{W}^{n+1} = \mathcal{W}^n - \frac{\Delta t}{2}\left(\mathcal{F}(\mathcal{W}^n) + \mathcal{F}(\mathcal{W}^{(1)})\right) \end{array} \tag{2}$$

3 A second order impermeability condition

The canonical finite-volume scheme described in the previous section is modified as follows for fluid cells that have one or more neighbouring cells separated by an interface. The boundary condition on this interface derives from the impermeability assumption, i.e., given normal velocity to the boundary (zero for a steady wall, but non-zero for a moving wall). We are concerned with recovering second order accuracy on the impermeability condition, as for convection dominated flows on slender bodies, i.e. thin boundary layers, this is expected to significantly affect the computational results near the solid walls at high Reynolds numbers.

3.1 Level set method

In order to improve accuracy at the solid walls crossing the grid cells we need additional geometric information. This information, mainly the distance from the wall and the wall normal, is provided by the distance function. The level set method, introduced by Osher and Sethian [28], is used to implicitly represent the interface of solid in the computational domain. We refer the interested reader to [33], [34] and [27] for recent reviews of this method. The zero isoline of the level set function represents the boundary Σ of the immersed body. The level set function is defined by:

$$\varphi(x) = \begin{cases} dist_\Sigma(x) & \text{outside of the solid} \\ -dist_\Sigma(x) & \text{inside of the solid} \end{cases} \tag{3}$$

A useful property of the level set function is:

$$\mathbf{n}(x) = \nabla\varphi(x) \tag{4}$$

where $\mathbf{n}(x)$ is the outward normal vector of the isoline of φ passing on x. In particular, this allows to compute the values of the normal to the interface, represented by the isoline $\varphi = 0$. The computation of the distance function follows state of the art numerical techniques [35].

3.2 Euler equations in one dimension

To make the ideas clear, let us start from a simple one-dimensional configuration. The typical situation for a grid that does not fit the body is shown in Fig. 1. The plan is to modify the numerical flux at the cell interface nearest to the solid boundary, in order to impose the boundary condition at the actual fluid-solid interface location with a given accuracy. For a steady body, we impose $u_b = u(x_b) = 0$ at the boundary point x_b where $\varphi(x_b) = 0$.

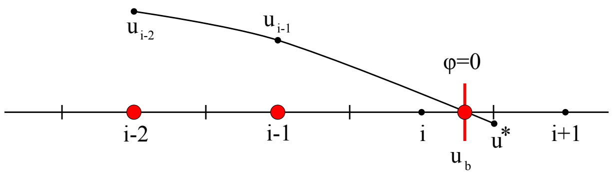


Figure 1: Mesh points and zero level set. The red segment represents $\varphi = 0$; hence the interface lies between the center of cell i (fluid) and the center of cell $i+1$ (solid). The numerical flux in $i+1/2$ has to be modified in order to account for the boundary conditions.

Let u^* be the contact discontinuity speed resulting from the solution of the Riemann problem defined at the interface between cell i and cell $i+1$. We define a fictitious fluid state to the right of $i+1/2$ such that u^* takes into account, at the desired degree of accuracy, the boundary condition $u_b = 0$ in x_b. In particular, using a second order polynomial with the collocation points marked by red dots on Fig. 1, we determine the value of u^* at $x_{i+1/2}$:

$$u^* = \mathcal{P}(x_{i+1/2})$$

where $\mathcal{P}(x) = a_2x^2 + a_1x + a_0$ and the coefficients are determined by

$$\begin{pmatrix} x_{i-2}^2 & x_{i-2} & 1 \\ x_{i-1}^2 & x_{i-1} & 1 \\ x_b^2 & x_b & 1 \end{pmatrix} \begin{pmatrix} a_2 \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} \mathbf{u}_{i-2} \\ \mathbf{u}_{i-1} \\ \mathbf{u}_b \end{pmatrix}$$

and where $\mathbf{u}_b$ is the velocity of the obstacle ($= 0$ for a steady body). The point x_i is not considered to avoid stability issues in the interpolation when the point where $\varphi = 0$ is close to x_i.

At $x_{i+1/2}$ the left fluid state of the Riemann problem $U_- = (u_-, p_-, c_-)$ is computed as usual with the MUSCL reconstruction. The right state is $U_+ =$

$(-u_- + 2u^*, p_-, c_-)$, so that u^* is the contact discontinuity speed of the resulting Riemann problem. The left and right states of the variables p and c are identical to enforce the continuity of these variables at the interface.

The scheme is non conservative at the numerical interface $x_{i+1/2}$. However, the loss of conservativity at this point is negligible compared to all the other points and the shocks are correctly resolved as shown in the following.

3.3 Euler equations in two dimensions

In two dimensions the flow equations are solved by computing independently the flux in each direction. Therefore, we apply in each direction the method employed in one dimension in order to accurately enforce the boundary condition. When the level set function changes sign between two cells, we modify the numerical fluxes at the interface between these cells. In particular, we compute a value of the contact discontinuity velocity at the cell side that will take into account the boundary condition at a point belonging to the solid-fluid interface.

The interface point naturally appearing in this context is the intersection between the interface ($\varphi = 0$) and the segment connecting the two cell centers concerned by the sign variation of φ (for example the points A or B on Fig. 2). For the numerical flux computation, a fictitious state is created for instance between the cells (i, j) and $(i+1, j)$ on Fig. 2. This fictitious state will be used to compute the numerical flux at this cell interface. However, the boundary condition that we have to impose now is $\mathbf{u}_A.\mathbf{n}_A = 0$, where $\mathbf{u}_A$ is the speed of the fluid at the boundary, and $\mathbf{n}_A$ the outward normal vector of the body.

When $\mathbf{n}_A$ is parallel to the cell side normal $\mathbf{n}_{cell}$ we recover the one-dimensional case. However, when the scalar product between the normal to the physical boundary and the normal to the mesh side is close to zero, see Fig. 3 for example, we assume that the boundary condition only weakly affects the numerical flux, as it would happen for a body fitted mesh. The state corresponding to a cell in the solid is therefore modified compared to the one-dimensional case in order to take into account this requirement thanks to a convex interpolation.

We compute as before the left state primitive variables u_-, p_-, c_- relative to the Riemann problem at the concerned numerical interface by a standard MUSCL reconstruction. According to Fig. 2 and considering the $(i+1/2, j)$-flux, the right state will be $U_+ = (-u_- + 2u_w, v_w, p_w, c_w)$, where:

- $\begin{pmatrix} u_w \\ v_w \\ p_w \\ c_w \end{pmatrix} = \alpha \begin{pmatrix} u^* \\ v^* \\ p_- \\ c_- \end{pmatrix} + (1-\alpha) \begin{pmatrix} u_f \\ v_f \\ p_f \\ c_f \end{pmatrix};$

- $\alpha = f(\mathbf{n}_A.\mathbf{n}_{cell})$, where $f(s) = s - \dfrac{1}{2\pi}\sin(-2\pi(s-0.5))$; see Fig. 3. Other choices for $f(s)$ do not significantly influence the results, as it is shown in Section 4;

- (u_f, v_f, p_f, c_f) is an extrapolated fluid state detailed hereafter.

and $U_- = (u_-, v_w, p_-, c_-)$.

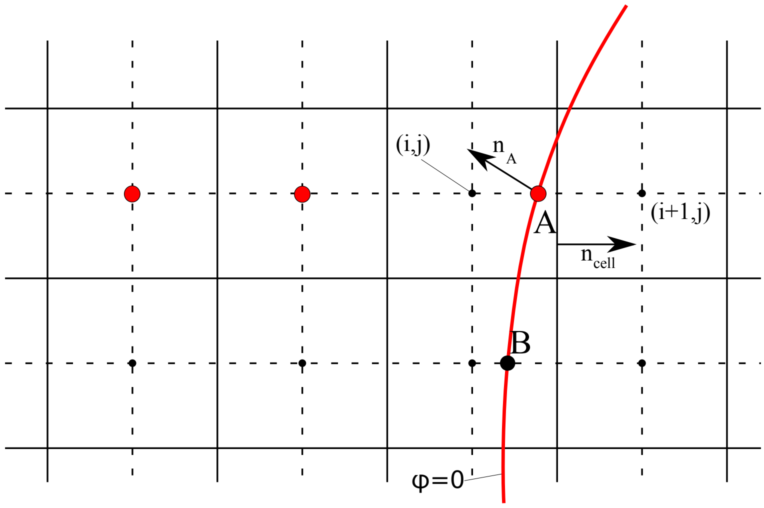


Figure 2: Example of geometric configuration at the interface. A is the interface point located between (i,j) and $(i+1,j)$. The flux on cell interface $(i+1/2,j)$ is modified to enforce the boundary condition on A.

The speeds u^* and v^* : The normal component of the contact discontinuity speed $\mathbf{u}^*$ is calculated as in the one-dimensional case. With reference to Fig. 2, the level set function changes sign between $x_{i,j}$ and $x_{i+1,j}$ at point A. Let the physical boundary normal point to the fluid side. Thanks to the level set function φ, the normal the the level set is defined at each cell center. If we assume that the directions of the level-set normal vectors $\mathbf{n}_{i,j}$ and $\mathbf{n}_{i+1,j}$ are parallel, we find that the distance between $x_{i,j}$ and A is

$$d = \frac{\Delta_x |\varphi_{i,j}|}{|\varphi_{i,j}| + |\varphi_{i+1,j}|} \tag{5}$$

and the normal vector $\mathbf{n}_A$ is computed by

$$\mathbf{n}_A = \mathbf{n}_{i,j} + \frac{d}{\Delta_x}\left(\mathbf{n}_{i+1,j} - \mathbf{n}_{i,j}\right) \tag{6}$$

where $\mathbf{n}_{i,j}$ is a fourth order centered finite-difference approximation of $\nabla\varphi$ at point (i,j). We determine the value of the contact discontinuity speed $\mathbf{u}^*$, relative to a Riemann problem defined in the direction normal to the cell side through $x_{i+1/2,j}$, consistent at second order accuracy with $\mathbf{u}_A \cdot \mathbf{n}_A = 0$ in A. The vector $\mathbf{u}^*$ is determined as follows:

$$\left.\begin{array}{l} \mathbf{u}^*.\mathbf{n}_A = u_n^* = \mathcal{P}_n\left(x_{i+1/2\,j}\right) \\ \\ \mathbf{u}^*.\tau_A = u_\tau^* = \mathcal{P}_\tau\left(x_{i+1/2\,j}\right) \end{array}\right\} \Rightarrow \mathbf{u}^* = \begin{pmatrix} u_n^* n_x + u_\tau^* \tau_x \\ u_n^* n_y + u_\tau^* \tau_y \end{pmatrix}$$

Where $\mathbf{n}_A = (n_x, n_y)^t$ and $\tau_A = (\tau_x, \tau_y)^t$ are respectively the normal and tangential vectors to the boundary at point A and $\mathcal{P}_n(x) = a_2 x^2 + a_1 x + a_0$. The coefficients of $\mathcal{P}_n(x)$ are determined as follows:

$$\begin{pmatrix} x_{i-2\,j}^2 & x_{i-2\,j} & 1 \\ x_{i-1\,j}^2 & x_{i-1\,j} & 1 \\ x_A^2 & x_A & 1 \end{pmatrix} \begin{pmatrix} a_2 \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} \mathbf{u}_{i-2\,j}.\mathbf{n}_A \\ \mathbf{u}_{i-1\,j}.\mathbf{n}_A \\ \mathbf{u}_A.\mathbf{n}_A \end{pmatrix}$$

Where $\mathbf{u}_A$ is the velocity of the obstacle ($= 0$ for a steady body). Similarly, $\mathcal{P}_\tau$ is determined by the values of the velocity in the direction τ_A at points (i, j), $(i-1, j)$ and $(i-2, j)$.

Extracted fluid state: (u_f, v_f, p_f, c_f) is computed as a linear extrapolation of the fluid variables to the point $x_{i+1/2\,j}$ using the variables and the slopes of the closest upstream fluid cell. For example, considering the velocity direction is west to east, the extrapolated variables are determined with the variables and slopes of the closest upwind fluid cell with respect to u (starting cell of the arrow on Fig. 3), in order to respect the domain of dependence of the entropy equation. For a $(i, j + 1/2)$-flux, the same procedure is applied considering the closest upwind fluid cell with respect to v.

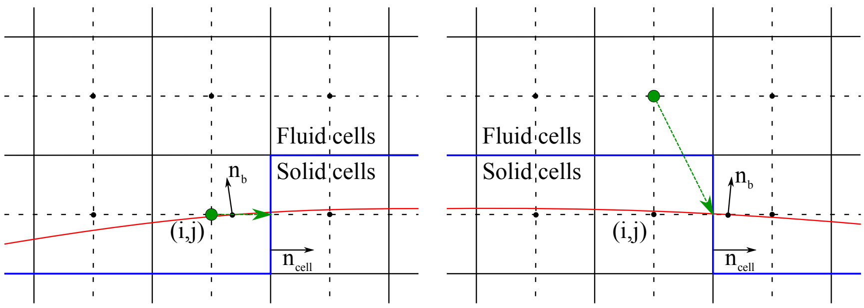


Figure 3: Cells and stencils used for the extrapolation state, considering $u_n^* > 0$.

3.4 Euler equations in 3D

The method described in the previous sections is easily extended in three dimensions. The interior cells, having all neighbours in the fluid, are treated by the classical finite-volume method described in Section 2. For the numerical flux at the mesh interface across two points where the distance function changes sign, exactly the same procedure described in two dimensions is applied. First of all the intersection point A between the solid boundary and the segment connecting the cell centers concerned is found. At this point the normal is determined thanks to the level set function and u_n^* is computed thanks to a polynomial extrapolation in the direction orthogonal to the numerical interface considered and taking into account the boundary condition at A.

Two vectors τ_1 and τ_2 are defined in the tangent plane through A. As before, the speeds in the tangential directions $u_{\tau_1}^*$ and $u_{\tau_2}^*$ are extrapolated from the fluid side.

The extrapolated fluid state is retrieved in the same as in two dimensions: we extrapolate starting from the closest upwind fluid cell. When several choices are possible, we consider the fluid cell whose velocity component in the extrapolation direction is higher.

Finally, the same correction based on the scalar product between the normal to the solid interface and the normal to the cell side is employed for the convex interpolation described before.

4 Numerical results

In this section, we will compare in some cases our results to two other existing methods that are comparable in terms of complexity of implementation. The first one is a simple symmetry technique, and the second one the ghost-cell CCST method, see [11], that relies on a local isoentropic flow model at the wall.

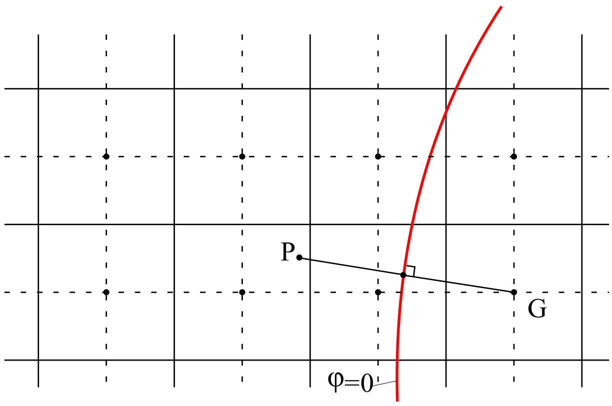


Figure 4: Illustration of the determination of ghost-cell variables.

For both methods, two rows of ghost points are computed, and slopes are calculated independently in each region (fluid or solid). With respect to Fig. 4, the first step is to determine the value of all the variables $U_P = (u_P^n, u_P^\tau, p_P, c_P)$ at point P, u_P^n and u_P^τ being respectively the velocity component normal and tangential to the wall at point P. U_P is determined with a bilinear interpolation of the four surrounding cell centers. Then, the value on the ghost point will be:

- $\begin{pmatrix} -u_P^n \\ u_P^\tau \\ p_P \\ c_P \end{pmatrix}$ for the symmetry technique
- $\begin{pmatrix} -u_P^n \\ u_P^\tau + \frac{2\gamma}{\gamma-1}\left(\frac{p_P}{\rho_P} - \frac{p_G}{\rho_G}\right) + u_P^n - u_G^n \\ p_P - \rho_P \frac{u_P^\tau}{R} \Delta n \\ c_P \left(\frac{p_G}{p_P}\right)^{\frac{1}{\gamma}} \end{pmatrix}$ for the ghost cell CCST technique called GCM in [11].

R is the wall curvature and Δn the distance between P and G.

4.1 Ringleb flow

The Ringleb flow refers to an exact solution of Euler equations. The solution is obtained with the hodograph method, see [36].

The exact solution is formulated in (θ, V) variables with $u = V\cos\theta$, $v = V\sin\theta$ and $V = \sqrt{u^2 + v^2}$. The stream function is given by:

$$\Psi = \frac{\sin\theta}{V}$$

The streamlines equations are:

$$x = \frac{1}{2\rho}\left(\frac{1}{V^2} - 2\Psi^2\right) + \frac{L}{2}, \quad y = \frac{\sin\theta\cos\theta}{\rho V^2}$$

with :

$$L = -\left(\frac{1}{2}\ln\frac{1+c}{1-c} - \frac{1}{c} - \frac{1}{3c^3} - \frac{1}{5c^5}\right) \quad , \quad c^2 = 1 - \frac{\gamma-1}{2}V^2 \quad , \quad \rho = c^5$$

In our test case, the computational domain is $[-0.5; -0.1] \times [-0.6; 0]$ and we numerically solve the flow between the streamlines $\Psi_1 = 0.8$ and $\Psi_2 = 0.9$. The inlet and outlet boundary condition are supersonic for $y = -0.6$ and $y = 0$ respectively. The convergence orders are calculated in L_2 and L_∞ norms on four different grids 32×48, 64×96, 128×192 and 256×384.

We start by investigating the effect on the accuracy order of the correction introduced to take into account the cases where the angle between the physical interface and the grid lines is small. We compare the solution obtained taking $\alpha = 1$, i.e., no upwind extrapolation, to those where α is computed as described in the previous section. As shown on table 4.1 and 2, the accuracy orders are significantly improved by the weighting procedure. Also the error amplitude, not shown, is lower than in the $\alpha = 1$ case.

	L_1	L_2	L_∞
u	2.04	1.68	1.28
v	1.97	1.6	1.13
p	2.0	2.02	1.97
c	1.95	1.58	1.03
s	1.9	1.49	1.08

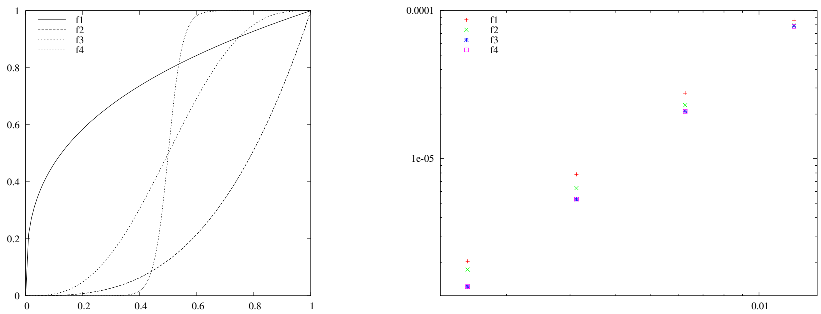
Table 1: Without extrapolated fluid state ($\alpha = 1$).

	L_1	L_2	L_∞
u	1.98	1.90	1.30
v	1.91	1.70	1.30
p	1.92	1.94	1.97
c	2.03	1.92	1.60
s	2.03	1.94	1.64

Table 2: With extrapolated fluid state.

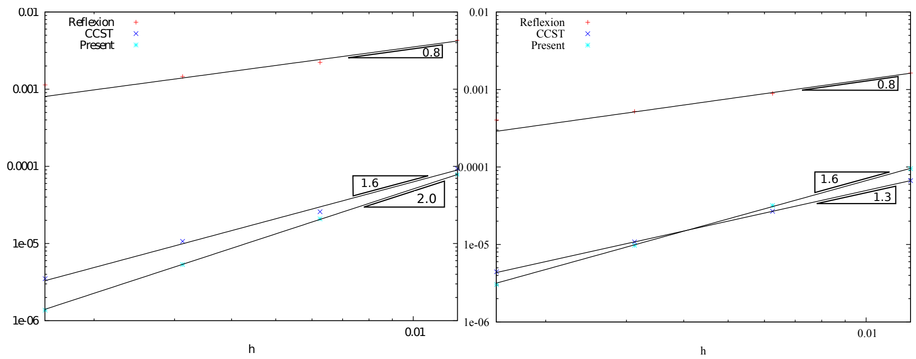
The particular choice of the function $f(\mathbf{n}_A.\mathbf{n}_{cell})$ that determines the value of the convex interpolation parameter α does not affect significantly the results. Computations have been performed with four different functions, see Fig. 5(a), and the accuracy orders are compared in Fig. 5(b). Symmetric functions lead to slightly more accurate results.

The results for the L_2 and L_∞ norm are now compared with the symmetry technique, and the ghost-cell CCST method [11] in Fig. 6 and Fig. 7. The symmetry technique requires a special interpolation if one or more cell centers surrounding point P are in the solid part, see Fig. 4. To avoid any bias, at grid points not belonging to the fluid we take $U_P = U_{ex}(x_P, y_P)$, with $U_{ex}(x_P, y_P)$ the exact solution of the Ringleb flow at point (x_P, y_P). For the CCST method, we proceed to a bilinear interpolation for all cases, the ghost points being a smooth extrapolation of the fluid region.



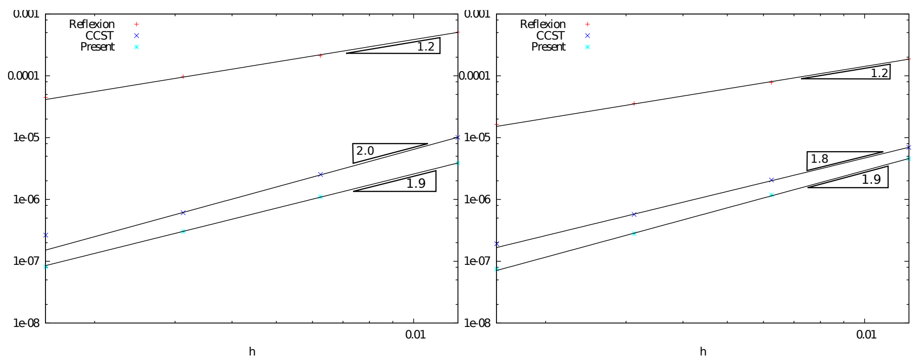
(a) $f(\mathbf{n}_b.\mathbf{n}_{cell})$. (b) Order of accuracy for pressure.

Figure 5: Comparison of the L_∞ error for pressure for different choices of $f(\mathbf{n}_b.\mathbf{n}_{cell})$.



(a) Pressure (b) Speed of sound

Figure 6: Comparison of the L_∞ accuracy of the classical symmetry technique, the ghost-cell CCST method [11] and the present method.



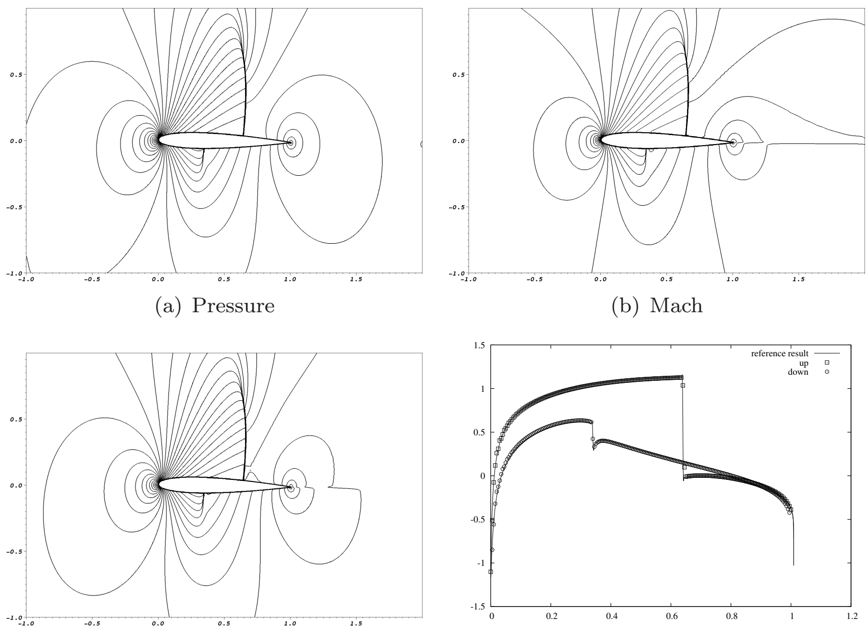
(a) Pressure (b) Speed of sound

Figure 7: Comparison of the L_2 accuracy of the classical symmetry technique, the ghost-cell CCST method [11] and the present method.

The overall results show that the classical symmetry scheme is at most first order accurate in the L_∞ norm, even if the ghost points are based on the reflection of the exact solution. In the same norm, the CCST method (which is based on a locally isoentropic flow extrapolation) is more accurate than the simple symmetry scheme. The accuracy of the present method is comparable to that of the CCST scheme in the L_2 norm and systematically more accurate in the L_∞ norm.

4.2 Transonic NACA0012 airfoil

The simulation of the flow over a transonic NACA0012 airfoil is performed. We choose a case with $M_\infty = 0.8$ and angle of attack $\alpha = 1.25°$ to test our method. This test case is classical. However, the precise solution is difficult to obtain even though the overall pattern of the flow may be recovered with a low grid resolution (see a recent and complete discussion in [39]). Here the computation is performed on a $30c \times 20c$ domain, where c is the airfoil chord and with $\Delta_x = \Delta_y = \frac{c}{100}$. A numerical zoom is then performed on $[-c, 2c] \times [-c, c]$ in order to achieve further resolution on the airfoil: the fully converged solution obtained on the $30c \times 20c$ domain is interpolated on the boundary points of the $[-c, 2c] \times [-c, c]$ domain where $\Delta_x = \Delta_y = \frac{c}{200}$. Pressure, mach, density contours and the pressure coefficient thus obtained are shown on Fig. 8.



(a) Pressure

(b) Mach

(c) Density

(d) Pressure coefficient

Figure 8: NACA0012, pressure, mach and density contours. Pressure coefficient on the airfoil surface compared to reference results.

Further resolution was needed for the lift and drag coefficients. The results obtained with $\Delta_x = \Delta_y = \frac{c}{400}$ in $[-c, 2c] \times [-c, c]$ are detailed and compared to [39] on table 4.2. The reference results are obtained with state of the art schemes on body fitted orthogonal meshes with over 16 million points in the whole domain and more than 4000 points on the airfoil. To be remarked that the last significant digits of the drag coefficient are strongly influenced by the particular closure of the NACA0012 at the trailing edge.

	Min [39]	Max [39]	Present
C_l	0.348226045	0.356208937	0.34951
C_d	0.022453440	0.022684938	0.02056

Table 3: Drag and lift coefficient.

4.3 Blunt body

This example is relative to a Mach 2 flow around a circular cylinder of radius 1 centered in $(0, 0)$. Because of the symmetry of the problem with respect to axis $y = 0$, the computational domain is restricted to $[-3\,,\,0] \times [0\,,\,6]$. A reference simulation, relative to a polar body fitted grid, is performed to compare to the results obtained on the cartesian mesh. For the reference simulation the computational domain is $r \in [1, 6]$ and $\theta \in [\frac{\pi}{2}, \pi]$. In Fig. 9 the comparison between the results obtained on the cartesian grid are contrasted to those obtained using the polar grid. The shock position and the isolines are basically the same for both methods and coincide on the horizontal axis as the grid is refined.

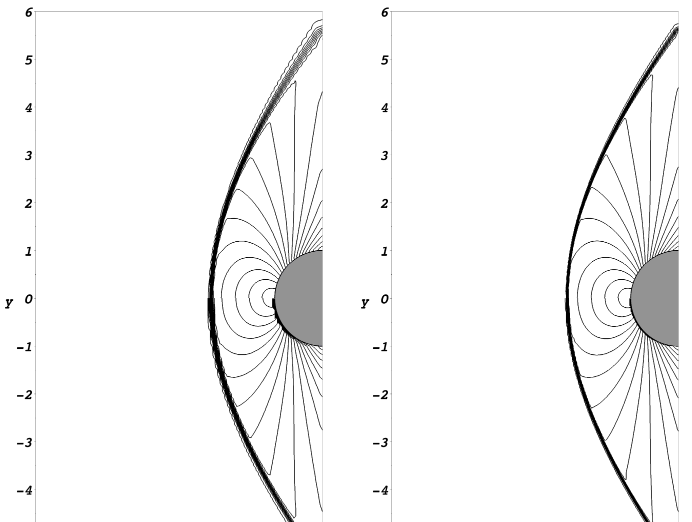


Figure 9: 20 iso-pressure lines between 0.6 and 4, $\Delta_x = \Delta_y = \frac{1}{20}$ on the left and $\frac{1}{40}$ on the right. The results with the body fitted mesh are on upper half of the figure, and with the cartesian mesh on the lower half.

4.4 Mach 10 shock over three spheres

The computation of a planar shock reflecting over three spheres is performed. The spheres are located at (0,0,0), (-0.75, 1.4,-1.4), (0,-2,0) with radii 1, 0.4, 0.3, respectively. The size of the domain is $[-2.5; 2.5]^3$. The numerical computation is performed on a 256^3 mesh. Four snapshots are shown on Fig. 10. The interactions of the bow shocks give rise to complicated flow structures in the wake of the spheres.

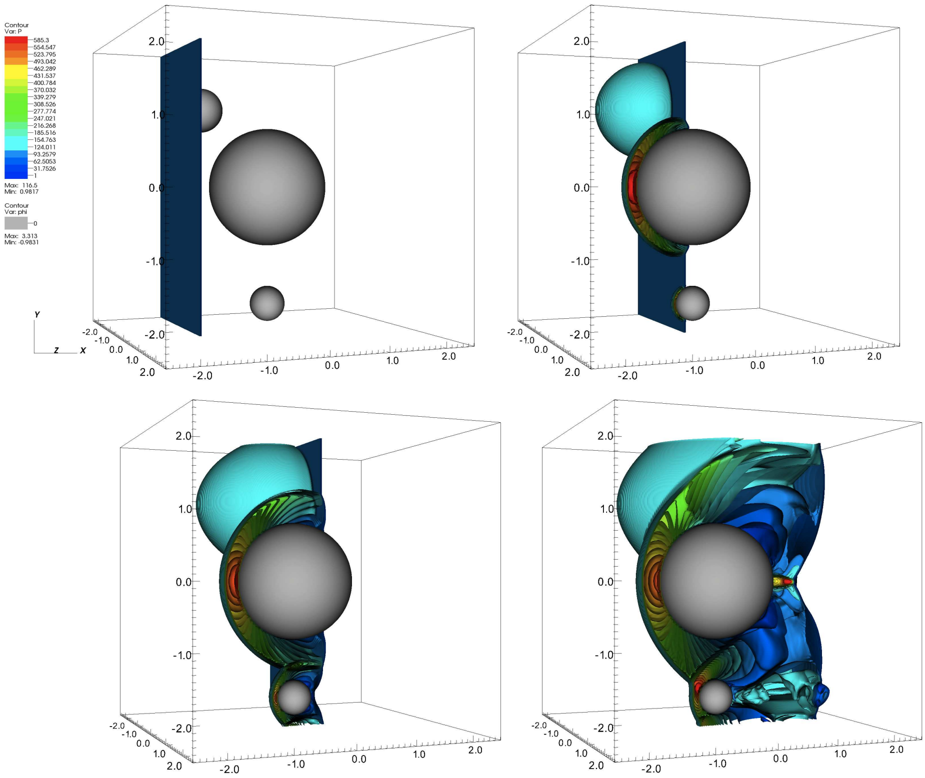


Figure 10: Mach 10 planar shock reflecting on a 3D sphere. 20 iso-pressure surfaces.

5 Conclusion

In this paper we have presented a cartesian method to solve compressible flows in complex domains with second order accuracy in smooth regions. Results show that pressure is locally and globally second accurate, whereas the accuracy of other variables is between 1 and 2. This method is based on a classical finite volume approach, but the values used to compute the fluxes at the cell interfaces near the solid boundary are determined solving a Riemann problem where the left and right states are modified so to satisfy the boundary conditions with second order accuracy. The method is therefore non-conservative at solid boundaries, but shocks are shown to be correctly captured. This method is particularly simple to implement, as it doesn't require any special cell reconstruction at the solid-wall interface. The extension to three-dimensional cases is natural as the same procedure at the boundary is repeated in each direction. Forthcoming work will concern the extension of the present approach to multi-physics problems.

References

[1] Angot, P., Bruneau, C.-H., and Fabrie, P. A penalization method to take into account obstacles in incompressible flows. Numer. Math. 81, 4 (1999), 497–520.

[2] Arquis, E., and Caltagirone, J. Sur les conditions hydrodynamiques au voisinage d'une interface milieu fluide-milieux poreux: application la convection naturelle. C.R. Acad. Sci. Paris II 299 (1984), 1–4.

[3] Berger, M., Helzel, C., and LeVeque, R. h-box methods for the approximation of hyperbolic conservation laws on irregular grids. SIAM J. on Numer. Analysis 41, 3 (2004), 893–918.

[4] Berger, M., and LeVeque, R. An adaptive cartesian mesh algorithm for the euler equations in arbitrary geometries, 1989.

[5] Berger, M., and LeVeque, R. Stable boundary conditions for cartesian grid calculations. Computing systems in Engineering 1, 2-4 (1990), 305–311.

[6] Bergmann, M., and Iollo, A. Modeling and simulation of fish-like swimming. J. Comput. Phys. 230 (2011), 329–348.

[7] Boiron, O., Chiavassa, G., and Donat, R. A high-resolution penalization method for large mach number ows in the presence of obstacles. Computers and Fluids 38, 3 (2009), 703–714.

[8] Chaudhuri, A., Hadjadj, A., and Chinnayya, A. On the use of immersed boundary methods for shock/obstacle interactions. Journal of Computational Physics 230, 5 (2011), 1731 – 1748.

[9] Coirier, W., and Powell, K. An accuracy assessment of cartesian mesh approaches for the euler equations. J. Comput. Phys. 117 (1995), 121–131.

[10] Colella, P., Graves, D., Keen, B., and Modiano, D. A cartesian grid embedded boundary method for hyperbolic conservation laws. J. Comput. Phys. 211, 1 (2006), 347–366.

[11] Dadone, A., and Grossman, B. Ghost-cell method for analysis of inviscid three dimensionnal flows on cartesian grids. Computers and Fluids 36 (2007), 1513–1528.

[12] Fadlun, E., Verzicco, R., Orlandi, P., and Mohd-Yusof, J. Combined immersed boundary finite difference methods for three dimensional complex flow simulations. J. Comput. Phys. 161 (2000), 35–60.

[13] Fedkiw, R. Coupling an eulerian fluid calculation to a lagrangian solid calculation with the ghost fluid method. J. Comput. Phys. 175 (2002), 200–224.

[14] Fedkiw, R., Aslam, T., Merriman, B., and Osher, S. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). J. Comput. Phys. 152 (1999), 457–492.

[15] Forrer, H., and Jeltsch, R. A high order boundary treatment for cartesian grid methods. J. Comput. Phys. 140 (1998), 259–277.

[16] Ghias, R., Mittal, R., and Dong, H. A sharp interface immersed boundary method for compressible viscous flows. J. Comput. Phys. 225 (2007), 528–553.

[17] Hu, X., Khoo, B., Adams, N., and Huang, F. A conservative interface method for compressible flows. J. Comput. Phys. 219 (2006), 553–578.

[18] Kim, J., Kim, D., and Choi, H. An immersed boundary finite volume method for simulations of flow in complex geometries. J. Comput. Phys. 171 (2001), 132–150.

[19] Leer, B. V. Towards the ultimate conservative difference scheme i. the quest of monotonicity. Lecture notes in Physics 18 (1972), 163.

[20] Leveque, R., and Li, Z. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. SIAM Numerical Analysis 31, 4 (1994), 1019–1044.

[21] Linnick, M., and Fasel, H. A high order interface method for simulating unsteady incompressible flows on irregular domains. J. Comput. Phys. 204 (2005), 157–192.

[22] Liu, Q., and Vasiliyev, O. A brinkman penalization method for compressible flows in complex geometries. J. Comput. Phys. 227 (2007), 946–966.

[23] Majumdar, S., Iaccarion, G., and Durbin, P. Rans solver with adaptive structured boundary non conforming grids, 2001.

[24] Miller, G., and Colella, P. A conservative three-dimensionnal eulerian method for coupled solid-fluid shock capturing. J. Comput. Phys. 183 (2002), 26–82.

[25] Mittal, R., Dong, H., Bozkurttas, M., Najjar, F., Vargas, A., and Von Loebbecke, A. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. J. Comput. Phys. 227 (2008), 4825—-4852.

[26] Mittal, R., and Iaccarino, G. Immersed boundary methods, 2005.

[27] Osher, S., and Fedkiw, R. Level Set Methods and Dynamic Implicit Surfaces. Springer, 2003.

[28] Osher, S., and Sethian, J. A. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton–jacobi formulations. J. Comput. Phys. 79, 12 (1988).

[29] Osher, S., and Soloman, F. Upwind difference schemes for hyperbolic systems of conservation laws. Math. Comp. 38, 158 (April 1982), 339–374.

[30] Pember, R., Bell, J., Colella, P., Crutchfield, W., and Welcome, M. An adaptive cartesian grid method for unsteady compressible flows in irregular regions. J. Comput. Phys. 120 (1995), 278–304.

[31] Peskin, C. The fluid dynamics of heart valves: experimental, theoretical and computational methods. Annu. Rev. Fluid Mech. 14 (1981), 235–259.

[32] Sarthou, A., Vincent, S., Angot, P., and Caltagirone, J. The sub-mesh penalty method. Finite Volumes for Complex Applications V (2008), 633–640.

[33] Sethian, J. A. Level Set Methods and Fast Marching Methods. Cambridge University Press, Cambridge, UK, 1999.

[34] Sethian, J. A. Evolution, implementation, and application of level set and fast marching methods for advancing fronts. J. Comput. Phys. 169 (2001), 503–555.

[35] shan Jiang, G., Shu, C.-W., and L, I. Efficient implementation of weighted eno schemes. J. Comput. Phys 126 (1995), 202–228.

[36] Shapiro, A. The Dynamics and Thermodynamics of Compressible Fluid Flow. Ronald Press, 1953.

[37] Sjogreen, B., and Petersson, N. A. A cartesian embedded boundary method for hyperbolic conservation laws. Communications in Computational Physics, 2, 6 (2007), 1199–1219.

[38] Tseng, F. A ghost cell immersed boundary method for flow in complex geometry. J. Comput. Phys. 192 (2003), 593 – 623.

[39] Vassberg, J. C., and Jameson, A. In pursuit of grid convergence for two-dimensional euler solutions. Journal of Aircraft 47, 4 (2010), 1152–1166.

[40] Ye, T., Mittal, R., Udaykumar, H., and Shyy, W. An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries. J. Comput. Phys. 156 (1999), 209–240.

Centre de recherche INRIA Bordeaux – Sud Ouest
Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex (France)

Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399