A second order finite-volume scheme on cartesian grids for Euler equations

Yannick Gorsse, Angelo Iollo, Lisl Weynans

A second order finite-volume scheme on cartesian grids for Euler equations

Y. Gorsse [a], A. Iollo [b], L. Weynans [b]

a. *CEA Cadarache*
b. *Univ. Bordeaux, IMB, UMR5251, F-33400, Talence, France*
CNRS, IMB, UMR5251, F-33400, Talence, France
INRIA, F-33400, Talence, France

Résumé :

Nous présentons un schéma de type volumes-finis d'ordre deux pour la résolution des équations d'Euler compressible dans des géométries complexes. La discrétisation est faite sur une grille cartésienne, qui en général ne coincide pas avec la géométrie considérée. Dans le domaine fluide, nous utilisons une méthode de volumes-finis classique. Sur les cellules situées près de la frontière avec le solide, nous résolvons un problème de Riemann ad hoc qui tient compte des conditions aux limites à imposer. Pour éviter les oscillations de pression près du solide, nous effectuons une pondération de la condition aux limites avec une extrapolation des valeurs du fluide, selon l'angle entre la normale au solide et celle à la cellule de la grille cartésienne. Le schéma est simple à implémenter et précis à l'ordre 2. Nous présentons une étude de la convergence de l'erreur, des comparaisons avec d'autres méthodes de la littérature, et des exemples d'écoulements en une, deux et trois dimensions.

Abstract :

We present a second-order finite-volume scheme for the resolution of Euler compressible equations in complex geometries. The discretization is performed on a Cartesian grid, which in general does fit with the considered geometry. In the fluid domain we use a classical finite-volume method. On the cells located near the boundary with the solid, we solve an ah hoc Riemann problem taking into account the boundary conditions that we want to impose. To avoid pressure oscillations near the solid we balance the boundary condition with an extrapolation of the fluid values, as a function of the angle between the normal to the solid and the normal to the cell. The scheme is straightforward to implement and second-order accurate. We present a convergence study, comparisons with other methods of the literature, and examples of flows simulations in one, two and three dimensions.

Mots clefs : **Euler equations ; complex geometries ; cartesian grids**

1 Introduction

The computation of flows in complex unsteady 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 [4], [1], [2]). 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 [8].
In this paper we present a simple globally second order scheme inspired by ghost cell approaches to solve compressible inviscid flows [7]. In the fluid domain, away from the boundary, we use a classical finite-volume method based on an approximate Riemann solver for the convective fluxes. 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 solid we balance the boundary condition with an extrapolation of the fluid values, as a function of the angle between the normal to the solid and the normal to the cell. Our objective is to device a method that can easily be implemented in existing codes and that is suitable for massive parallelization.
In section 2 we describe the finite volume scheme used in the fluid domain, away from the interface. In section 3 we introduce our method for solving the compressible Euler equations near the solid. Finally, in section 4 we present numerical tests to validate the expected order of convergence and to discuss performance compared to others immersed boundary or body fitted methods.

2 Resolution in the fluid domain

2.1 Governing equations

The compressible Euler equations are :

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} = 0 \tag{1}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + p\mathbf{n}) = 0 \tag{2}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot ((E+p)\mathbf{u}) = 0 \tag{3}$$

$$\tag{4}$$

where E denotes the total energy per unit volume. For a perfect gas

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho \mathbf{u}^2 \text{ and } p = \rho RT \tag{5}$$

2.2 Discretization

We focus on a two-dimensional setting. Let i and j be integers and consider the rectangular lattice generated by i and j, with spacing h_x and h_y in the x and y direction, respectively.

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

$$\frac{dW_{i\,j}}{dt} + \frac{1}{h_x}(\mathcal{F}^x{}_{i+1/2\,j} - \mathcal{F}^x{}_{i-1/2\,j}) + \frac{1}{h_y}(\mathcal{F}^y{}_{i\,j+1/2} - \mathcal{F}^y{}_{i\,j-1/2}) = 0 \tag{6}$$

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 flux in the x direction taken on the right cell side, and similarly for the other sides. The average convective fluxes at cell interfaces are approximated using the Osher numerical flux function. A second order Runge-Kutta scheme is used for the time integration.

3 A second order impermeability condition

For Euler equations, the boundary condition on the interface is the impermeability assumption, i.e., given normal velocity to the boundary (zero for a static wall, but non-zero for a moving solid). We are concerned with recovering second order accuracy on the impermeability condition.

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 [9], is used to implicitly represent the interface of solid in the computational domain. The zero isoline of the level set function represents the boundary Σ of the immersed body. The level set function is defined here by :

$$\varphi(x) = \begin{cases} dist_\Sigma(x) & \text{outside of the solid} \\ -dist_\Sigma(x) & \text{inside of the solid} \end{cases} \tag{7}$$

A useful property of this level set function is :

$$\mathbf{n}(x) = \nabla\varphi(x) \tag{8}$$

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 $\varphi = 0$.

3.2 The impermeability condition in one dimension

The typical situation in one dimension for a grid that does not fit the body is shown in Fig. 1. For a fixed body, we want to impose $u_b = u(x_b) = 0$ at the boundary point x_b where $\varphi(x_b) = 0$.

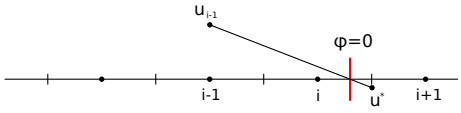


FIGURE 1 – Mesh near the solid. Red line is $\varphi = 0$. The interface lies between the center of cell i (fluid) and the center of cell $i+1$ (solid). The flux in $i+1/2$ has to be modified in order to account for the boundary conditions.

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 in x_b. In particular, the value of u^* at $x_{i+1/2}$ is determined by the following formula :

$$u^* = u_b + \left(\frac{1}{2} - d\right) s_b \,, \tag{9}$$

where $d = \dfrac{\varphi_i}{\Delta x}$, u_b is the velocity of the obstacle ($= 0$ for a steady body) and s_b is a slope defined by :

$$s_b = d\, s_1 + (1-d) s_2 \,, \tag{10}$$

where $s_1 = \dfrac{u_b - u_i}{d}$ and $s_2 = \dfrac{u_b - u_{i-1}}{1+d}$ are the slopes represented in Fig. 1. This formulation has been chosen to avoid stability issues in the slope computation when x_b is close to x_i since

$$s_b = u_b - u_i + \frac{1-d}{1+d}\left(u_b - u_{i-1}\right). \tag{11}$$

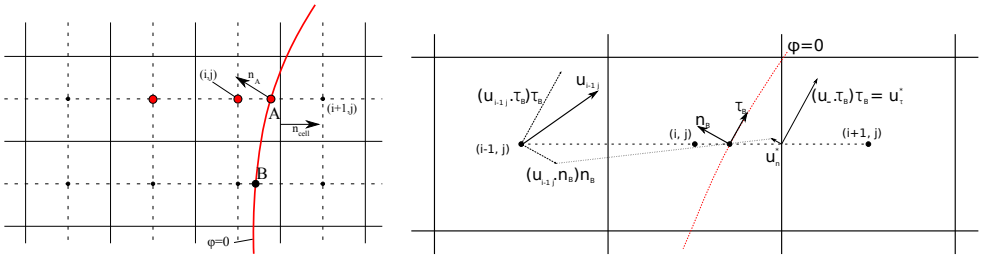
If a slope limiter is needed, the limited slope is defined by $s_b^l = \text{minmod}(s_b, s_3)$, where $s_3 = u_i - u_{i-1}$. An higher polynomial reconstruction to retrieve s_b does not significantly modify the convergence rates shown in the following.

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 in order to impose the correct wave reflection.

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 The impermeability condition in two dimensions

In two dimensions the flow equations are solved by computing independently the flux in each direction, so we want to apply in each direction the same kind of ideas as in one dimension. The interface points are the intersections between the interface ($\varphi = 0$) and the segment connecting the two cell centers concerned by the sign change (for example the points A, B and C on Fig. 2(a)). For the flux computation, a fictitious state is created for instance between the cells (i, j) and $(i+1, j)$ on Fig. 2(a). 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.



(a) Geometric configuration (b) Construction of vector

FIGURE 2 – Fig. 2(a) : 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. Fig. 2(b) : Graphical illustration of the construction of the $\mathbf{u}^*$ vector.

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, 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 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(b) and considering the $(i + 1/2, j)$-flux, the right state will be $U_+ = (-u_- + 2u_w, v_w, p_w, c_w)$, where :

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

– $\alpha = \mathbf{n}_A.\mathbf{n}_{cell}$;

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

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

The speeds u^* and v^* : The normal component of the contact discontinuity speed $\mathbf{u}^*$ is calculated as in the one-dimensional case. 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 :

$$\left.\begin{array}{l} \mathbf{u}^*.\mathbf{n}_A = u^*_n = \mathbf{u}_A.\mathbf{n}_A + \left(\frac{1}{2} - d\right) s^{\mathbf{n}}_A \\ \\ \mathbf{u}^*.\tau_A = u^*_\tau = \mathbf{u}_-.\tau_A \end{array}\right\} \Rightarrow \mathbf{u}^* = \begin{pmatrix} u^*_n n_x + u^*_\tau \tau_x \\ u^*_n n_y + u^*_\tau \tau_y \end{pmatrix}$$ Where $\mathbf{u}_A$ is the velocity of the obstacle (= 0 for a steady body), $\mathbf{n}_A = (n_x, n_y)^t$ and $\tau_A = (\tau_x, \tau_y)^t$ are respectively the normal and tangential vectors to the boundary at point A and the slope $s^{\mathbf{n}}_A$ is defined as in 1D :

$$s^{\mathbf{n}}_A = \mathbf{u}_A.\mathbf{n}_A - \mathbf{u}_i.\mathbf{n}_A + \frac{1-d}{1+d}\left(\mathbf{u}_A.\mathbf{n}_A - \mathbf{u}_{i-1}.\mathbf{n}_A\right). \tag{12}$$

If a slope limiter is needed, the limited slope is defined by $s^{\mathbf{n},l}_A = \text{minmod}(s^{\mathbf{n}}_A, s^{\mathbf{n}}_3)$, where $s^{\mathbf{n}}_3 = \mathbf{u}_i.\mathbf{n}_A - \mathbf{u}_{i-1}.\mathbf{n}_A$.

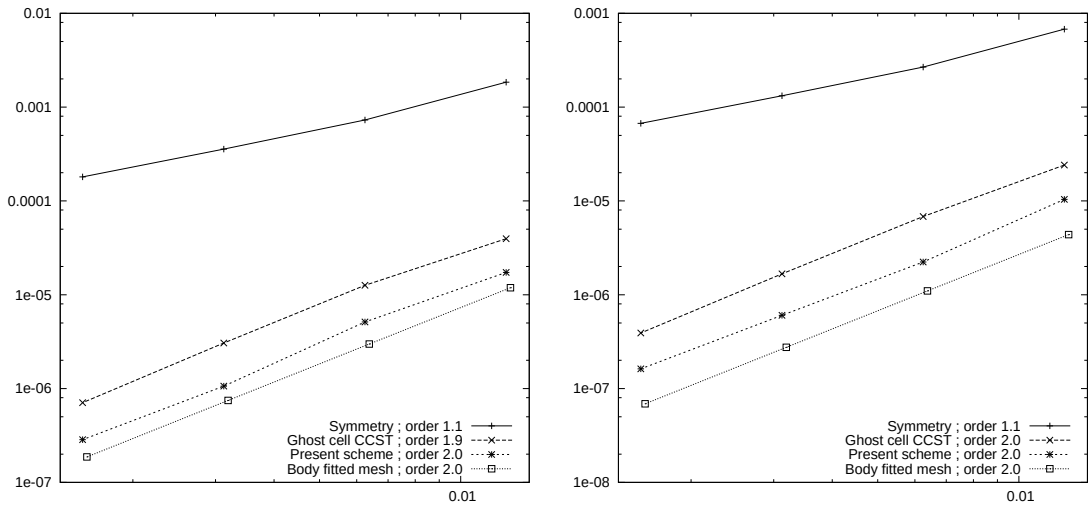
Extrapolated fluid state : (u_f, v_f, p_f, c_f) is computed as a linear extrapolation of the fluid variables to the point $x_{i+1/2\,j}$ using the variables and the slopes of the closest upstream fluid cell.

4 Numerical illustrations

The Ringleb flow

The Ringleb flow refers to an exact solution of Euler equations. The solution is obtained with the hodograph method. In our test case, the computational domain is $[-0.5;\ -0.1] \times [0;\ 0.6]$ 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 $y = 0$ and $y = 0.6$ respectively.

The convergence orders are calculated in L_2 norm on four different grids 32×48, 64×96, 128×192 and 256×384. The results for the L_2 norm of our method are compared to a simple symmetry technique, the ghost-cell CCST method [6], that relies on a local isoentropic flow model at the wall, and a standard finite-volume scheme case with a body-fitted meshs in Fig. 3.



(a) Pressure (b) Sound speed

FIGURE 3 – Comparison of the L_2 accuracy of the present scheme with several methods. The convergence orders are detailed in the legend.

The overall results show that the classical symmetry scheme is first order accurate in the L_2 norm. The other schemes have overall comparable accuracy, although the amplitude of the error is lower for the present scheme compared to the ghost-cell CCST method. For the same test case, Coirier and Powell [3] observed also a convergence order between one and two in the case of their own cartesian method.

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. 4. The interactions of the bow shocks give rise to complicated flow structures in the wake of the spheres.

5 Conclusion

We have presented a new cartesian method to solve compressible flows in complex domains. 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 so to satisfy the boundary conditions with a second order accuracy. Several numerical validations have been presented to assess its accuracy. This method is particularly simple to implement, as it doesn't require any special cell reconstruction at the solid-wall interface.

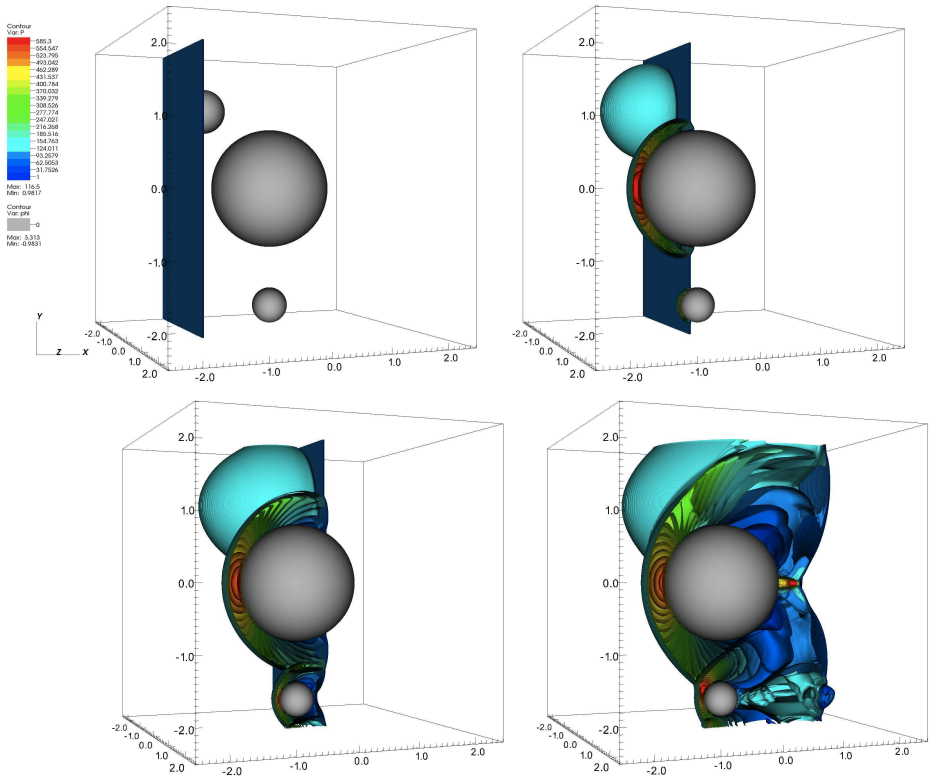


FIGURE 4 – Mach 10 planar shock reflecting on a 3D sphere. 20 isopressure surfaces.

Références

[1] Berger M.J. and LeVeque R.J. 1989 An adaptive cartesian mesh algorithm for the Euler equations in arbitrary geometries *AIAA-89-1930*

[2] Berger M.J. and LeVeque R.J. 1990 Stable boundary conditions for cartesian grid calculations *Computing systems in Engineering* **1** 305–311

[3] Coirier W. J. and Powell K. G. 1995 An accuracy assessment of cartesian mesh approaches for the Euler equations *J. Comput. Phys.* **117** 121–131

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

[5] Colella, Phillip and Graves, Daniel T. and Keen, Benjamin J. and Modiano, David 2006 A Cartesian grid embedded boundary method for hyperbolic conservation laws *J. Comput. Phys.* **211** 347–366

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

[7] Y. Gorsse, A. Iollo, H. Telib, L. Weynans 2012 A simple second order cartesian scheme for compressible Euler flows *J. Comput. Phys.* **231** 7780-7794

[8] Mittal R. and Iaccarino G. 2005 Immersed Boundary Methods *Annu. Rev. Fluid. Mech* 1–27

[9] Osher, S. and Sethian, J. A. 1988 Fronts propagating with curvature-dependent speed : Algorithms based on HamiltonÐJacobi formulations *J. Comput. Phys.* **79** 12–49