Journal of Computational Physics 231 (2012) 7780–7794

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

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

A simple second order cartesian scheme for compressible Euler flows

Yannick Gorsse [a,b,c], Angelo Iollo [a,b,c,*], Haysam Telib [d], Lisl Weynans [a,b,c]

[a] Univ. Bordeaux, IMB, UMR 5251, F-33400 Talence, France
[b] CNRS, IMB, UMR 5251, F-33400 Talence, France
[c] INRIA, F-33400 Talence, France
[d] Dipartimento di Ingegneria Aeronautica e Spaziale, Politecnico di Torino, and Optimad Engineering srl, Via Giacinto Collegno 18, 10143 Turin, Italy

ARTICLE INFO

Article history:
Received 18 October 2011
Received in revised form 19 March 2012
Accepted 5 July 2012
Available online 1 August 2012

Keywords:
Immersed boundary
Compressible flows

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. Error convergence rates with respect to several exact test cases are investigated and examples of flow solutions in one, two and three dimensions are presented.

© 2012 Elsevier Inc. All rights reserved.

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 [32,5,6]). 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 [27]. 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 [32]. The boundary is represented by lagrangian markers exerting a forcing in the momentum equations. Also the penalty method [2,3] and its recent developments [23,8,7], 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 [15,14] for multiphase flows belongs to this category. The first developments [24,13] were followed by many other extensions [40,17,26]. Ghost cells are cells in the solid with at least one neighbor 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.

* Corresponding author at: Univ. Bordeaux, IMB, UMR 5251, F-33400 Talence, France.
E-mail address: iollo@math.u-bordeaux1.fr (A. Iollo).

0021-9991/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2012.07.014

The sub-mesh penalty method introduced by Sarthou et al. [33] is also related to this family. The immersed interface method is another approach, developed by Leveque and Li [21] and extended to flow problems in [22]. 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 [31,42,25,11,18]. 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 [40], 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. [26] and Ghias et al. [17], using a ghost-cell method, also obtained an order two accuracy for the same kind of test case. Ye et al. [42], 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. [19], 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, it seems more difficult to get order two accuracy near the boundary. In practice, only a few cartesian methods obtain local second order accuracy in the case of Euler flows. In this sense, by investigating the semi-discrete approximation of hyperbolic and parabolic equations, Abarbanel et al. [1] show that if the order of the space approximation at the boundary is lower than that in the interior domain, the overall order of convergence in the hyperbolic case will be lower than that corresponding to the parabolic case. Pember et al. [31] 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 [16], with a ghost-cell type method reported similar results on the same test case. Coirier and Powell [10] had similar convergence results too for the Ringleb flow, with another cut cell method. Colella et al. [11] 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. [31]. Hu et al. [18] 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 neighbors. 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. [38] 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. [9] 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 accuracy between one and two for shock/obstacle interaction problems. Dadone and Grossman [12] 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 [16] computed integrals of piecewise constant functions on polygons corresponding to the reflection of the computational fluid cells along the boundary. Dadone and Grossman [12] used an isentropic model near the interface, and computed 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 [5,6,4].

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 neighbors contain only fluid too. The method consists in a classical finite-volume scheme based on a MUSCL [20] reconstruction with slope limiters.

2.1. Governing equations

The compressible Euler equations are:

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

where the symbols have usual meaning. For a perfect gas

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

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

$$\frac{dW_{ij}}{dt} + \frac{1}{\Delta_x}(\mathcal{F}^x_{i+1/2j} - \mathcal{F}^x_{i-1/2j}) + \frac{1}{\Delta_y}(\mathcal{F}^y_{ij+1/2} - \mathcal{F}^y_{ij-1/2}) = 0$$

where W_{ij} is the average value of the conservative variables on the cell considered, $\mathcal{F}^x_{i+1/2j}$ 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 [30], so for example

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

where W_- and W_+ are the values of W at the left and the right of the appropriate interface retrieved by a MUSCL [20] 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:

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

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 neighboring 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 [29], is used to implicitly represent the interface of solid in the computational domain. We refer the interested reader to [34,35,28] 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 [36].

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 $\varphi(x_b) = 0$.

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:

$$u^* = u_b + \left(\frac{1}{2} - d\right)s_b, \tag{5}$$

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

$$s_b = ds_1 + (1-d)s_2, \tag{6}$$

where $s_1 = \frac{u_b - u_i}{d}$ and $s_2 = \frac{u_b - u_{i-1}}{1+d}$ 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

$$s_b = u_b - u_i + \frac{1-d}{1+d}(u_b - u_{i-1}). \tag{7}$$

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. 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

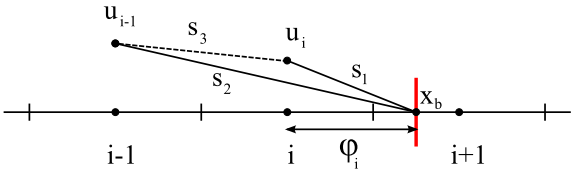


Fig. 1. Mesh points and zero level set. The red segment represents $\varphi = 0$; hence the interface lies between the center of cell i (fluid) and the center of cell $i+1$ (solid). Slopes s_1, s_2 and s_3 are used to compute the velocity at $i+1/2$. Slope s_3 is only used when a slope limiter is applied.

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:

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

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

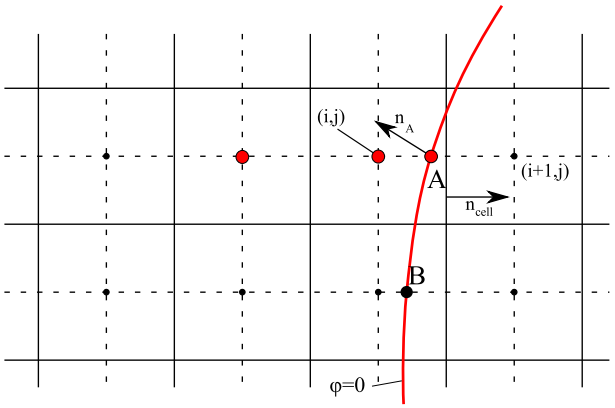


Fig. 2. Example of geometric configuration at the interface. A is the interface point located between (i,j) and $(i+1,j)$. The flux on cell interface $(i+1/2,j)$ is modified to enforce the boundary condition on A.

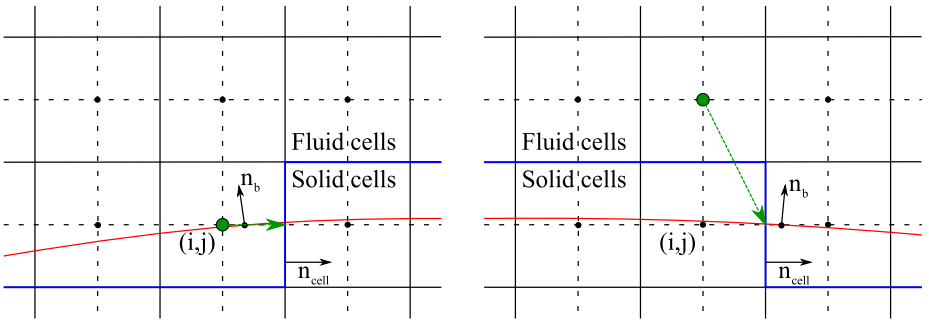


Fig. 3. Cells and stencils used for the extrapolation state, considering $u_n^* > 0$.

boundary normal point to the fluid side. Thanks to the level set function φ, the normal of 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 normalized distance between $x_{i,j}$ and A is

$$d = \frac{|\varphi_{i,j}|}{|\varphi_{i,j}| + |\varphi_{i+1\,j}|} \tag{8}$$

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

$$\mathbf{n}_A = \mathbf{n}_{i,j} + d(\mathbf{n}_{i+1\,j} - \mathbf{n}_{i,j}) \tag{9}$$

where $\mathbf{n}_{i,j}$ is a fourth order centered finite-difference approximation of $\nabla\varphi$ at point (i,j), or the exact normal if the geometry is analytically known. 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:

$$\left.\begin{array}{l} \mathbf{u}^*.\mathbf{n}_A = u_n^* = \mathbf{u}_A.\mathbf{n}_A + (\frac{1}{2} - d)s_A^{\mathbf{n}} \\ \mathbf{u}^*.\tau_A = u_\tau^* = \mathbf{u}_-.\tau_A \end{array}\right\} \Rightarrow \mathbf{u}^* = \begin{pmatrix} u_n^* n_x + u_\tau^* \tau_x \\ u_n^* n_y + u_\tau^* \tau_y \end{pmatrix}$$

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 $s_A^{\mathbf{n}}$ is defined as in 1D:

$$s_A^{\mathbf{n}} = \mathbf{u}_A.\mathbf{n}_A - \mathbf{u}_i.\mathbf{n}_A + \frac{1-d}{1+d}(\mathbf{u}_A.\mathbf{n}_A - \mathbf{u}_{i-1}.\mathbf{n}_A). \tag{10}$$

If a slope limiter is needed, the limited slope is defined by $s_A^{\mathbf{n},l} = \text{minmod}(s_A^{\mathbf{n}}, s_3^{\mathbf{n}})$, where $s_3^{\mathbf{n}} = \mathbf{u}_i.\mathbf{n}_A - \mathbf{u}_{i-1}.\mathbf{n}_A$.

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

3.4. Euler equations in 3D

The method described in the previous sections is easily extended in three dimensions. The interior cells, having all neighbors 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

We present a set of numerical illustrations of increasing complexity ranging from one-dimensional to non-symmetrical three-dimensional configurations. In particular, we compute convergence rates to reference solutions of well known test cases. We define the L_p error as $\left(\sum_i^N |e_i|^p / N\right)^{1/p}$ where e_i is the error relative to the i-th cell and N is the number of computational cells; the L_∞ error is defined as $\max(|e_i|)$ over the grid. The convergence rates are computed as the linear regression of the error data.

4.1. Analysis of the accuracy in 1D

In the following, the accuracy of the method is evaluated on three test cases involving a reflection on a solid wall, as a function of the relative position of the reflecting wall and the computational boundary. The test cases concern a rarefaction wave, a continuous pulse and a shock wave. The exact solutions are represented on Fig. 4. We assume that the wall is located

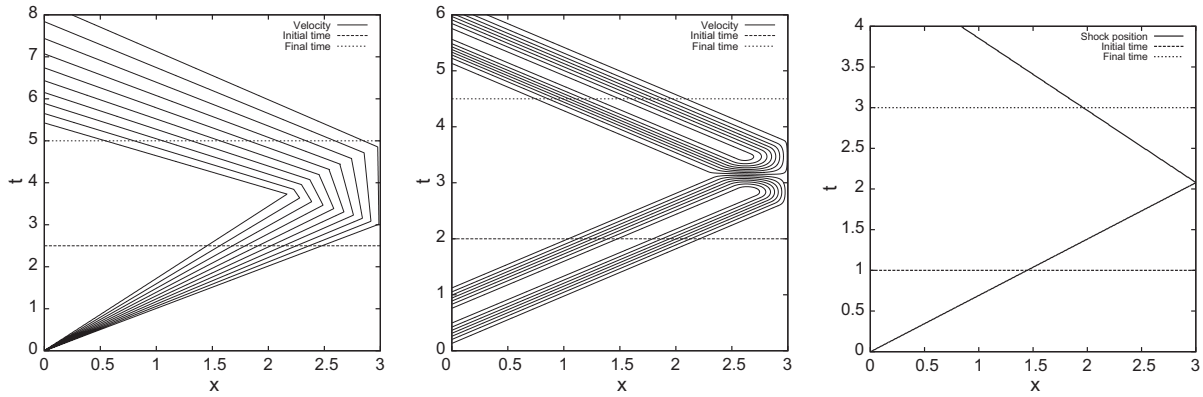


Fig. 4. (x,t) Diagrams of exact solutions in terms of velocity isocontours for the three test cases.

at $x_b = x_i + \psi(x_{i+1} - x_i),\ \psi \in [0,1]$. The accuracy of the present method is compared with the case of a reflexion on a solid wall exactly located at $i + 1/2$ ($\psi = 1/2$), where the boundary condition can be imposed without approximations.

The exact solution for the reflection of the rarefaction wave is obtained using Riemann invariants. The velocity to the left of the rarefaction is -0.35. To the right, where the gas is steady, the pressure is 1 and the speed of sound is 1. Concerning the continuous pulse, the velocity inlet boundary condition is a time-dependent portion of a sinusoid of amplitude 0.01. The inlet pressure is 1. The reference solution is computed using 10000 discretization points, with the wall located at a grid interface ($\psi = 1/2$). The exact solution for the reflection of the shock is calculated with the Rankine–Hugoniot conditions and an exact Riemann problem at the wall. The shockwave has a speed corresponding to a Mach number of 1.22. The computations are initialized at the time corresponding to initial time in the figures and the errors are calculated at the time corresponding to final time, with respect to the exact or reference solutions (see Fig. 4).

The error in several norms versus ψ is computed on 128 grid points and is represented on Fig. 5. On the plots we compare the error of our scheme with the error of a standard scheme for which $\psi = 1/2$. This corresponds to exactly imposing the impermeability boundary condition in the computation of the numerical flux. We denote with "optimal" the error corresponding to this ideal situation.

Overall, for the three test cases, the errors in all norms are close to the optimal one. Let us first consider the continuous solution cases. We opted for a numerical set up where the relative position of the initial condition and the wall is fixed. We remark that for increasing ψ the error slightly increases in L_1 and L_2 norms. This is coherent with the fact that we compute a boundary condition using a larger distance for extrapolation. The L_∞ norm is less regular because the grid point of the largest error may vary with ψ. It is however, on average, of the order of the optimal error. The case $\psi = 1$ corresponds to the case $\psi = 0$ and therefore the errors are identical.

For the shock case, the initial position of the discontinuity can be determined at most with an accuracy of the size of a grid cell at the discrete level. For this reason we opted for a wall location that varies with respect to the shock position as a function of ψ. Therefore the numerical solution corresponding to $\psi = 0$ does not exactly match to that obtained for $\psi = 1$. The error norms can be discontinuous for the same reason: negligible differences in the shock position lead to finite errors as function of the relative position of the exact solution, the approximate solution and the discretization points. This is of course particularly evident in the L_∞ norm.

4.2. Ringleb flow

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 [12], that relies on a local isoentropic flow model at the wall.

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. 6, 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

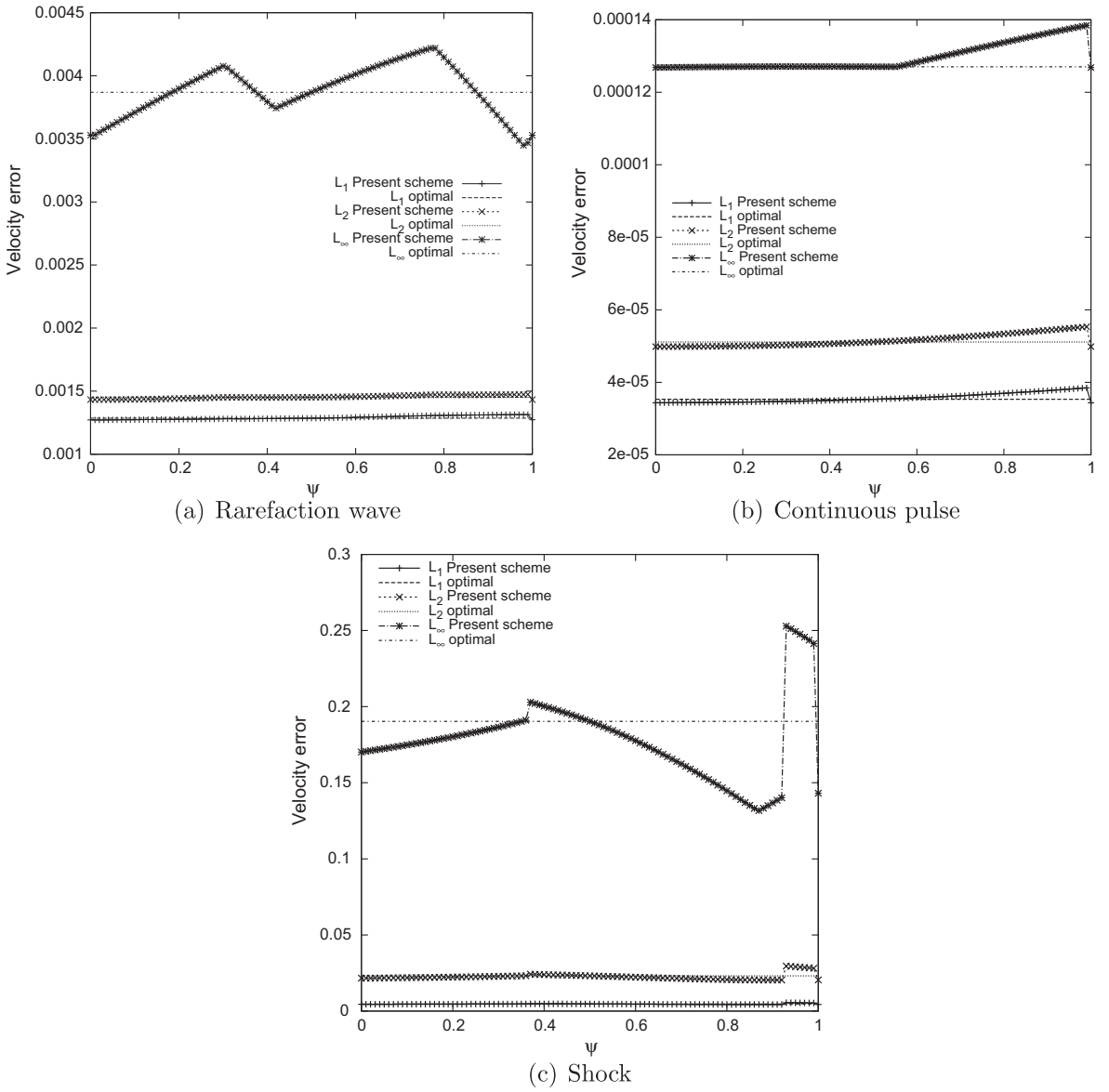


Fig. 5. Present scheme error and optimal error.

- $> \begin{pmatrix} -u_P^n \\ u_P^\tau + \frac{2\gamma}{\gamma-1}\left(\frac{p_P}{\rho_P} - \frac{p_G}{\rho_G}\right) + u_P^n - u_G^n \\ p_P - \rho_P \frac{u_P^\tau}{R}\Delta n \\ c_P \left(\frac{p_G}{p_P}\right)^{\frac{1}{\gamma}} \end{pmatrix}$ for the ghost cell CCST technique called GCM in [12].

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

The Ringleb flow refers to an exact solution of Euler equations. The solution is obtained with the hodograph method, see [37].

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:

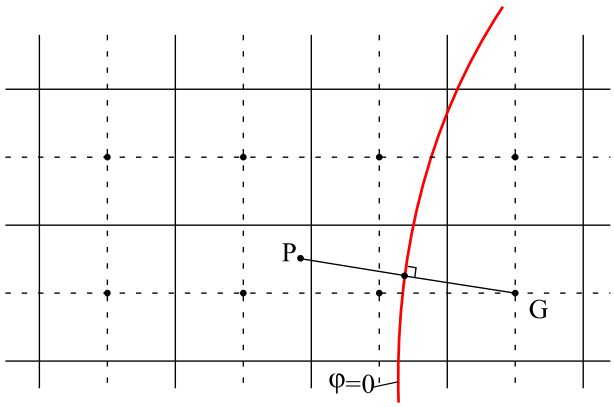


Fig. 6. Illustration of the determination of ghost-cell variables.

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

with:

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

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 and L_∞ norms on four different grids $32\times48, 64\times96, 128\times192$ and 256×384. We also perform a similar convergence study using a sequence of body fitted meshes and the same finite-volume scheme in the interior domain. The meshes are composed of streamlines and potential lines and an example is represented on Fig. 7.

The results for the L_2 and L_∞ norm are compared to the symmetry technique, the ghost-cell CCST method [12] and the case with body-fitted mesh and standard wall boundary conditions in Fig. 8 and Fig. 9. The symmetry technique requires

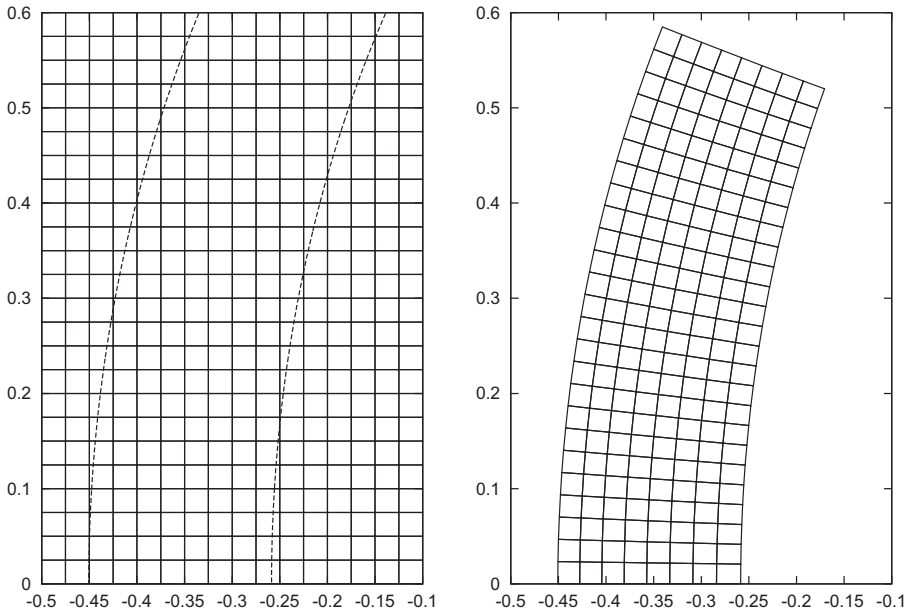


Fig. 7. Comparison between the cartesian mesh with the isoline $\varphi = 0$ and the body fitted mesh.

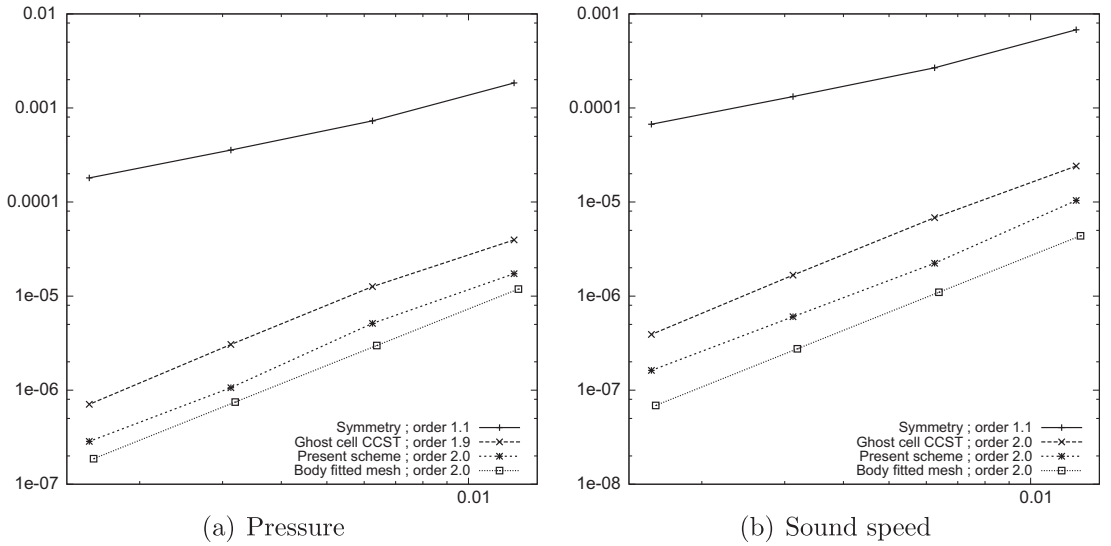


Fig. 8. Comparison of the L_2 accuracy of the present scheme with several methods. The convergence orders are detailed in the legend.

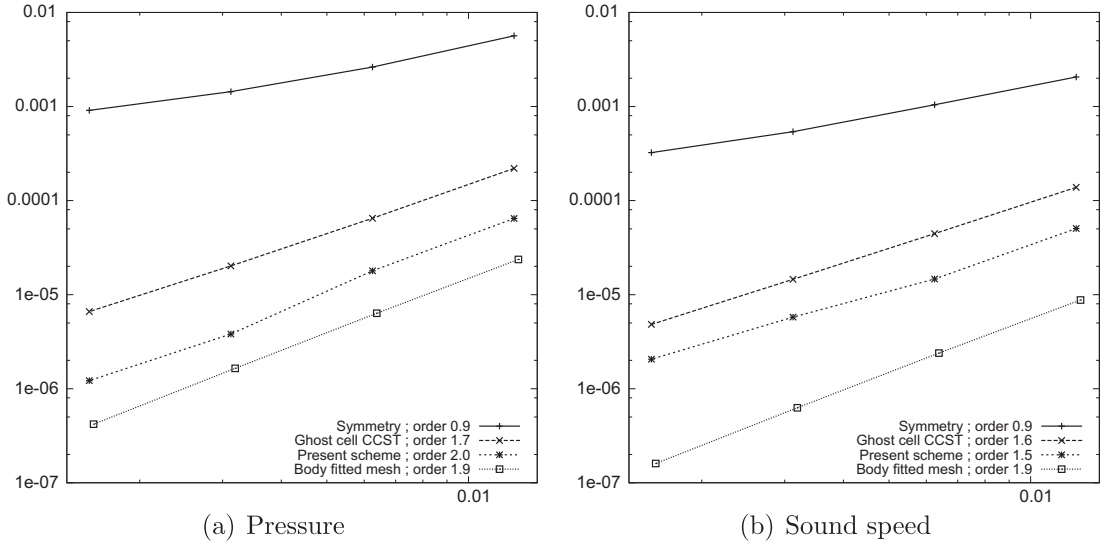


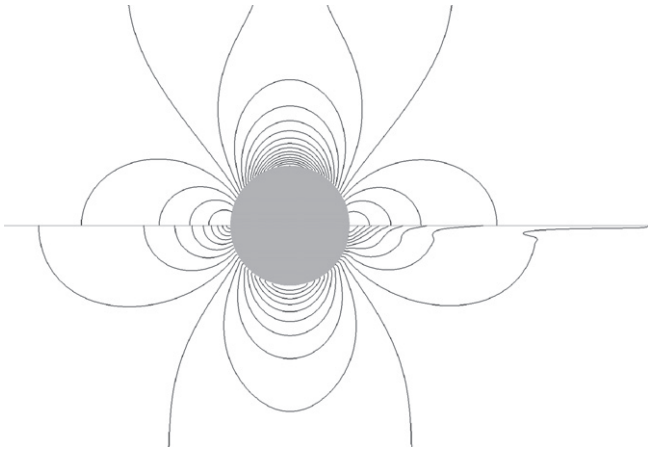
Fig. 9. Comparison of the L_∞ accuracy of the present scheme with several methods. The convergence orders are detailed in the legend.

a special interpolation if one or more cell centers surrounding point P are in the solid part, see Fig. 6. 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 ghost cell CCST method, we proceed to a bilinear interpolation for all cases, the ghost points being a smooth extrapolation of the fluid region.

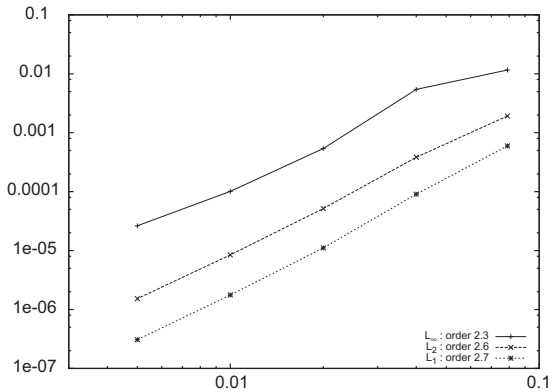
The overall results show that the classical symmetry scheme is first order accurate in the L_∞ norm, even if the ghost points are based on the reflection of the exact solution. The other schemes have overall comparable accuracy, although the amplitude of the error is lower for the present scheme compared to the ghost-cell CCST method.

4.3. Subsonic circular cylinder

We consider a circular cylinder of radius 1, centered at $(0, 0)$, in a computational domain of $[-3, 3] \times [0, 2]$. The boundary conditions at the left and right borders correspond to an unperturbed flow at Mach 0.2; at the bottom and top boundaries we impose symmetry conditions. The finest grid is 1200×400. To compute the convergence rates, we have successively reduced by half the grid points in each space direction. The pressure and Mach number fields obtained on the finest grid are shown in Fig. 10(a). For pressure we plot 29 isocontours equally distributed in $[0.89, 1.03]$. The Mach number isocontours are 19 and



(a) Upper half plane: pressure; lower: mach



(b) Entropy convergence rates

Fig. 10. Flow field and convergence rates for entropy.

they are equally distributed in $[0., 0.45]$. The pressure field is almost perfectly symmetric, whereas the Mach field shows the usual entropy layer. The convergence rates of the entropy error are presented in Fig. 10(b). They are even higher than expected thanks to the fact that the solution is approximately linear in space before and after the cylinder.

4.4. 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 $\alpha = 1.25^\circ$ 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 [41]). 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. 11.

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 [41] on Table 1. 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.

4.5. 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

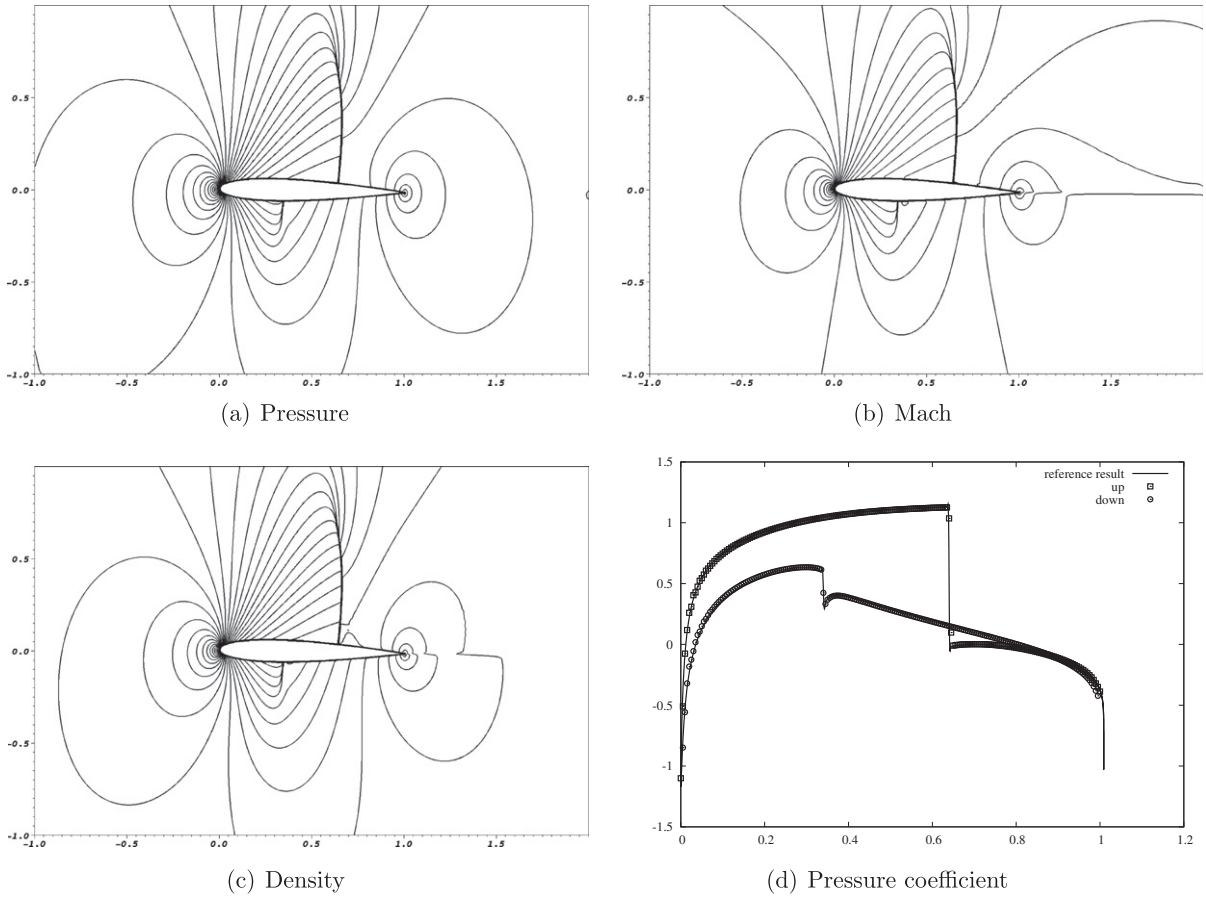


Fig. 11. NACA0012, pressure, mach and density contours. Pressure coefficient on the airfoil surface compared to reference results.

Table 1
Drag and lift coefficient.

	Min [41]	Max [41]	Present
C_l	0.348226045	0.356208937	0.34951
C_d	0.022453440	0.022684938	0.02056

simulation the computational domain is $r \in [1,6]$ and $\theta \in [\frac{\pi}{2}, \pi]$. In Fig. 12 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.

4.6. Supersonic wedge

A more complicated shock flow pattern can be investigated considering a supersonic wedge. For given upstream Mach number, and below a critical wedge angle, a straight oblique shock develops at the wedge corner. In this configuration there exist a supersonic uniform flow solution downstream the shock. Using the present scheme with a 300×200 grid, the upstream Mach set at 4 and a 20° wedge, we computed a shock angle of 32.49, whereas the exact value is 32.46.

Above a maximum flow deflection angle the oblique shock wave is no longer attached to the corner and is replaced by a detached bow shock. We consider a steady 25° wedge with an initial datum that consists in two discontinuous states connected through the Rankine–Huginot conditions for a shock moving with Mach = 1.7. The grid is 1200×800. The discontinuity is placed ahead of the wedge angle. The speed of the shock and the wedge deflection are such that this configuration corresponds to a super-critical case. In Fig. 13 we present the Mach and pressure fields. For pressure we plot 40 isocontours equally distributed in $[0.97, 4.82]$. The Mach number isocontours are 40 and they are equally distributed in $[0., 0.91]$, the time

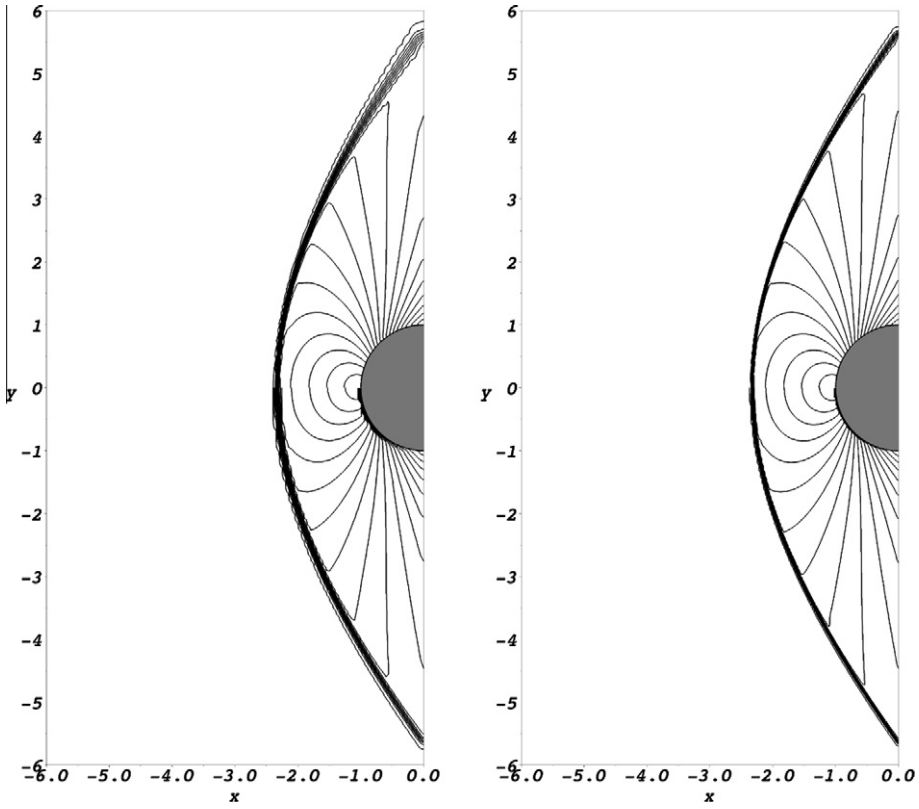


Fig. 12. Twenty pressure isolines between 0.6 and 4, $\Delta_x = \Delta_y = \frac{1}{20}$ on the left and $\frac{1}{40}$ on the right. The results with the body fitted mesh are on upper half of the figure, and with the cartesian mesh on the lower half.

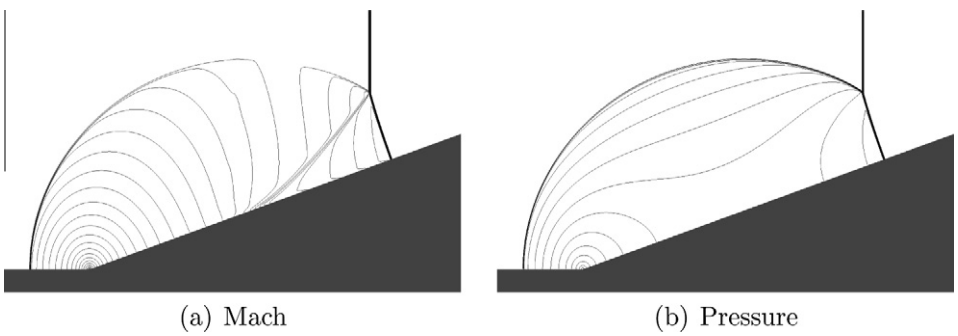


Fig. 13. Flow field past a super-critical wedge.

is 0.318. In the Mach field we remark a contact discontinuity resulting from the bow shock interaction with the shock to the right. The pressure isocontours do not exhibit any perturbation at the contact discontinutity. These results are in good accordance with the literature (see for example [39, Fig. 17.9, p. 590]).

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

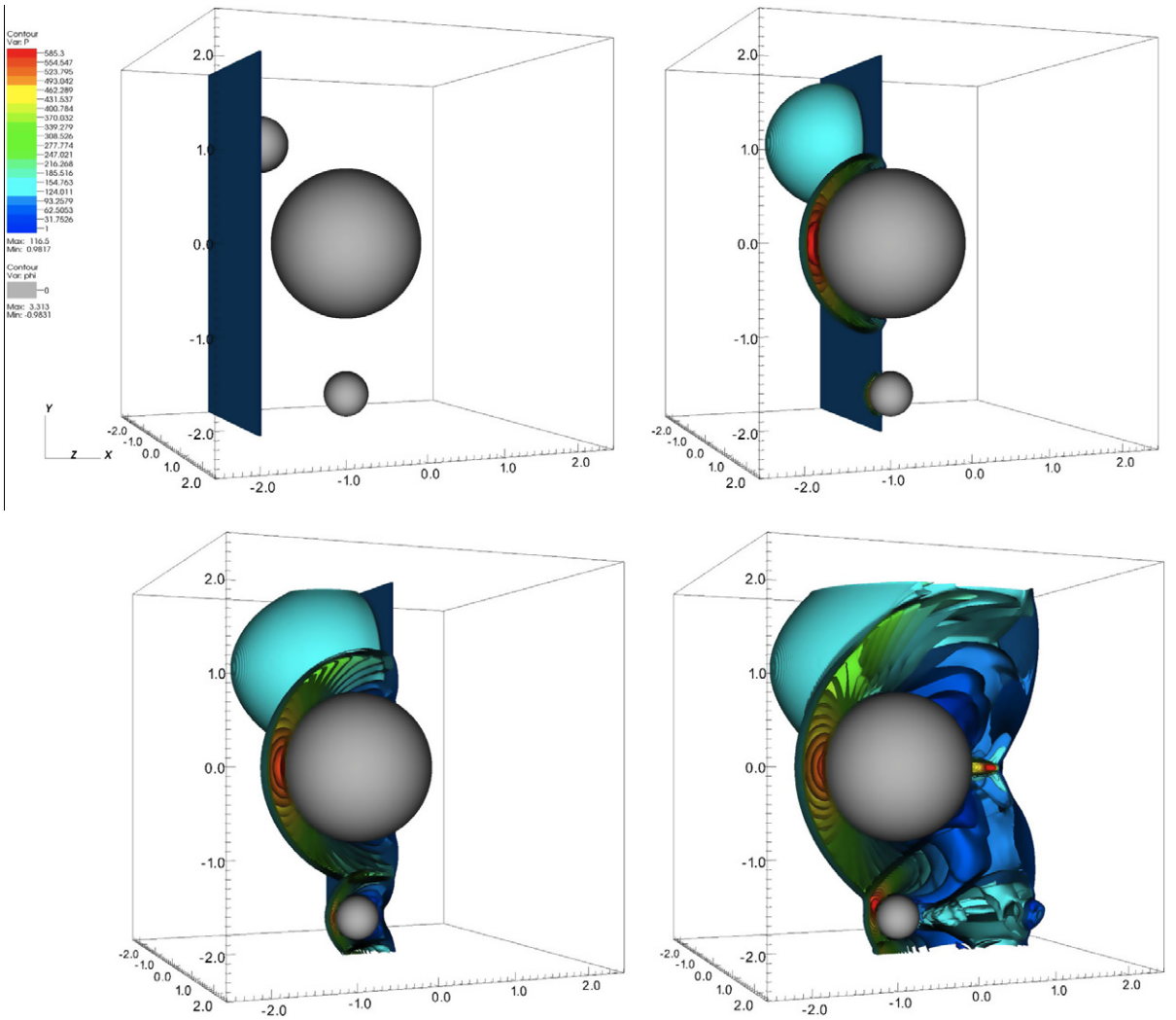


Fig. 14. Mach 10 planar shock reflecting on a 3D sphere. 20 isopressure surfaces.

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 the scheme is overall locally and globally second order accurate. 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 does not 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.

Acknowledgments

This research is funded in part by the EU FP7 project FFAST, ACP8-GA-2009-233665.

References

[1] S. Abarbanel, A. Ditkowski, B. Gustafsson, On error bounds of finite difference approximations to partial differential equations – temporal behaviour and rate of convergence, J. Scientific Comput. 15 (1) (2000) 79–116.
[2] P. Angot, C.-H. Bruneau, P. Fabrie, A penalization method to take into account obstacles in incompressible flows, Numer. Math. 81 (4) (1999) 497–520.

[3] E. Arquis, J. Caltagirone, 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.
[4] M. Berger, C. Helzel, R. LeVeque, h-box methods for the approximation of hyperbolic conservation laws on irregular grids, SIAM J. Numer. Anal. 41 (3) (2004) 893–918.
[5] M. Berger, R. LeVeque, An adaptive cartesian mesh algorithm for the euler equations in arbitrary geometries, 1989.
[6] M. Berger, R. LeVeque, Stable boundary conditions for cartesian grid calculations, Comput. Syst. Eng. 1 (2–4) (1990) 305–311.
[7] M. Bergmann, A. Iollo, Modeling and simulation of fish-like swimming, J. Comput. Phys. 230 (2011) 329–348.
[8] O. Boiron, G. Chiavassa, R. Donat, A high-resolution penalization method for large mach number ows in the presence of obstacles, Comput. Fluids 38 (3) (2009) 703–714.
[9] A. Chaudhuri, A. Hadjadj, A. Chinnayya, On the use of immersed boundary methods for shock/obstacle interactions, J. Comput. Phys. 230 (5) (2011) 1731–1748.
[10] W. Coirier, K. Powell, An accuracy assessment of cartesian mesh approaches for the euler equations, J. Comput. Phys. 117 (1995) 121–131.
[11] P. Colella, D. Graves, B. Keen, D. Modiano, A cartesian grid embedded boundary method for hyperbolic conservation laws, J. Comput. Phys. 211 (1) (2006) 347–366.
[12] A. Dadone, B. Grossman, Ghost-cell method for analysis of inviscid three dimensionnal flows on cartesian grids, Comput. Fluids 36 (2007) 1513–1528.
[13] E. 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.
[14] R. Fedkiw, Coupling an eulerian fluid calculation to a lagrangian solid calculation with the ghost fluid method, J. Comput. Phys. 175 (2002) 200–224.
[15] R. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (1999) 457–492.
[16] H. Forrer, R. Jeltsch, A high order boundary treatment for cartesian grid methods, J. Comput. Phys. 140 (1998) 259–277.
[17] R. Ghias, R. Mittal, H. Dong, A sharp interface immersed boundary method for compressible viscous flows, J. Comput. Phys. 225 (2007) 528–553.
[18] X. Hu, B. Khoo, N. Adams, F. Huang, A conservative interface method for compressible flows, J. Comput. Phys. 219 (2006) 553–578.
[19] J. Kim, D. Kim, H. Choi, An immersed boundary finite volume method for simulations of flow in complex geometries, J. Comput. Phys. 171 (2001) 132–150.
[20] B.V. Leer, Towards the ultimate conservative difference scheme I. The quest of monotonicity, Lect. Notes Phys. 18 (1972) 163.
[21] R. Leveque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM Numer. Anal. 31 (4) (1994) 1019–1044.
[22] M. Linnick, H. Fasel, A high order interface method for simulating unsteady incompressible flows on irregular domains, J. Comput. Phys. 204 (2005) 157–192.
[23] Q. Liu, O. Vasiliyev, A brinkman penalization method for compressible flows in complex geometries, J. Comput. Phys. 227 (2007) 946–966.
[24] S. Majumdar, G. Iaccarion, P. Durbin, Rans solver with adaptive structured boundary non conforming grids, 2001.
[25] G. Miller, P. Colella, A conservative three-dimensionnal eulerian method for coupled solid–fluid shock capturing, J. Comput. Phys. 183 (2002) 26–82.
[26] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, A. Von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, J. Comput. Phys. 227 (2008) 4825–4852.
[27] R. Mittal, G. Iaccarino, Immersed boundary methods, 2005.
[28] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, 2003.
[29] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (12) (1988).
[30] S. Osher, F. Soloman, Upwind difference schemes for hyperbolic systems of conservation laws, Math. Comp. 38, 158 (April 1982) 339–374.
[31] R. Pember, J. Bell, P. Colella, W. Crutchfield, M. Welcome, An adaptive cartesian grid method for unsteady compressible flows in irregular regions, J. Comput. Phys. 120 (1995) 278–304.
[32] C. Peskin, The fluid dynamics of heart valves: experimental, theoretical and computational methods, Annu. Rev. Fluid Mech. 14 (1981) 235–259.
[33] A. Sarthou, S. Vincent, P. Angot, J. Caltagirone, The sub-mesh penalty method, Finite Vol. Complex Appl. V (2008) 633–640.
[34] J.A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, Cambridge, UK, 1999.
[35] J.A. Sethian, Evolution, implementation, and application of level set and fast marching methods for advancing fronts, J. Comput. Phys. 169 (2001) 503–555.
[36] shan Jiang, G., Shu, C.-W., and L, I. 1995. Efficient implementation of weighted eno schemes. J. Comput. Phys 126, 202–228.
[37] A. Shapiro, The Dynamics and Thermodynamics of Compressible Fluid Flow, Ronald Press, 1953.
[38] B. Sjogreen, N.A. Petersson, A cartesian embedded boundary method for hyperbolic conservation laws, Commun. Comput. Phys. 2 (6) (2007) 1199–1219.
[39] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics – A Practical Introduction, second ed., Springer, Berlin, 1999.
[40] F. Tseng, A ghost cell immersed boundary method for flow in complex geometry, J. Comput. Phys. 192 (2003) 593–623.
[41] J.C. Vassberg, A. Jameson, In pursuit of grid convergence for two-dimensional Euler solutions, J. Aircraft 47 (4) (2010) 1152–1166.
[42] T. Ye, R. Mittal, H. Udaykumar, W. Shyy, An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries, J. Comput. Phys. 156 (1999) 209–240.