International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:5, No:10, 2011

2D validation of a high-order adaptive Cartesian-grid finite-volume characteristic-flux model with embedded boundaries

C. Leroy, G. Oger, D. Le Touzé, B. Alessandrini

***Abstract*—** A Finite Volume method based on Characteristic Fluxes for compressible fluids is developed. An explicit cell-centered resolution is adopted, where second and third order accuracy is provided by using two different MUSCL schemes with Minmod, Sweby or Superbee limiters for the hyperbolic part. Few different times integrator is used and be describe in this paper. Resolution is performed on a generic unstructured Cartesian grid, where solid boundaries are handled by a Cut-Cell method. Interfaces are explicitely advected in a non-diffusive way, ensuring local mass conservation. An improved cell cutting has been developed to handle boundaries of arbitrary geometrical complexity. Instead of using a polygon clipping algorithm, we use the Voxel traversal algorithm coupled with a local floodfill scanline to intersect 2D or 3D boundary surface meshes with the fixed Cartesian grid. Small cells stability problem near the boundaries is solved using a fully conservative merging method. Inflow and outflow conditions are also implemented in the model. The solver is validated on 2D academic test cases, such as the flow past a cylinder. The latter test cases are performed both in the frame of the body and in a fixed frame where the body is moving across the mesh. Adaptive Cartesian grid is provided by Paramesh without complex geometries for the moment.

***Keywords*—**Finite volume method, Cartesian grid, Compressible solver, complex geometries, Paramesh.

I. Introduction

NUMERICAL simulation is more and more widely used in hydrodynamics. Limited to simplified models of the flow around a body in frequency domain two decades ago, it now deals with the temporal modeling of complex unsteady phenomena. In particular, models based on the Navier-Stokes equations in Reynolds averaging (RANS) are now used routinely in applied research to solve complex realistic problems. Meanwhile, the commercialization of general codes of this kind knows an increasing success in the industry, starting to replace the previous empirical solutions. The RANS models based on volume mesh implicit methods still do not respond to all situations encountered in naval hydrodynamics and offshore. One can give as examples the simulation of two-phase phenomena caused by the progression of a ship at sea with its propeller (air entrainment in the jet bow, bubbles in the wake), violent sloshing impacts on long duration, etc. These examples raise the problem of safety of floating structures, their staff, their passengers and their cargo (often polluting) but they also concern energy reduction, ship signature, etc. The main limitations faced by standard solvers based on implicit methods on body-fitted unstructured meshes are : the presence of multiple bodies of complex geometry in arbitrary motion in the flow, non-diffusive interfaces between fluids, multi-physics within the solver, automatic mesh refinement, mesh adaptation in the frame of fluid-structure coupling. Here, we develop a different model based on a fixed Cartesian grid, an explicit resolution based on compressible Finite Volumes (FVCF method) enabling easy inclusion of multi-physics. Arbitrary complex geometries can be embedded in the fixed grid and move freely thanks to a developed cut-cell technique. Fully-conservative treatment of interfaces and automatic mesh refinement are presently being implemented. In the present paper, these different components are first presented, and followed by validation test cases.

II. FVCF Framework

A. Navier-Stokes & Inviscid Euler Equations

We model 3D Navier-Stokes equations for viscous compressible flow :

$$\overrightarrow{W_t} + F\overrightarrow{(W)_x} + G\overrightarrow{(W)_y} + H\overrightarrow{(W)_z} = \overrightarrow{Visc} \qquad (1)$$

which can be written in conservative form as

$$\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ \rho uw \\ (\rho E + P)u \end{pmatrix}_x + \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + P \\ \rho vw \\ (\rho E + P)v \end{pmatrix}_y + \begin{pmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + P \\ (\rho E + P)w \end{pmatrix}_z = \overrightarrow{Visc} \qquad (2)$$

The internal energy is linked to the total energy per unit volume and the kinetic energy by

$$E_{tot} = \rho E + \frac{\rho(u^2+v^2+w^2)}{2} \qquad (3)$$

An additional equation is needed to close the system by coupling the pressure with conservative variables $EOS(\rho,\rho U,\rho E) = 0$. We use either an ideal gas assumption or a barotropic fluid. On the one hand Tammann-Gibson's equation of state expresses the pressure as a function of the density, the internal energy and γ the polytrophic constant.

$$P = (\gamma - 1)\rho E = (\gamma - 1)(E_{tot} - \frac{\rho(u^2+v^2+w^2)}{2}) \quad (4)$$

On the other hand Tait's equation writes the pressure as a function of the density, γ the polytrophic constant, reference pressure P_0, nominal density ρ_0 and nominal speed of sound c_0

$$P - P_0 = \frac{\rho_0 c_0^2}{\gamma}\left[\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right] \quad (5)$$

The viscosity terms in Navier-Stokes equation can be represented in terms of the viscous stress tensor components as :

$$\overrightarrow{Visc} = \begin{pmatrix} 0 \\ \begin{pmatrix}\tau_{xx}\\ \tau_{xy}\\ \tau_{xz}\end{pmatrix}_x + \begin{pmatrix}\tau_{xy}\\ \tau_{yy}\\ \tau_{yz}\end{pmatrix}_y + \begin{pmatrix}\tau_{xz}\\ \tau_{yz}\\ \tau_{zz}\end{pmatrix}_z \\ \vec{\nabla}.\begin{pmatrix} u\tau_{xx} + v\tau_{xy} + w\tau_{xz} \\ u\tau_{xy} + v\tau_{yy} + w\tau_{yz} \\ u\tau_{xz} + v\tau_{yz} + w\tau_{zz}\end{pmatrix} \end{pmatrix} \quad (6)$$

B. Finite Volume Characteristic Flux (FVCF) schemes

The FCVF scheme, proposed by *Ghidaglia et al* [1] in 1996, is a finite volume scheme with cell centered variables, including the kinetic vector. The conservation laws are discretized by calculating numerical sum fluxes $\emptyset_{FVCF}$ through cell edges. These fluxes are expressed in terms of cell defined physical fluxes F, G and H in 3D, and not cell variables W as classical Godunov scheme. Introducing the flux of conservative variables in normal direction $\vec{n}$ as

$$\Psi(W,\vec{n}) = \vec{W}(\vec{U}.\vec{n}) + P\begin{pmatrix} 0 \\ \vec{n} \\ (\vec{U}.\vec{n})\end{pmatrix} \quad (7)$$

Let us define $V(t)$ a volume and its boundary $\delta V(t)$ which can be decomposed in time independent edges Γ and moving edges Ξ(t) with local velocity $\vec{U}_{int}$.

$$\iiint_{V(t)} \left(\overrightarrow{W_t} + F\overrightarrow{(W)_x} + G\overrightarrow{(W)_y} + H\overrightarrow{(W)_z}\right) d\vartheta = \frac{d}{dt}\iiint_{V(t)} \vec{W}\, d\vartheta - \iint_{\Xi(t)} \vec{W}\,(\vec{U}_{int}.\vec{n})dS + \iint_{\Xi(t)} \Psi(W,\vec{n})dS + \iint_{\Gamma} \Psi(W,\vec{n})dS \quad (8)$$

Introducing (7) one gets

$$\iint_{\Xi(t)} \Psi(W,\vec{n})dS = \iint_{\Xi(t)} \vec{W}\,(\vec{U}_{int}.\vec{n})dS + \iint_{\Xi(t)} P_{int}\begin{pmatrix} 0 \\ \vec{n} \\ (\vec{U}_{int}.\vec{n})\end{pmatrix} dS \quad (9)$$

The integrated system becomes :

$$\frac{d}{dt}\iiint_{V(t)} \vec{W}\, d\vartheta + \iint_{\Xi(t)} P_{int}\begin{pmatrix} 0 \\ \vec{n} \\ (\vec{U}_{int}.\vec{n})\end{pmatrix} dS + \iint_{\Gamma} \Psi(W,\vec{n})dS = 0 \quad (10)$$

In the finite volume framework we obtain :

$$\frac{Vol^{n+1}.\overrightarrow{W^{n+1}} - Vol^{n}.\overrightarrow{W^{n}}}{dt} + A.P_{int}\begin{pmatrix} 0 \\ \vec{n} \\ (\vec{U}_{int}.\vec{n})\end{pmatrix} + A.\emptyset_{FVCF} = 0 \quad (11)$$

where dt is a time step, and superscripts n and n+1 indicate time levels; $\vec{n}$ the unit normal vector to the face, and A the face area.

We discretize the Navier-Stokes equation as an hyperbolic part plus some viscosity terms. We use the Jacobian matrix $\bar{\bar{J}}$ associated to the hyperbolic system and the Tait's equation of state to build a flux solution $\emptyset_{FVCF}$. For this study we can rewrite (2) :

$$W_t + \sum_{i=1}^{\dim} \bar{\bar{J}}(W,\vec{e_i})F(W)_{x_i} = 0 \quad (12)$$

Then we need to multiply (12) by the same Jacobian matrix express at the mid interface of the two cells. MUSCL method can be used here to improve the reconstruction of the new interface conservative variables that will be explained later with more details. We use simple linear interpolation as follow, where L and R refer respectively left and right state of the mid interface.

$$\begin{cases} (F(W).\vec{n})_t + \overline{J}(W_{\text{int}},\vec{n}).(F(W).\vec{n})_n = 0 \\ W_{\text{int}} = \dfrac{Vol_L W_L + Vol_R W_R}{Vol_L + Vol_R} \end{cases} \quad (13)$$

For barotropic fluids hyperbolic conservation laws :

$$\bar{\bar{J}} = \frac{\partial \Psi(W,\vec{n})}{\partial W} = \begin{pmatrix} 0 & \vec{n} \\ c^2\vec{n} - (\vec{u}.\vec{n})\vec{u} & \vec{u}\otimes\vec{n} + \vec{u}.\vec{n}Id_{\dim+2} \end{pmatrix} \quad (14)$$

And with energy conservation equation:

$$\bar{\bar{J}} = \begin{pmatrix} 0 & \vec{n} & 0 \\ \Theta\vec{n} - (\vec{u}.\vec{n})\vec{u} & \vec{u}\otimes\vec{n} - \kappa\vec{n}\otimes\vec{u} + \vec{u}.\vec{n}Id_{\dim+2} & \kappa\vec{n} \\ (\Theta - H)(\vec{u}.\vec{n}) & H\vec{n} - \kappa(\vec{u}.\vec{n})\vec{u} & (1+\kappa)(\vec{u}.\vec{n}) \end{pmatrix} \quad (15)$$

Next step consist to compute eigensystem see in [2] for more details.

We remember two classical form of flux calculation, Godunov flux [14] :

$$\emptyset_{GODUNOV} = \Psi(\text{Wexa}(W_l, W_r), \vec{n}) \tag{16}$$

Where $\text{Wexa}(W_l, W_r)$ is the exact solution of a Riemann problem between left and right states of fluid. Roe flux [14] :

$$\emptyset_{ROE} = \frac{\Psi(W_r, \vec{n}) + \Psi(W_l, \vec{n})}{2} - |A_{ROE}| \frac{W_r - W_l}{2} \tag{17}$$

The algebraic reduction provides us with everything to work in the characteristic basis. The FVCF flux $\emptyset_{FVCF}$ can then be written in physical space in the following form

$$l_k(\overrightarrow{W_{int}}, \vec{n}).\emptyset_{FVCF} = \begin{cases} l_k(\overrightarrow{W_{int}}, \vec{n}).\Psi(W_r, \vec{n}) & \text{if } \lambda_k(\overrightarrow{W_{int}}, \vec{n}) > 0 \\ l_k(\overrightarrow{W_{int}}, \vec{n}).\left(\frac{\Psi(W_r, \vec{n}) + \Psi(W_l, \vec{n})}{2}\right) & \text{if } \lambda_k(\overrightarrow{W_{int}}, \vec{n}) = 0 \\ l_k(\overrightarrow{W_{int}}, \vec{n}).\Psi(W_l, \vec{n}) & \text{if } \lambda_k(\overrightarrow{W_{int}}, \vec{n}) < 0 \end{cases} \tag{18}$$

That can be regroup in general upwind form

$$\emptyset_{FVCF} = \frac{\Psi(W_r, \vec{n}) + \Psi(W_l, \vec{n})}{2} - \bar{\bar{R}}.Matrix_{sign(\bar{\bar{D}})}.\bar{\bar{L}} * \frac{\Psi(W_r, \vec{n}) - \Psi(W_l, \vec{n})}{2} \tag{19}$$

Conservation : $\emptyset_{FVCF}(W_l, W_r, \vec{n}) = -\emptyset_{FVCF}(W_r, W_l, \vec{n})$

Consistancy : $\emptyset_{FVCF}(\text{W}, \text{W}, \vec{n}) = \Psi(\text{W}, \vec{n})$

Remark 1. FCVF method have a close definition with Roe method, nonetheless the Roe matrix A_{ROE}isn't all the time calculable in complex case as multiphase flow whereas $Matrix_{sign(\bar{\bar{D}})}$ been.

Remark 2. FCVF method key point is to stay consistent in any linearization case of the left and the right state.

Remark 3. FCVF method naturally translates interface exchange between adjacent cells.

The timestep is defined by the CFL condition.

$$dt \le \min_i \left(\frac{Vol_i}{\Gamma_{\text{int}} \max_k |(\lambda_i)_k|} \right) \tag{20}$$

Where Γ_{int} is the interface face area.

C. Characteristic Boundary conditions

Inlet-oulet boundary conditions are written in the characteristic space, see [1]. Inflow velocity is given at the inlet which allows to calculate density and then pressure through the EOS.

$$\begin{cases} \overrightarrow{W_{inflow}} \\ l_{DIM}(\overrightarrow{W_{int}}, \vec{n}).\overrightarrow{W_{inflow}} = l_{DIM}(\overrightarrow{W_{int}}, \vec{n}).\vec{W} \\ \Rightarrow \rho_{inflow} = \frac{\rho c}{c + (\overrightarrow{U_{inflow}} - \vec{U})\vec{n}} \end{cases} \tag{21}$$

At the outlet we impose the pressure from which we calculate the density.

$$P_{outflow} \Rightarrow \rho_{outflow} = \rho \left(\frac{\gamma . P_{outflow}}{\rho_0 c_0^2} + 1 \right)^{\frac{1}{\gamma}} \tag{22}$$

At a wall, we consider the normal velocity to be equal to zero. Thus, to calculate the numerical flux at the wall we only have to determine the pressure as

$$\begin{cases} \emptyset_{FVCF}^{\text{Wall}} = P_{wall} \begin{pmatrix} 0 \\ \vec{n} \end{pmatrix} \\ l_{DIM}(\overrightarrow{W}, \vec{n}).P_{wall} \begin{pmatrix} 0 \\ \vec{n} \end{pmatrix} = l_{DIM}(\vec{W}, \vec{n}).\Psi(\vec{W}, \vec{n}) \\ \Rightarrow P_{wall} = P + \rho c(\vec{U}.\vec{n}) \end{cases} \tag{23}$$

D. Characteristic hybrid cut-cell method

Flow simulation around moving complex bodies is a challenging problem, especially when a fixed Cartesian grid is used. We have to find local grid modification on body surface, without significant increasing computational cost. We present here a new characteristics implementation procedure for an existing two-dimensional cell-merging method to overcome the problem of conservation laws. The present method may have a better potential for 3D extension.

The first step is to immerse a 2D or 3D complex geometry in a Cartesian grid and to provide volumes, intersection coordinates and solid edge normals for each intersected cell. Instead of using a polygon clipping algorithm [3] we use the Voxel traversal algorithm [10] coupled with a local floodfill scanline to intersect 2D or 3D complex surface meshes with fix Cartesian grids.

The second step is to define the cut-cell topologies which can be found in 2D/3D [7], and to define a modified FCVF scheme. Considering the 2D problem for simplicity, 12 configurations exist which can be viewed as 6 symmetric configurations [2] .These cell configurations modify the discretized scheme since one has now between 3 to 5 flux calculations to perform for each cell on specific non-equal edges. The finite volume scheme becomes:

$$W_{i,j}^{n+1} = W_{i,j}^{n} - \frac{dt}{Vol_{i,j}} \sum_{keedges} \delta l^{(k)} . \varphi_{FVCF}^{(k)} \quad (24)$$

Where *dt* is a time step, and superscripts *n* and *n*+1 indicate time levels; *m* is the total number of cell faces in the cell under consideration, *k* the face number, *n* the unit normal vector to the face, and $\delta l^{(k)}$ the face area.

(1) Computing the total flux of all cells at time level *n*;
(2) Merging the cut cells;
(3) Moving the body to the next time level, *n* + 1;
(4) Computing cell geometrical properties;
(5) Computing the solution;
(6) Separating the merged cut cells.

Cell merging treats very small cut cells, which can greatly reduce the size of a time step. Furthermore, it has been shown by Coirier et al.[17] that cell merging affects only the local computational accuracy, not its global value. However Cut-cell configurations induce the presence of some small volume cells whose local time step is low due to the CFL criterion. A fully conservative cell-merge method is thus added to avoid this difficulty. We thus define a minimum volume under which the small cell is merged with it closest neighbor cell. This technique results in a lower local accuracy but has no significant effect on global accuracy, see [11]. Since Van Leer's work [4], the MUSCL scheme (Monotonic Upstream-centered Scheme for Conservation Law) has been studied and widely used in simulation. In MUSCL schemes, the conservative solution vectors are described as piecewise linear polynomials for recovering second-order accuracy. A gradient ratio is introduced in each direction. We write a new conservative vector at the edge between two conform Cartesian cells using the two closest neighbor cells. A monotonic slope limiter is used to forbid non realistic or inversion reconstruction. The Minmod slope limiter function preserves a strict monotonic reconstruction but in practice we use the Sweeby slope limiter with β=1.5 which provides better results.

III. Improvements

We implemented two MUSCL [4],[7] methods on conservative solution vectors which are used for recovering second and third order accuracy and we investigated especially the role of flux limiters. Regarding time advance we investigated various 3rd and 4th-order schemes.

To get the best accuracy/CPU time compromise we selected a 4th-order multistep scheme that permited to use a CFL coefficient of 3.5, instead of 0.8 for a classical 3rd-order Runge-Kutta scheme. A class of high-order strong stability preserving (SSP) time discretization method was first developed in [16] and was called TVD (Total Variation Diminishing) like SSP Runge-Kutta integrator

$$\begin{cases} u^{(0)} = u^{(n)} \\ u^{(i)} = \sum_{k=0}^{i-1} \left(\alpha_{i,k} u^{(k)} + \Delta t \beta_{i,k} L(u^{(k)})\right) \\ u^{(n+1)} = u^{(m)} \\ \mathrm{c} = \min_{\mathrm{i,k}} \frac{\alpha_{i,k}}{|\beta_{i,k}|} \end{cases} , \alpha_{i,k} > 0 , i = 1,..,m \quad (25)$$

In order to find the best compromise between computation time and accuracy, it has been decided to implement the VFFC solver in an adaptive mesh refinement. Paramesh is a parallel AMR library based on the principle of block AMR developed by Berger et al [8]. Paramesh is a parallel and portable block AMR code. It allows the user to automatically refine a structured mesh (a Cartesian one in this case) according to a user specified criteria. Fig. 1 illustrates the principle of block refinement. Paramesh builds a structure of nested rectangular grids covering the entire studied area. The specificity of Paramesh is that each block has the same topology (here 4x4). This simplifies the refinement process and speeds up the calculation at the cost of slightly decreased code flexibility.

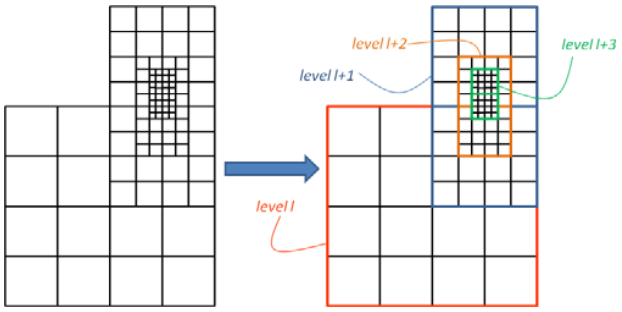

Fig. 5 Block AMR (4x4)

To implement the code parallelization at the least possible cost, the cells are distributed on different processors along a "Z-order "or" Morton order ". This optimizes the communication intra and extra processors. We assign guard cells to each block at its periphery. We need these guard cells to calculate the flux *balance* on the block boundary cells. The

guard cell filling is made according to the boundary condition type around the block. We distinguish three cases: fine/fine interface, coarse/fine interface and external boundary condition Fig. 3. In the first case, the guard cells take the neighboring cells values. In the second case, we must extrapolate the coarse neighboring cells to the guard cells. Finally, in the last case, the guard cell filling depends on the physical boundary condition used. As soon as guard cell filling is done, the flux *balance* is calculated for the block cells.

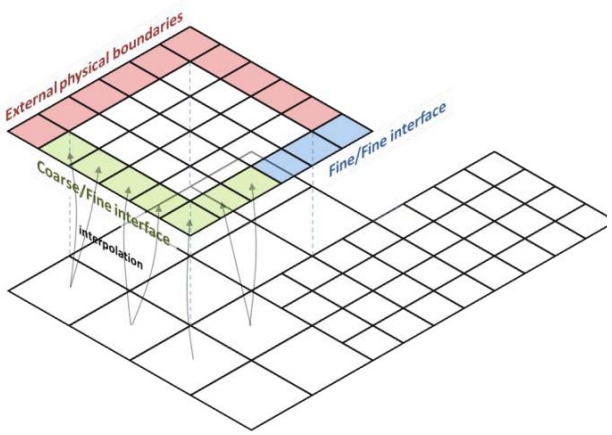

Fig. 7 Guard cell filling process

An interesting aspect of Paramesh is that it allows a non-permanent storage of the guard cells data, this leads to a reduction of the required memory size. Indeed, the guard cells data storage can be very costly, since the number of guard cells of a block is often higher than the number of cells in the block itself.

Paramesh supports conservation laws. Actually, at the interface between two blocks, inlet flux and outlet flux must be equal. This is ensured by a specific routine which applies an appropriate correction to the flow at the interfaces when necessary. Finally, post-treatment can be done with VisIt. This software can read HDF5 format, which is very light. A specific routine in Paramesh ensures the generation of an output HDF5 file. The implementation of a calculation with Paramesh is relatively easy. It is divided into six steps:

1. Boundary conditions definition
2. Initial conditions definition
3. Refinement criteria definition
4. CFL timestep definition
5. Computation of FVCF flux
6. Time advance

IV. Validations

A. 2D validation on 2D Shock Tube

Two dimensional Riemann problems can be seen as single one dimensional shock & rarefaction waves or two dimensional slip line with tangential velocity discontinuity, that define four couples of initial interface location which define nineteen possible configurations.

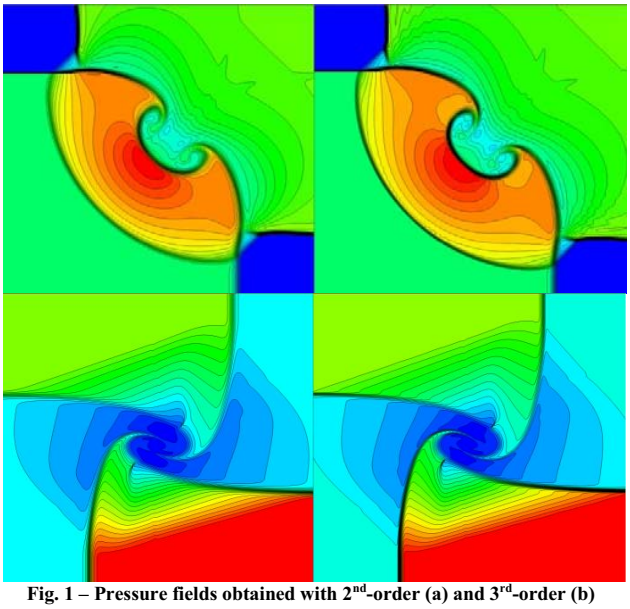

Fig. 1 – Pressure fields obtained with 2nd-order (a) and 3rd-order (b) schemes on 2D SOD configurations 11 (left) and 6 (right), see [16]

B. 2D Validation with simple & complex obstacles

Validation is made here against the SPH solver SPH-flow which uses a Riemann solver based on a Godunov resolution with MUSCL in an ALE formalism, see e.g. [5] In the present solver and in SPH-flow we use the same EOS, the same time integration (4^{th}-order Runge-Kutta scheme) and space convergence order. We define simple academic tests to validate FVCF solver in 2D with infinite inlet/outlet conditions. We fix inlet velocity to 5.0m/s and outlet pressure to the reference pressure. A second order MUSCL scheme with a minmod limiter is used in both cases. The figure below presents the comparison between the horizontal velocity obtained by SPH in a eulerian frame (particles are fixed on a cartesian lattice in SPH, a second-order convergence is thus recovered) and by the present solver. Very comparable fields are obtained by both methods. [2-17]. Next figure shows the field past a cylinder in terms of pressure, velocity components, and vorticity. This test case was made to compare the solution obtained in a frame of the cylinder with inlet-outlet conditions, and in a fixed frame where the cylinder is moving within the mesh. The same field patterns were obtained, showing the ability of the method to simulate the flow around moving bodies embedded in the fixed grid.

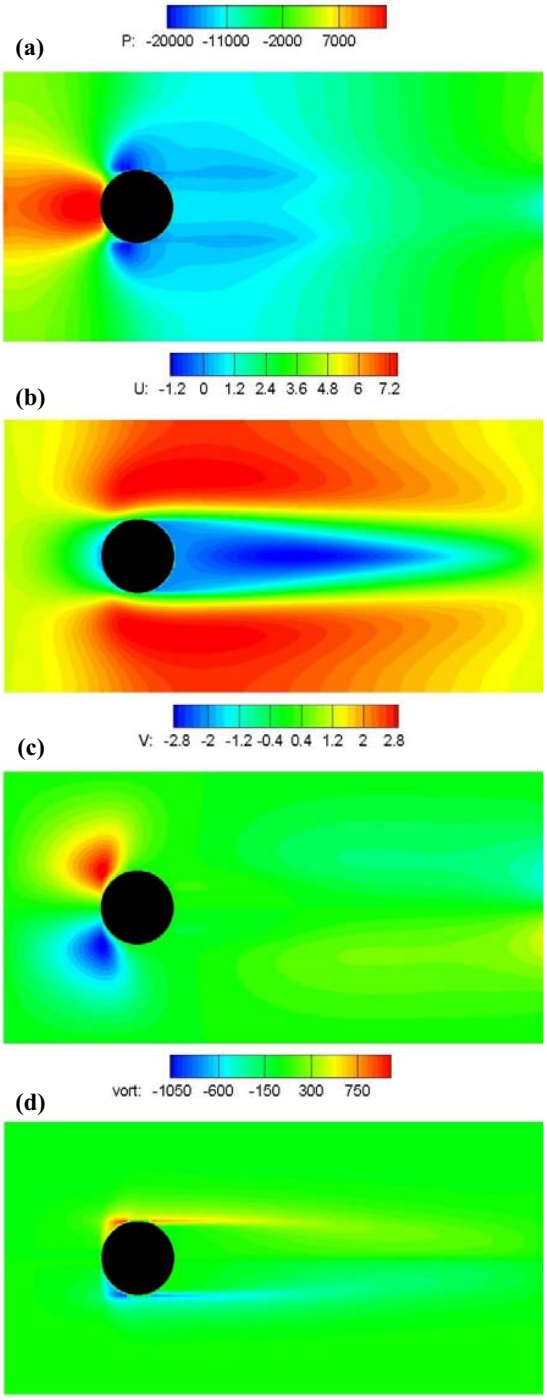

Fig. 2 - Flow past a cylinder, (a) Pressure (b) Longitudinal velocity (c) Normal velocity (d) Vorticity

C. 2D Validation for low Renolds flow.

The lid-driven cavity problem has long been used as validation test case for new codes or new solution methods. The geometry involved an this test case is a simple squared box tank, allowing to impose easily the boundary condition. The standard case is a fluid contained in a square domain with Dirichlet boundary conditions on all sides, with three fixed sides and one moving side (with velocity tangent to the side). Similar simulations have also been done at various aspect ratios, and it can also be done with the lid replaced with a moving fluid. This problem is a somewhat different situation, and is usually referred to as the shear-driven cavity. You may see the two names (lid-driven and shear-driven) used interchangeably in spite of the fact that they are distinct (and different) problems.

This problem has been solved with both laminar flow and a turbulent flows, and many different numerical techniques have been used to compute these solutions. A good set of data for comparison is the data of Ghia, and Shin (1982), since it includes tabular results for various Reynolds numbers. These simulation results are obtained using a non-primitive variable approach.

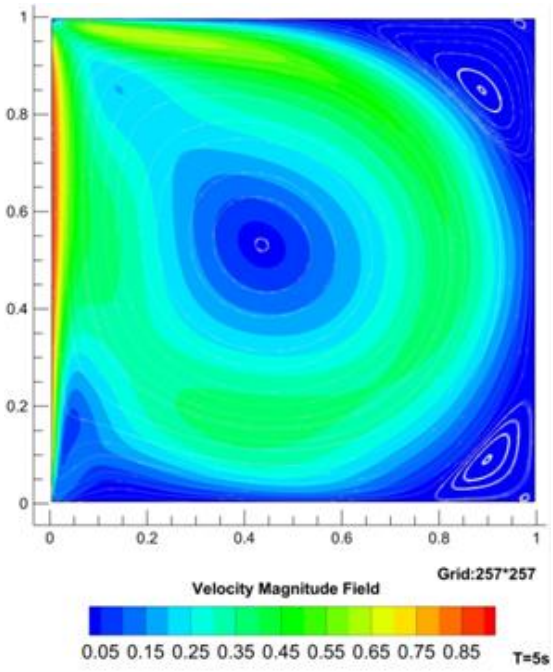

Fig. 4 Driven Flow in square cavity at Re=1000 after 5s

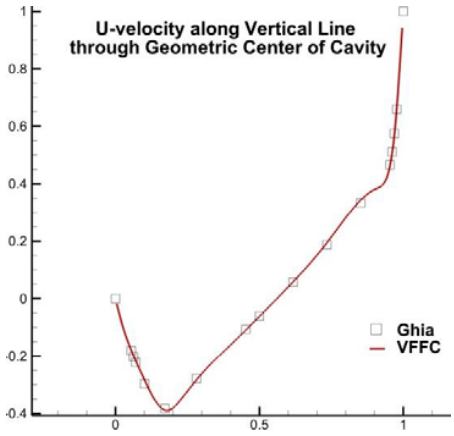

Fig. 5 U-velocity through Geometric Center of Cavity at Re=1000

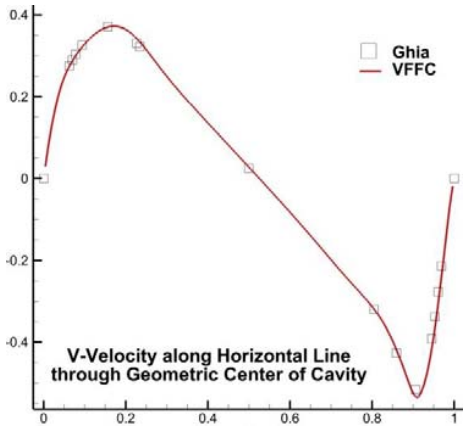

Fig. 6 V-velocity through Geometric Center of Cavity at Re=1000

Even with a full explicit viscous fluxes solver, we obtain an excellent comparison for Re<5000. Different discretizations are used and a very close agreement is observed already from the coarsest resolution. The results are also very good from Re=100 to 1000, at Re<1000 we obtain a stationary solution.

V. Conclusion

A new solver based on the Finite Volume Characteristic Flux method was developed. It is based on an explicit resolution relying on a fixed non-conform Cartesian grid into which bodies can freely move thanks to a dedicated cut-cell technique. First validations are presented showing very encouraging results. The solver has now to be extended to handle interfaces by means of a fully conservative technique. The mass loss of the level-set solver will be especially investigated, in particular by comparing the Particle Level-Set and Level-Set/VOF techniques on representative marine applications. Further developments will deal with automatic refinement of the mesh which will be eased by the Cartesian nature of the grid. Finally, turbulence will be modeled in the future through an LES scheme.

References

[1] J.-M. Ghidaglia, A. Kumbaro, G. Le Coq, Une méthode volumes finis à flux caractéristiques pour la résolution numérique des systèmes hyperboliques de lois de conservation, C.R. Acad. Sc. Paris, Vol.322, p. 981–988, (1996).
[2] C. Leroy and al., Development of a Cartesian-Grid Finite-Volume Characteristic Flux Model for Marine Applications", Materials Science and Engineering, Vol 10, (2010).
[3] Sutherland, I.E. and Hogdman, G. W. (1974) "Reentrant Polygon Clipping", Communication of the ACM, Graphics and Image Processing, Vol. 17, No 1, pp 32-42.
[4] B.V Leer, Towards the ult,mate conservative d,fference scheme V. A second order sequel to Godunov's methods, J. Comput Phys. 39, 101-136, (1979)
[5] J. P. Vila, "On particle weighted methods and SPH," Mathematical Models and Methods in Applied Sciences, vol. 9, pp. 161-210, 1999.
[6] Le Touzé D., Oger G. & Alessandrini B., "Smoothed Particle Hydrodynamics simulation of fast ship flows", Proc. of 27th Symp. on Naval Hydrodynamics (SNH 2008), Seoul, Korea, 2008.
[7] A. Kurganov, E Tadmor. Solution of Two-Dimensional Riemann Problems for Gas Dynamics without Riemann Problem Solvers.
[8] M. Berger, R. J. Leveque. Stable boundary conditions for Cartesian grid calculations. Computing Systems in Engineering, 1:305-311, 1990.
[9] D.M. Ingram, D.M. Causon, C.G. Mingham, Developments in Cartesian cut cell methods, Mathematics and Computers in Simulation 61 (2003) 561–572
[10] J. Amanatides, A. Woo. A Fast Voxel Traversal Algorithm for Ray Tracing Dept. of Computer Science University of Toronto , Ontario, Canada.
[11] B.V Leer, Towards the ultimate conservative difference scheme V. A second order sequel to Godunov's methods, JCP, (1979)
[12] D.M. Ingram, D.M. Causon, Developments in Cartesian cut cell methods, Mathematics and Computers in Simulation 61 (2003) 561–572
[13] J. Amanatides, A. Woo. A Fast Voxel Traversal Algorithm for Ray Tracing Dept. of Computer Science University of Toronto, Canada.
[14] Toro, Eleuterio F. (1999). Riemann Solvers and Numerical Methods for Fluid Dynamics. Berlin: Springer Verlag.
[15] A. Kurganov, E Tadmor. Solution of Two-Dimensional Riemann Problems for Gas Dynamics without Riemann Problem Solvers.
[16] CW Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, JCP, 77 (1988), pp. 439-471
[17] Coirier, W. J. and Powell, K. G.: An Accuracy Assessment of Cartesian-Mesh Approaches for The Euler Equations, J. Comput. Physics, **117** (1995), pp. 121–131.