



HAL
open science

A second-order curvilinear to Cartesian transformation of immersed interfaces and boundaries. Application to fictitious domains and multiphase flows

Arthur Sarthou, Stéphane Vincent, Jean-Paul Caltagirone

► To cite this version:

Arthur Sarthou, Stéphane Vincent, Jean-Paul Caltagirone. A second-order curvilinear to Cartesian transformation of immersed interfaces and boundaries. Application to fictitious domains and multiphase flows. *Computers and Fluids*, 2010, 46 (1), pp.422-428. 10.1016/j.compfluid.2010.11.008 . hal-00479705

HAL Id: hal-00479705

<https://hal.science/hal-00479705>

Submitted on 2 May 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A second-order curvilinear to Cartesian transformation of immersed interfaces and boundaries. Application to fictitious domains and multiphase flows

A. Sarthou, S. Vincent, JP. Caltagirone

TREFLE - ENSCPB, UMR 8508 16, avenue Pey-Berland, 33607 Pessac Cedex, France
sarthou@ipb.fr

Keywords: Curvilinear grids, structured Cartesian grids, fictitious domains, grid transformation, Front Tracking, VOF, Level-Set, multiphase flows

Abstract

A global methodology dealing with fictitious domains of all kinds on orthogonal curvilinear grids is presented. The main idea is to transform the curvilinear workframe and its associated elements (velocity, immersed interfaces...) into a Cartesian grid. On such a grid, many operations can be performed much faster than on curvilinear grids. The method is coupled with a Thread Ray-casting algorithm which work on Cartesian grids only. This algorithm computes quickly the Heaviside function related to the interior of an object on an Eulerian grid. The approach is also coupled with an immersed boundary method (L^2 -penalty method) or with phase advection with VOF-PLIC, VOF-TVD, Front-tracking or Level-set methods. Applications, convergence and speed tests are performed for shape initializations, immersed boundary methods, and interface tracking.

1 Introduction

Structured Cartesian grids are convenient to use in CFD simulations. However, this kind of mesh structure involves many limitations on the shape of the numerical domain. A first solution is to use fictitious domain methods (*e.g.* immersed boundary methods) which enable the immersion of complex boundaries on Cartesian grids. An other solution is to use curvilinear structured grids which couple some advantages of the structured approach (trivial connectivities) with the possibility to deal accurately with complex numerical boundaries and refined boundary layers.

In [18], Thompson proposes a general method to work with curvilinear grids. The main idea is to consider a fixed square grid in a rectangular transformed region where the numerical calculations are performed with modified operators. This approach can be extended to moving boundaries or interfaces and does not require the effective construction of the transformed domain. Nevertheless, the question of the motion and reconstruction of the mesh (especially the implementation effort and the computational cost), as for the ALE method, is

asked. This point is critical for multiphase flows and particularly for dispersed phases or for cases with many moving objects, such as fluidized beds. Even if less accurate, the fictitious domain methods seem to be more convenient and fast as the calculation grid can remain fixed.

One can find some examples of fictitious domains on curvilinear grids in the literature. An adaptation of the VOF-PLIC method on curvilinear grids can be found in [6]. In [10], Muradoglu and Kayaalp use an auxiliary Cartesian grid superimposed with the curvilinear mesh to manage the front tracking operations. Kernel functions are used to interpolate the velocity from one grid to another. In [5], Huang *et al.* extend the Ghost-fluid method [4] to curvilinear grids. In [3, 14], authors perform a fluid-structure coupling with an immersed boundary method on curvilinear grids. In these works, the transformation defined by [18] is not used.

The present work proposes an optimized approach using the transformation of [18] to deal with immobile boundaries while moving interfaces and boundaries are treated with fictitious domain methods. A transformation of the original immersed interface is proposed and its projection onto a volume function is accelerated thanks to the use of the transformed Cartesian space. Contrary to the aforementioned references, immersed boundary methods and phase advection are performed considering only the transformed Cartesian grids with the original *ad hoc* Cartesian algorithms.

2 Methodology

The method is presented in 2D. The extension to 3D problems is straightforward. Let us consider a curvilinear orthogonal structured mesh \mathcal{T}_h covering a numerical domain Ω . Its structured vertices are denoted \mathbf{x}_c (c denotes the doublet of indexes (i, j)), with $1 \leq i \leq L$, $1 \leq j \leq M$ (and $1 \leq k \leq N$ in 3D). The mesh cells are denoted K_c with \mathbf{x}_c the south-west vertex of each cell. An embedded domain of complex interface Σ divides Ω in two subdomains Ω_0 and Ω_1 . If fluid flows are considered, each subdomain can represent a phase or a solid object. The base principle is to unfold the curvilinear triangulation \mathcal{T}_h to obtain a Cartesian grid $\hat{\mathcal{T}}_h$ of unit space step defining a new computational domain $\hat{\Omega}$. For this purpose, we define a bi-continuous bijective operator $F : \Omega \mapsto \hat{\Omega}$ such as the transformed position of a grid node \mathbf{x}_c is $F(\mathbf{x}_c) = \hat{\mathbf{x}}_c = (i, j)^T$. Once the Cartesian grid has been built, the immersed interface Σ has to be unfolded too. Numerically, Σ is piecewise linear and can be defined by its vertices denoted by σ_l of position \mathbf{x}_l . In the Cartesian frame, Σ becomes a deformed interface $\hat{\Sigma}$ composed of vertices $\hat{\sigma}_l$. The transformation F is built cell by cell. The restriction of F to K_c is the operator $F_c = (F_c^1, F_c^2)^T$, and we have $F_c^1(\mathbf{x}_c) = \hat{x}_c$, $F_c^2(\mathbf{x}_c) = \hat{y}_c$. Each operator F^n is built with a Q_1 polynomial. In 2D, $Q_1(x, y) = a_1xy + a_2x + a_3y + a_4$ and in 3D, $Q_1(x, y, z) = a_1xyz + a_2xy + a_3yz + a_4zx + a_5x + a_6y + a_7z + a_8$. The coefficients of each Q_1 are defined by solving two systems composed of the equations $F_c^1(\mathbf{x}_{c,n}) = \hat{x}_{c,n}$, $n = 1, 2, 3, 4$ and the equations $F_c^2(\mathbf{x}_{c,n}) = \hat{y}_{c,n}$, $n = 1, 2, 3, 4$ where $\mathbf{x}_{c,n}$ are the positions of the 4 vertices of the cell K_c (see Fig. 1). Consequently, 3 systems of 8 equations are solved in 3D for all cells. The analytical solution can be used in 2D while a numerical solver is required in 3D.

The new position of each Lagrangian vertex σ_l can be obtained with $\hat{\mathbf{x}}_l =$

$F(\mathbf{x}_l)$. The cell containing the vertex is found thanks to a search algorithm. The distance between the vertex and a first Eulerian node is calculated. A new node which is a neighbor of the previous one is chosen such as the distance to the vertex is decreasing. Such a "point-in-cell" operation has to be performed only at the start of the whole calculation as the Cartesian frame is used for the further localizations. As demonstrated below, the computational cost of

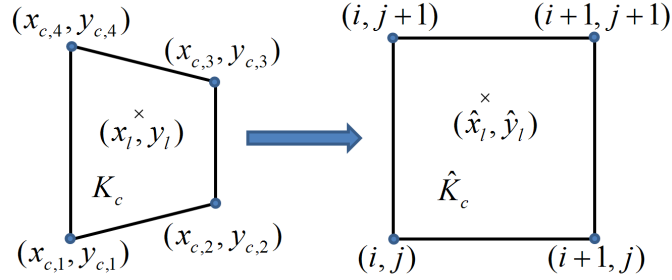


Figure 1: Notations in 2D for original and transformed cells

this transformation is negligible. The Fig. 2 shows an initial interface in a curvilinear mesh and its transformed shape in the Cartesian frame.

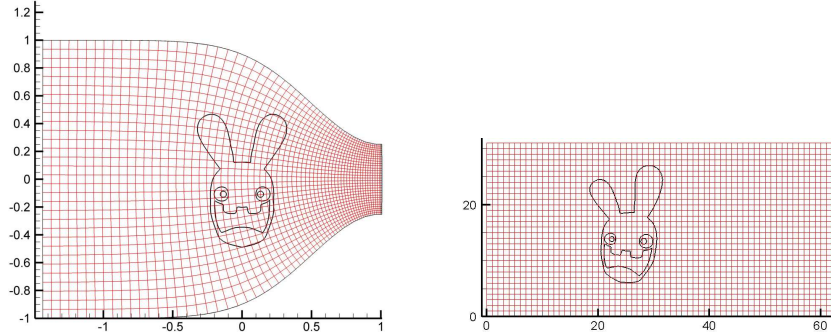


Figure 2: The original and transformed meshes \mathcal{T}_h and $\hat{\mathcal{T}}_h$ and the related immersed interfaces Σ and $\hat{\Sigma}$

2.1 Construction of volume functions

2.1.1 Heaviside function

The discrete binary Heaviside function χ is defined as:

$$\chi(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_1 \\ 0.5 & \text{if } \mathbf{x} \in \Sigma \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

This function is the binary indicator of the presence of an Eulerian point in Ω_1 and is built with a point-in-solid method presented below. An first-order accurate localization of the interface can be retrieved with χ . This function is

obtained with a Ray-casting (RC) method. The principle is to cast a ray from each Eulerian point to infinity and to test the number of intersections between the ray and the Lagrangian mesh. If the number of intersections is odd, the Eulerian point is inside the object, and outside otherwise. The RC method can be enhanced by classifying the elements of the Lagrangian mesh with an octree sub-structure [12] which recursively subdivides the space in boxes. If a ray does not intersect a box, it does not intersect the triangles inside the box. Some details on the implementation and a short review of point-in-solid strategies can be found in [11].

Concerning the standard RC method, its computational cost is generally very expensive as $L \times M \times N \times S$ intersections tests have to be performed, with S the number of elements of the Lagrangian mesh. Acceptable times are reached (with $L \times M \times N \times \log(S)$ intersections) with an octree but the implementation effort is non-negligible. A simple optimization, called here the Thread Ray-casting (TRC) which is known in the computer graphics community, is used. For the standard RC method, the direction of a ray is indifferent. If all rays are launched in the same direction, Ox for instance, many intersection tests are done more than once for a set of points in a same Eulerian mesh row in the Ox direction. Hence, only one ray can be cast per row. If rays are cast in a given direction (the best choice is the one with the greatest number of cells), the computational cost is divided by the number of cells in this direction and we have $(L \times M \times N \times S) / \max(L, M, N)$ intersections to test. Ideally, this method is coupled with the octree.

It is important to notice that this method works only on Cartesian meshes. The computational cost of the curvilinear to Cartesian transformation is justified by the gain of times between the RC and the TRC.

As F is bi-continuous and bijective, $\mathbf{x}_{i,j,k} \in \Omega_1 \Leftrightarrow \hat{\mathbf{x}}_{i,j,k} \in \hat{\Omega}_1$ and $\hat{\chi}(\hat{\mathbf{x}}_{i,j,k}) = \chi(\mathbf{x}_{i,j,k})$. As a consequence, the discrete Heaviside function in structured storage verifies $\chi_{(i,j,k)} = \hat{\chi}_{(i,j,k)}$ and χ can be computed in the Cartesian frame. This property is however true for the continuous transformation. From a discrete point of view, this property is true for a sufficient number S of Lagrangian elements σ_l .

2.1.2 The Level-set function

The level-set (LS) function ϕ is

$$\phi(\mathbf{x}) = \begin{cases} -\text{dist}_{\Sigma}(\mathbf{x}) & \text{if } \mathbf{x} \in \Omega_1 \\ \text{dist}_{\Sigma}(\mathbf{x}) & \text{otherwise} \end{cases} \quad (2)$$

with $\text{dist}_{\Sigma}(\mathbf{x}) = \inf_{\mathbf{p} \in \Sigma} \|\mathbf{x} - \mathbf{p}\|$. The unsigned distance is computed geometrically and the sign is obtained from χ . If the distance is calculated for all Eulerian nodes (a simple optimisation is to compute the distance only near the interface), this operation is generally expensive as $L \times M \times N \times S$ distance tests have to be performed. Details on implementation and optimisations can be found in [7]. In 1D, the LS function gives the exact location of the interface. For higher dimensions, a second-order localization of the interface can be obtained by taking $\phi = 0$. The normal \mathbf{n} and the curvature κ of the LS function are obtained with

$$\mathbf{n} = \frac{\nabla\phi}{\|\nabla\phi\|}, \quad \kappa = \nabla \cdot \left(\frac{\nabla\phi}{\|\nabla\phi\|} \right) \quad (3)$$

These quantities are wrong in the curvilinear domain if the Level-set function is computed in the Cartesian domain. However, the interface normal \mathbf{n} in the curvilinear frame can be retrieved from $\hat{\phi}$ computed in the Cartesian frame with

$$\mathbf{n} = \frac{\left(\frac{1}{\Delta\xi}, \frac{1}{\Delta\eta}, \frac{1}{\Delta\zeta}\right)^T \hat{\nabla} \cdot \hat{\phi}}{\left\| \left(\frac{1}{\Delta\xi}, \frac{1}{\Delta\eta}, \frac{1}{\Delta\zeta}\right)^T \hat{\nabla} \cdot \hat{\phi} \right\|} \quad (4)$$

with ξ, η, ζ the curvilinear coordinates, $\Delta\xi, \Delta\eta, \Delta\zeta$ the local mesh steps and $\hat{\nabla}$ the Cartesian gradient operator. Using \mathbf{n} , the curvature can be computed with the curvilinear divergence operator.

2.1.3 The VOF function

The volume-of-fluid (VOF) function C , also called the color or phase function, is the volume ratio of a given phase in an elementary volume (generally the finite volume cell).

This function is typically used to localize a fluid phase in multiphase flows and is the base of the 1-fluid model [8]. This function can be computed from the values of the Level-set function near the interface with the formula of [17]:

$$C(\mathbf{x}) \approx \begin{cases} 0 & \text{if } \phi < -\epsilon \\ \frac{1}{2} \left(1 + \frac{\phi}{\epsilon} - \frac{1}{\pi} \sin(\pi\phi/\epsilon) \right) & \text{if } |\phi| \leq \epsilon \\ 1 & \text{if } \phi > \epsilon \end{cases} \quad (5)$$

with ϵ the interface width. An alternative method is proposed in [16] and does not require χ . The following Poisson equation is solved:

$$\nabla^2 C = \nabla \cdot \int_{\Sigma} \mathbf{n} \delta(\mathbf{x} - \mathbf{x}_i) d\Gamma \quad (6)$$

where \mathbf{n} is the unit normal to the interface, δ a smoothed Dirac function, \mathbf{x} a position on the Eulerian grid and \mathbf{x}_i a marker (vertex) position on Σ . The original interface is recovered in $C = 0$.

2.2 Immersed boundary method

A L^2 -penalty method is used to impose an immersed Dirichlet boundary condition on Σ . A penalty term $\frac{\chi}{\epsilon}(u - u_l)$ is added to the initial conservation equation. As $\epsilon \ll 1$, the initial equation is negligible where $\chi = 1$ (*i.e.* in Ω_1). To reach a second order of accuracy, the sub-mesh penalty method (SMPM) [15] is used. The penalty term is discretized as $\frac{\chi}{\epsilon}(P_i u - u_l)$ where P_i is a local interpolator of the solution from both sides of the interface. Polynomials are used to construct P_i . As the interface location is required to build the penalty term, the intersections between Σ and the mesh lines are used. These points can be found with a geometric intersection test, or using the Level-set and VOF functions. Both give an implicit representation of the interface through their isosurfaces ($\phi = 0$ and $C = 0.5$). As demonstrated later, choosing one or the other impacts on the accuracy of the SMPM.

2.3 Phase advection

The advection of phases can be performed using the Cartesian frame. The initial advection equation is written

$$\frac{\partial C}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{C} = \mathbf{0} \quad (7)$$

with ∇ the curvilinear gradient operator. One can write

$$\nabla = \left(\frac{\partial}{\partial \xi}, \frac{\partial}{\partial \eta}, \frac{\partial}{\partial \zeta} \right)^T = \left(\frac{\partial x}{\partial \xi}, \frac{\partial y}{\partial \eta}, \frac{\partial z}{\partial \zeta} \right)^T \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)^T \quad (8)$$

Consequently,

$$\frac{\partial C}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{C} = \frac{\partial C}{\partial t} + \hat{\mathbf{u}} \cdot \nabla \hat{\mathbf{C}} = \mathbf{0} \quad (9)$$

with the transformed velocity in the Cartesian frame $\hat{\mathbf{u}} = \left(\frac{u_\xi}{\Delta \xi}, \frac{u_\eta}{\Delta \eta}, \frac{u_\zeta}{\Delta \zeta} \right)^T$ and the Cartesian gradient operator $\hat{\nabla} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)^T$. For each cell, $\frac{\partial x}{\partial \xi}$, $\frac{\partial y}{\partial \eta}$ and $\frac{\partial z}{\partial \zeta}$ are replaced by the ratios of the local space steps (with $\Delta x = \Delta y = \Delta z = 1$). If we consider in 1D the initial position x_0 of a characteristic curve, its evolution in the curvilinear frame is given by $x(t) = x_0 + ut$. If the characteristic curve covers a cell in a time T with a velocity u in the original frame, one can see that a velocity $u\Delta x/\Delta \xi$ is required to cover a transformed cell in the same time.

2.4 Other applications

As the Cartesian frame has a unit space step, any point (x_l, y_l, z_l) of the Cartesian space belongs to the cell $\hat{K}_{[x_l], [y_l], [z_l]}$. This property saves computational time for point-in-cell operations, *e.g.* when a quantity has to be extrapolated from the Eulerian grid to a Lagrangian point. In [20], the 3D simulation of the hydroplaning of a tire is performed on an irregular mesh and the present algorithms are used to deal with multiphase flows and the calculation of the forces exerted by the fluid on the tire. The pressure and stress tensor are extrapolated on the Lagrangian surface of the tire in order to determine the drag and lift forces. For each surface element, the extrapolation requires to pick 3 points in the fluid and to interpolate a quantity from the Eulerian nodes to these Lagrangian nodes. The same occurs for the Euler-Lagrange advection methods, such as the VOF-SM method [21], which requires many coupling between Lagrangian and Eulerian informations. For each interpolations from the Cartesian grid to a Lagrangian point, a unique simple Q_1 interpolation is used.

2.5 Parallelism

As these operations are generally independently performed with respect to Eulerian or Lagrangian elements, the present algorithms are easy to parallelize with OpenMP or GPGPU approaches. Concerning domain decomposition, for MPI approaches, the interface Σ is not always entirely inside or outside the computational domain treated by a processor. The present algorithms generally works in this case, except for the LCR VOF construction (6) (the resolved equation requires the surface to be entirely inside the domain), and the curvilinear to

Cartesian transformation. If a Lagrangian vertex σ_l is not inside an Eulerian cell, its transformation cannot be determined (eventually approximated for some curvilinear grids with irregular mesh steps and straight boundaries). For both cases, the solution is to cut Σ and to close it with parts of the segments which defines the boundaries of the Eulerian grid.

3 Results

3.1 Computational time

Speed tests have been performed on a P4 2.4 GHz for several meshes with and without the octree. Three Lagrangian meshes are used, a sphere (18000 triangles, regular), the Stanford bunny (Fig. 3, 10122 triangles, irregular) and the Lascaux cave ([9], 271136 triangles, irregular).

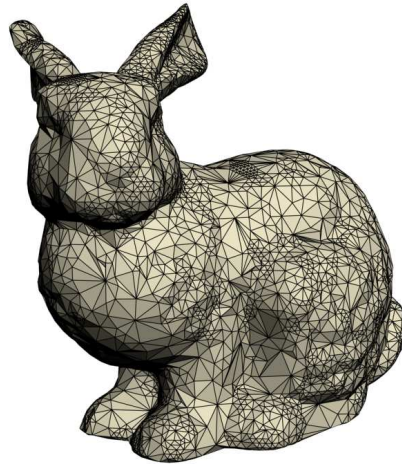


Figure 3: The Stanford bunny

The Tab. 1 shows computational times for these three Lagrangian meshes coupled with a 100^3 Eulerian mesh, for three operations: Transf. is the curvilinear to Cartesian transformation, TRC the Thread Ray-casting method and LS the computation of the Level-set function. The computational time of the standard Ray-casting method is not measured here but is supposed to be 100 times higher than for the TRC. Hence, even if the transformation is longer to perform than the TRC, the total cost is negligible compared to the RC. As expected, all the routines are faster with the octree. The gain for the curvilinear to Cartesian projection is between 1.35 and 1.73 only. For this case, a derivation of the octree for point-in-cell operations, a kD-tree [2], is used. For the Ray-casting, the gain is smaller than 2 on small meshes, and more than 10 for the Lascaux cave, where an octree with more subdivisions can be used. For the computation of the Level-set, the gain ratio is from 35 to 75.

The most important result is that the maximum computational time for the whole method is shorter than a minute on a P4 2.4 GHz. This time is generally

Mesh	Method	Transf.	TRC	LS
Sphere	Standard	2.06	0.536	270
	Octree	1.37	0.332	5.79
	Ratio	1.50	1.61	46.6
Bunny	Standard	1.19	0.172	123
	Octree	0.88	0.132	1.63
	Ratio	1.35	1.30	75.5
Lascaux	Standard	45.4	35.9	1970
	Octree	26.1	3.32	56.1
	Ratio	1.73	10.8	35.1

Table 1: Duration in second and performance ratio for three different meshes with and without the octree

negligible against the computational time required to solve the linear system resulting from the discretization of a conservation equation with a 100^3 mesh.

Concerning the computational time of the SMPM, the construction of the penalty term is negligible when volume functions are used. When the Lagrangian elements are used, the method takes roughly the same time as the TRC.

3.2 Immersed boundary method

The SMPM coupled with diverse surface representations is tested here. The accuracy of the surface representations can be deduced from the accuracy of the SMPM for each representation. The Laplace equation is solved with two curvilinear grids (see Fig. 4). A circular interface is considered and a Dirichlet boundary condition is imposed on the circle with the SMPM [15]. This method has a second order of space accuracy and requires the accurate location of the boundary. The computational mesh is a curvilinear converging pipe. Fig. 5 left shows a spatial convergence study for the L^2 and L^∞ error norms. The interface is located either with $\tilde{\Sigma}$ (GI), the curvilinear Level-set function (LS-CURVI), the Cartesian Level-set function (LS), the Sussman Heaviside reconstructed function (SUS) 5 and the VOF function from the Front-Tracking method (FT). As can be seen, the GI, LS-CURVI and LS methods allow the SMP method to almost keep its second order. The expected loss of accuracy between the LS-CURVI and the LS methods is too slight to disqualify the use of the Cartesian grid. The SUS method shows lower performances due to its construction using a sinus function (with a linear reconstruction, the same accuracy as with the LS is found). The FT shows a good convergence for the first meshes (with a second order) but is globally the less accurate function and does not converge for the finest meshes. An implementation error or a saturation effect could be involved.

3.3 Multiphase flows

The methodology is used with VOF-PLIC [22], Level-Set [17], VOF-TVD [19] and LCR Front-Tracking [16] advection methods. The Fig. 5 right shows a cross-shaped phase advected at a front positive speed on a curvilinear grid with

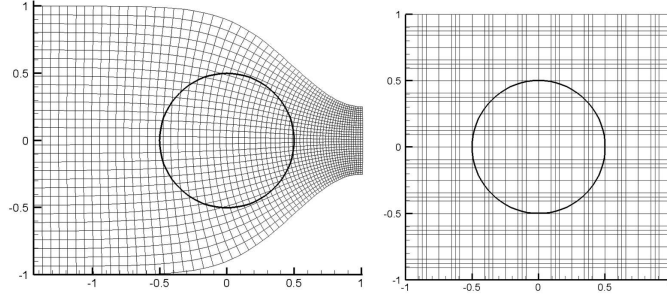


Figure 4: A and B meshes

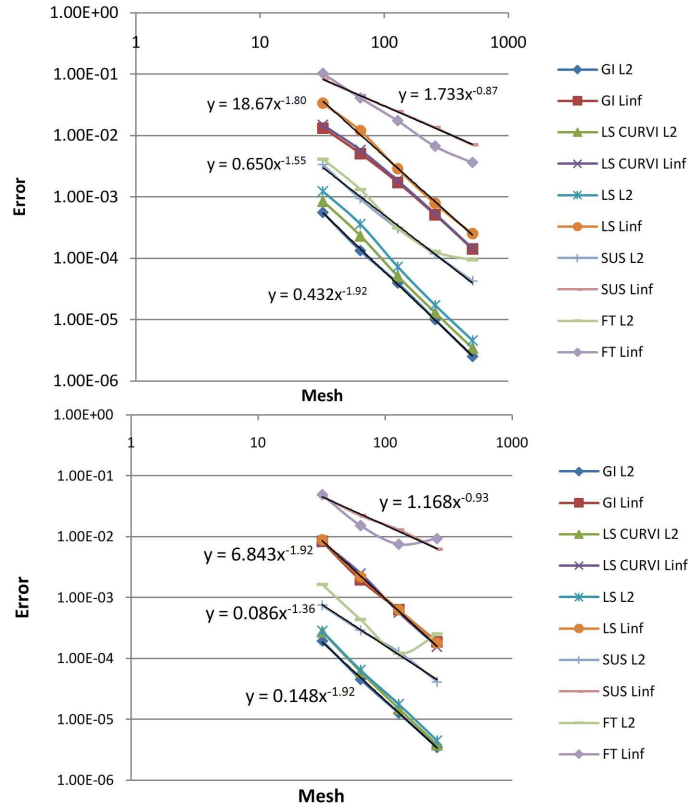


Figure 5: Relative L^2 and L^∞ errors for meshes A and B

a Front-Tracking method. The shape and mass of the initial phase are well conserved if only the Lagrangian markers are considered (which is not the case for the 1-fluid model where the VOF function is used). Fig. 7 shows the convergence of the error on the mass conservation. As expected, a regular second-order of convergence is found for the LS and FT methods. Concerning the VOF-PLIC and VOF-TVD methods, the transformation removes their ability to exactly conserve the phase volume and the conservation depends on the accuracy of the interface advection. It is known for the VOF-PLIC method that the convergence order of the interface position decreases with the mesh [13, 1]. Whatsoever, the

VOF-PLIC and VOF-TVD methods remain globally more accurate than the two other methods. For the FT method, the error on the advection of the Lagrangian markers is negligible against the error on the LCR operation 6. Contrary to the test in section 3.2, the saturation is not reached and a similar second order is obtained. However, the reconstruction of the VOF function using the method of [16] could be replaced by a more accurate method. Concerning the TVD method, good results are obtained but the cause of its irregular convergence has to be investigated.

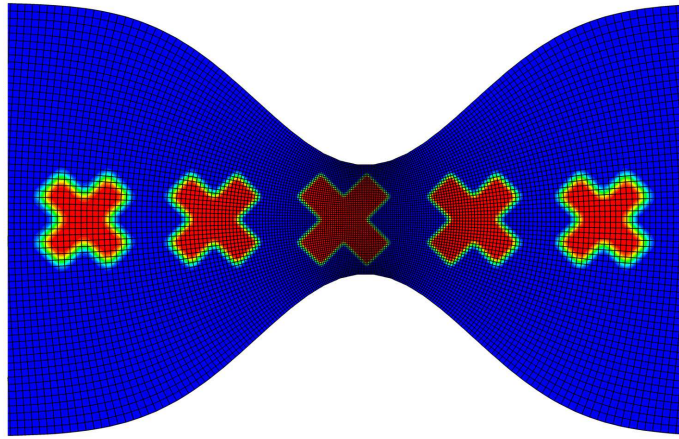


Figure 6: Advection of a cross-shaped phase with a Front-tracking method

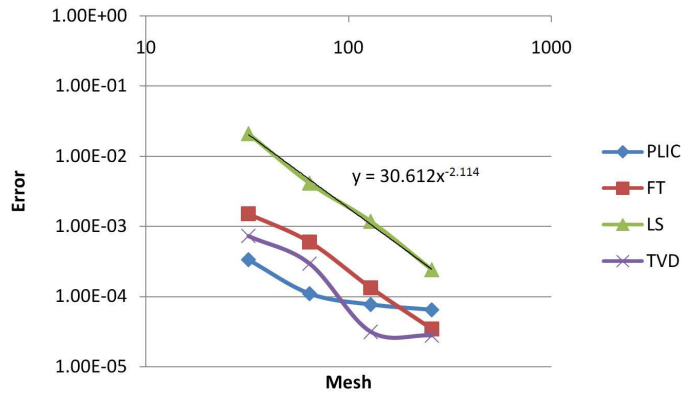


Figure 7: Relative error on the mass conservation

4 Conclusion

A new second-order methodology to work with fictitious domains on curvilinear grids has been drawn here. It allows methods implemented on Cartesian grids to be easily extended to curvilinear grids saving implementation efforts and computational cost. The Thread Ray-casting is used on curvilinear grids

and greatly improves the performances of the Heaviside construction. Used with many implicit and explicit interface representations, a second-order immersed boundary method keeps its convergence order. Concerning the phase advection, a transformed velocity field has been coupled with the Cartesian frame to perform the advection for curvilinear cases with Cartesian algorithms. Future works will be devoted to find a way to retrieve the volume conservation property for the VOF-TVD and VOF-PLIC methods. Higher-orders curvilinear to Cartesian transformations for the mesh will be studied too. Concerning the FT method (for all kinds of meshes), the initialisation of [16] will be replaced by the TRC and a more accurate reconstruction of the VOF function (using the Level-set function or a geometrical calculation of the volume ratio).

Acknowledgment

The authors gratefully thanks Antoine Lemoine for the implementation of the Octree and kdTree.

References

- [1] E. Aulisa, S. Manservigi, R. Scardovelli, and S. Zaleski. Interface reconstruction with least-squares fit and split advection in three-dimensional cartesian geometry. *Journal of Computational Physics*, 225(2):2301 – 2319, 2007.
- [2] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, 1975.
- [3] Iman Borazjani, Liang Ge, and Fotis Sotiropoulos. Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3d rigid bodies. *Journal of Computational Physics*, 227(16):7587 – 7620, 2008.
- [4] Ronald P. Fedkiw, Tariq Aslam, Barry Merriman, and Stanley Osher. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics*, 152(2):457 – 492, 1999.
- [5] Juntao Huang, Pablo M. Carrica, and Frederick Stern. Coupled ghost fluid/two-phase level set method for curvilinear body-fitter grids. *International Journal for Numerical Methods in Fluids*, 55:867–897, 2007.
- [6] W. Jang, J. Jilesen, F. S. Lien, and H. Ji. A study on the extension of a vof/plic based method to a curvilinear co-ordinate system. *International Journal of Computational Fluid Dynamics*, 22:241–257, 2008.
- [7] Mark W. Jones, J. Andreas Bærentzen, and Milos Sramek. 3d distance fields: A survey of techniques and applications. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):581–599, 2006.
- [8] I. Kataoka. Local instant formulation of two-phase flow. *Int. J. Multiphase Flow*, 12(5):745–758, 1986.

- [9] D. Lacanette, S. Vincent, A. Sarthou, P. Malaurent, and J.P. Caltagirone. An eulerian-lagrangian method for the numerical simulation of incompressible convection flows interacting with complex obstacles: Application to the natural convection in the lascaux cave. *International Journal of Heat and Mass Transfer*, 52:2528–2542, 2009.
- [10] Metin Muradoglu and Arif Doruk Kayaalp. An auxiliary grid method for computations of multiphase flows in complex geometries. *Journal of Computational Physics*, 214(2):858 – 877, 2006.
- [11] Carlos J. Ogayar, Rafael J. Segura, and Francisco R. Feito. Point in solid strategies. *Computers & Graphics*, 29(4):616–624, 2005.
- [12] Claude Puech and Hussein Yahia. Quadtrees, octrees, hyperoctrees: a unified analytical approach to tree data structures used in graphics, geometric modeling and image processing. In *SCG '85: Proceedings of the first annual symposium on Computational geometry*, pages 272–280, New York, NY, USA, 1985. ACM.
- [13] William J. Rider and Douglas B. Kothe. Reconstructing volume tracking,. *Journal of Computational Physics*, 141(2):112 – 152, 1998.
- [14] F. Roman, E. Napoli, B. Milici, and V. Armenio. An improved immersed boundary method for curvilinear grids. *Computers & Fluids*, 38(8):1510 – 1527, 2009.
- [15] A. Sarthou, S. Vincent, J.-P. Caltagirone, and P. Angot. Eulerian-Lagrangian grid coupling and penalty methods for the simulation of multiphase flows interacting with complex objects. *International Journal for Numerical Methods in Fluids*, 56(8):1093–1099, 2008.
- [16] S. Shin and D. Juric. Modelling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity. *Journal of Computational Physics*, 180:427–470, 2002.
- [17] M. Sussman, P. Smereka, and S.J. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114:146–159, 1994.
- [18] Joe F. Thompson. General curvilinear coordinate systems. *Applied Mathematics and Computation*, 10-11:1 – 30, 1982.
- [19] S. Vincent and J.-P. Caltagirone. Efficient solving method for unsteady incompressible interfacial flow problems. *International Journal for Numerical Methods in Fluids*, 30(6):795–811, 1999.
- [20] S. Vincent, A. Sarthou, J.-P. Caltagirone, F. Sonilhac, P. Février, C. Mignot, and G. Pianet. Augmented Lagrangian and penalty methods for the simulation of two-phase flows interacting with moving solids. Application to hydroplaning flows interacting with real tire tread patterns. *Under revision, Journal of Computational Physics*.

- [21] Stéphane Vincent, Guillaume Balmigère, Jean-Paul Caltagirone, and Erick Meillot. Eulerian-lagrangian multiscale methods for solving scalar equations - application to incompressible two-phase flows. *Journal of Computational Physics*, 229(1):73 – 106, 2010.
- [22] D. Youngs. *Time-dependent multimaterial flow with large fluid distortion*. K. W. Morton and M. J. Baines (eds), Academic Press, New-York, 1982.