

Tracking fronts in one and two-phase incompressible flows using an Adaptive Mesh Refinement approach

Stephanie Delage^{a,*}, Stephane Vincent^a,
Jean-Paul Caltagirone^a

^a*Laboratoire TREFLE, UMR 8508, site ENSCPB
Universite de science Bordeaux 1,
16 avenue PeyBerland
33607 Pessac, France.*

Abstract

Numerical computation is an essential tool for describing multi-phase and multi-scale flows accurately. One possibility consists in using very fine monogrids to obtain accurate solutions. However, this approach is very costly in time and memory size. As an alternative, an Adaptive Mesh Refinement method (*AMR*) has been developed in order to follow either interfaces in two-phase flows or concentration of a pollutant in one-phase flows. This method has also been optimized to reduce time and memory costs. Several 2D and 3D cases have been studied to validate and show the efficiency of the method.

Key words: Adaptive mesh refinement (*AMR*), implicit solving, conservative and non-conservative interpolations, numerical diffusion, interfaces, one and two-phase incompressible flows.

PACS: 47, 47.11. + *j*, 47.27. *Eq*, 47.55. *Kf*, 02.70. – *c*, 02.60. – *x*.

1 Introduction

One of the current difficulties in fluid mechanics is in describing multi-phase and multi-scale incompressible flows. It is difficult to simulate such flows in so far as they are complex and require important computer resources.

This paper focuses on tracking interfaces in two-phase incompressible flows

* Stephanie Delage, stephanie.delage@bordeaux.inra.fr

and inert species fronts in one-phase incompressible flows. The obtention of an accurate solution is essential for both flows. However the numerical diffusion introduced by most discretization schemes, such as *TVD* (Leveque, 1990), *WENO* (Liu, 1994) or *CIP* (Yabe, 2001) schemes, damages its accuracy. In the same way, discretization schemes such as the *VOF – PLIC* (Youngs, 1982) are responsible for an artificial interface fragmentation which damages the solution accuracy.

Given the fact that both numerical diffusion and fragmentation are linked to the mesh size, a very fine meshing can be used to reduce their effects. However, this is too expensive in memory size and time on standard fine Cartesian grids. As a result, we have developed an Adaptive Mesh Refinement (*AMR*) method based on (Berger, 1989) and (Vincent, 2000) to obtain accurate solutions with optimized memory size and time costs.

The physical requirements of incompressible flows involve the use of the Navier-Stokes equations (1), either coupled with the concentration species transport equation (2) in one phase flows or the interface transport equation (3) in two phase flows.

$$\vec{\nabla} \cdot \vec{V} = 0 \quad (1)$$

$$\rho \left(\frac{\partial \vec{V}}{\partial t} + (\vec{V} \cdot \vec{\nabla}) \vec{V} \right) = \rho \vec{g} - \vec{\nabla} p + \vec{\nabla} \cdot (\mu (\vec{\nabla} \vec{V} + \vec{\nabla}^T \vec{V}))$$

$$\frac{\partial S_p}{\partial t} + \vec{V} \cdot \vec{\nabla} (S_p) = \vec{\nabla} \cdot (D \vec{\nabla} (S_p)) \quad (2)$$

$$\frac{\partial C_i}{\partial t} + \vec{V} \cdot \vec{\nabla} (C_i) = 0 \quad (3)$$

with $\vec{V} = (u, w, v)$ the velocity vector, ρ the density, p the pressure, μ the viscosity, \vec{g} the gravity vector, $\vec{\nabla}^T$ the transpose vector of $\vec{\nabla}$, S_p the concentration and D the diffusion coefficient. C_i is the phase function repairing *fluid_i*. As a definition, we state $C_i = 1$ in *fluid_i* and $C_i = 0$ elsewhere. The interface between two fluids is defined by $C_i = 0.5$.

The equations are approximated by the Finite Volumes method described in (Patankar, 1980) on a staggered Cartesian grid (Harlow, 1965).

Concerning the Navier-Stokes equations, a one order Euler scheme is used for time discretization, the inertia term is treated by a linearized implicit centered scheme and the viscous terms are discretized by using a second order implicit centered scheme (for more details, see (Patankar, 1980)). The coupling between velocity and pressure is treated by the augmented Lagrangian method described in (Peyret, 1983), (Fortin, 1982) and (Vincent, 2004). They are solved in an implicit way by using a Bi-Conjugate Gradient Stabilized II algorithm (*BiCGStabII*) (Van Der Vorst, 1992) preconditioned by a Modified

and Incomplete LU Gauss factorization (Gustafsson, 1978).

Concerning equation (2), a one order classic scheme is used for time discretization. An explicit Total Variation Diminishing (*TVD*) scheme (Leveque, 1990) is used for the advection term. The diffusion term is solved implicitly, by using an iterative *Bi - CGStabII* algorithm with a *Jacobi* preconditioner (see (Gustafsson, 1978) or (Van Der Vorst, 1992)) is used.

We specify that for a two-phase incompressible flow, the one-fluid model introduced by (Kataoka, 1986) is used. A Volume Of Fluid (*VOF*) method is utilized to transport the interface (see (Hirt, 1981) for more details). Two methods have been tested to solve equation (3): the first one consists in discretizing the advection term of equation 3 by a *TVD* scheme described in (Hirsh, 1992), (Leveque, 1990) in *1D* and adapted to *2D* and *3D* for interface tracking by (Vincent, 1999) while the second one, called *VOF - PLIC* introduced by (Youngs, 1982), consists in using equation (3) to rebuild the interface geometrically and advect it using the Lagrangian approach.

Section 2 is dedicated to the principles of the *AMR* approach and more particularly to the connections, the extension operators and the refinement criteria. In section 3, we first evaluate the effects of the extension operators on the solution accuracy via a *2D* diffusion case. Then we show the efficiency of the developed method in conserving the solution accuracy and in minimizing the memory size and *CPU* time costs via some *2D* and *3D* scalar cases which consist in shearing a fluid disk. In section 4, the *AMR* approach is applied to both scalar and Navier-Stokes equations to evaluate its ability to deal with an academic *2D* case, called the Green Taylor vortex, and real two-phase incompressible flows related to *2D* dam break cases. Its efficiency in reducing the memory and *CPU* time costs are also studied. Some conclusions and perspectives are given in section 5.

2 Principle of the *AMR* method

In this subsection, we give a reminder of the essential points of the *AMR* approach described in (Delage, 2006). The current method is an improvement on the *OCLM* method developed by (Vincent, 2000), which was based on the concepts of (Berger, 1989).

This approach consists in generating fine grids of level G_L from a coarse one of level G_{L-1} , $L \in [1, niv]$ (where *niv* is the number of refinement levels), by means of a refinement criterion (see subsection 2.3 for more details). We specify that only a pressure node P_{L-1} of level G_{L-1} can generate an *AMR* cell, provided that P_{L-1} verifies the refinement criterion. This *AMR* cell is composed of a pressure fine grid and a velocity fine grid per component, as shown in figure 2. An odd cutting is required to ensure a consistent connection between *AMR* cells of a same refinement level (a three cutting has been

chosen in figures 1 and 2).

As explained in (Delage, 2006), the solution Φ_L on pressure and velocity nodes belonging to the limits of the *AMR* cell (see figure 2) of level G_L is calculated by means of an extension operator, namely non-conservative interpolations Q_1 , Q_2 or Q_3 . The solution Φ_L on pressure and velocity nodes belonging to the interior of the *AMR* cell (see figure 2) of level G_L is calculated thanks to the discretized equations.

The first essential point of this approach is its implicitation: it consists in introducing the interpolation coefficients inside the discretization matrix so as to solve every node in an implicit way and in only one step. More details can be found in (Delage, 2006).

Given that the solution Φ_L calculated on the nodes of an *AMR* cell of level G_L is more accurate than the solution Φ_{L-1} calculated on its father node of level G_{L-1} , a Full Weighted Control Volume (*FWCV*) method, introduced by (Hackbush, 1984), is used as a restriction operator. The solution on the father node is calculated from formula (4), where Ω_{father} is the control volume associated to the father node, $\Omega_{son(i)}$ is the control volume associated to the son node i , $i \in [1, N_{son}]$ and N_{son} the number of son nodes. *raf* is the refinement coefficient and *dim* corresponds to the space dimension. An illustration is given in figure 1 with *raf* = 3 and *dim* = 2.

$$\int_{\Omega_{father}} \Phi_{L-1} d\Omega \sim \sum_{i=1}^{N_{son}} \int_{\Omega_{son(i)}} \Phi_L d\Omega. \quad (4)$$

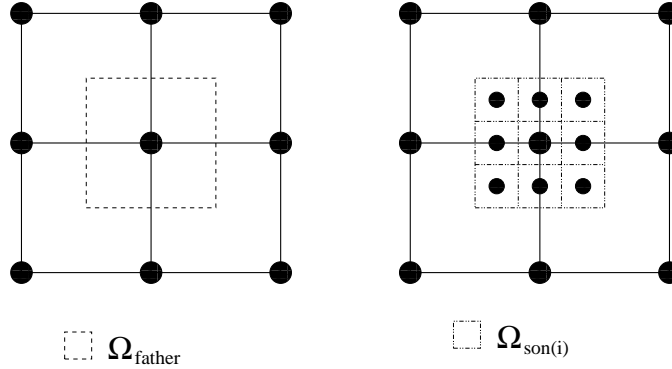


Fig. 1. Enrichment of a pressure node by means of a *FWCV* method as a restriction operator.

The second essential point of the method is the connections between *AMR* cells enabling transmission of information from cell to cell, which improves the accuracy of the solution. Given that this principle is an improvement on the method used in the description given by (Delage, 2006), the following paragraph is dedicated to this principle. When *AMR* cells of level G_L are

created from nodes of level G_{L-1} , pressure nodes $P_L \in G_L$ can generate *AMR* cells of level G_{L+1} , provided they belong to the interior of an *AMR* cell.

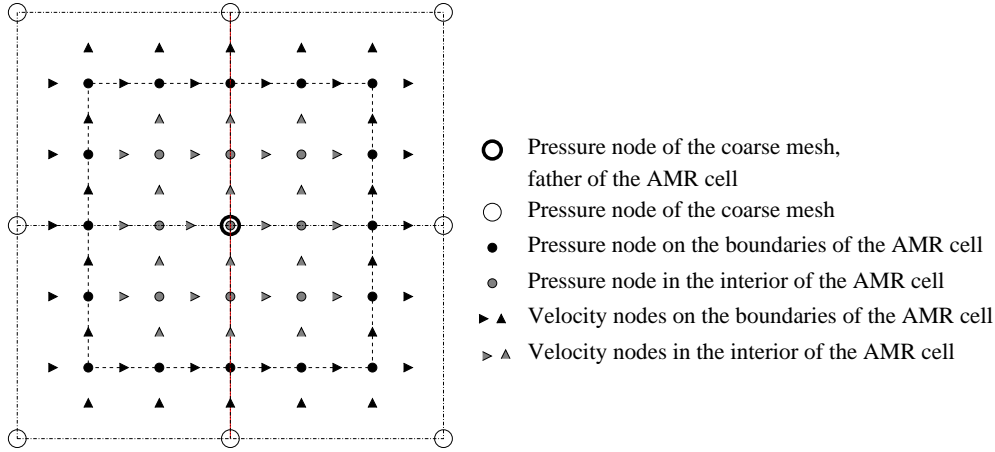


Fig. 2. Composition of an *AMR* cell.

2.1 Principle of connections

The purpose of this principle is to minimize the number of interpolated nodes to improve the accuracy of the solution. We take the example of two adjoining pressure nodes of level G_{L-1} which generate *AMR* cells. Figure 3 shows there is an overlapping. Each grey diamond pressure node of the left *AMR* cell has an east pressure neighbour belonging to the right boundary of the same *AMR* cell. However, the grey circle pressure nodes of the right *AMR* cell have the same coordinates as the east pressure neighbours of the grey diamond pressure nodes. As a result, each grey circle pressure node of the right *AMR* cell becomes the new east pressure neighbour of a diamond grey pressure node of the left *AMR* cell. In the same way, each grey diamond pressure node becomes the west pressure neighbour of a grey circle pressure node of the right *AMR* cell. Moreover, the pressure nodes belonging to the overlapping area and to the right (respectively left) boundary of the left (respectively right) *AMR* cell are no longer solved. However they retain their identity in the *AMR* tree structure (see figure 4) to make the refinement-derefinement dynamic management possible. The solution Φ_L is obtained by a non-conservative interpolation procedure on the black circle nodes.

Moreover, the connection procedure is performed by using the tree *AMR* structure of figure 4. If we take the example of figure 3, each grey diamond pressure node of the left *AMR* cell can be connected if and only if the west neighbour of the grey diamond pressure nodes father has created an *AMR* cell. This test is true in figure 3, therefore the grey diamond nodes can be connected to the grey circle pressure nodes.

The approach of connections can be expanded to several adjoining pressure

nodes of level G_{L-1} . Those which belong to the limits of an *AMR* cell as well as the overlapping area are not treated and those belonging to the interface $G_{L-1} - G_L$ are treated by using a non-conservative interpolation procedure. The solution Φ_L is calculated via the discretized equations on the pressure nodes belonging to the interior of the *AMR* cells.

We specify that level G_L is considered as a single grid for the solving whereas each *AMR* cell conserves its identity to make the refinement-derefinement procedure possible.

A similar approach is adopted to connect velocity nodes of *AMR* cells and to extend the connections procedure to $3D$.

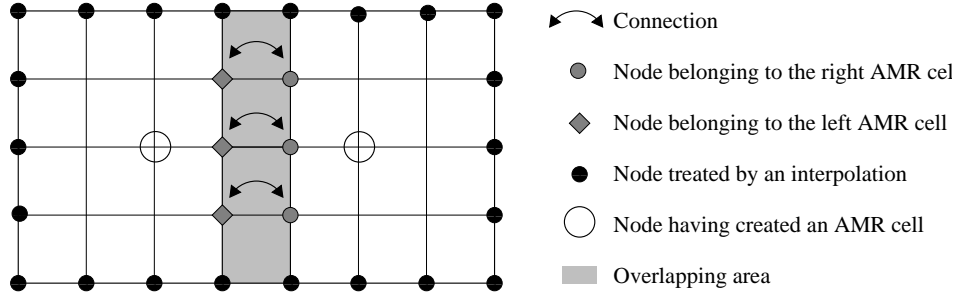


Fig. 3. Connections procedure of two *AMR* cells for the fine pressure grids in $2D$.

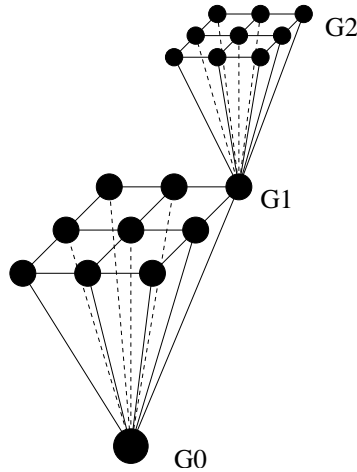


Fig. 4. Tree structure of several refinement levels with a refinement coefficient $raf = 3$.

2.2 Extension operators

Two kinds of extension operators have been developed in our *AMR* algorithm, depending on the treated problem. The first one is a non-conservative interpolation procedure whereas the second one is a conservative interpolation.

2.2.1 Non-conservative interpolation

The finite elements of Lagrange was chosen to build the interpolation functions. It consists in approximating the unknown by a polynomial of degree $n \in [1, 3]$, $\mathcal{P} \in Q_n = \text{vect}\{x^l z^m\} \forall (l, m) \in [1, n]$ (see (Delage, 2006) and (Delage Santacreu, 2006) for more details).

A fourth non-conservative interpolation, called binary interpolation, was implemented only for unknowns belonging to the pressure grid, such as the concentration S_p or the phase function C_i of *fluid_i*. Let Φ be this scalar unknown and P_{L-1} be a pressure node of level G_{L-1} which has generated an *AMR* cell of level G_L , $L \in [1, niv]$. For all pressure nodes P_L belonging to this cell, the binary interpolation consists in imposing: $\Phi(P_L) = 1$ if $\Phi(P_{L-1}) \geq 0.5$ and $\Phi(P_L) = 0$ if $\Phi(P_{L-1}) < 0.5$.

2.2.2 Conservative interpolation

This interpolation has been developed for the unknowns belonging to the pressure grid. The interesting results obtained with this procedure have led us to work on the adaptation of this interpolation to the velocity grids. This work is in progress.

The conservative interpolation developed here is based on the principle of the Flux Interface Correction (*FIC*) method introduced by (Angot, 1992). This method consists in writing a relation which ensures the continuity of fluxes through the interface $G_L - G_{L-1}$, $L \in [1, niv]$, in order to link the nested grids. The central idea of the *FIC* method and the conservative interpolation procedure explained in (Trottenberg, 2001) have been coupled in order to adapt them to the staggered cartesian grid of (Harlow, 1965).

We consider a pressure node P_{L-1} of level G_{L-1} . Its West (respectively East, South, North) pressure neighbour is noted W_{L-1} (respectively E_{L-1} , S_{L-1} , N_{L-1}), $L \in [1, niv]$ (see figure 5). We assume the node P_{L-1} has created an *AMR* cell of level G_L . The refinement coefficient is $raf = 3$. The pressure nodes of level G_{L-1} (respectively G_L) are represented by circles (respectively points) in figure 5. The pressure grids are represented by solid lines and the control volumes associated to pressure nodes are represented by dash-dot-dot lines. We note Ω_{P_L} the control volume associated to the node P_L , Σ_{W,P_L} (respectively Σ_{E,P_L} , Σ_{S,P_L} , Σ_{N,P_L}) the West (respectively East, South, North) face of Ω_{P_L} and $(u_{\Sigma_{I,P_L}}, w_{\Sigma_{I,P_L}})$, $I \in \{W, E, S, N\}$ the velocity on the centre of the face Σ_{I,P_L} .

Let Φ be the scalar unknown, solution of the advection-diffusion equation

$$\frac{\partial \Phi}{\partial t} + (\vec{V} \cdot \vec{\nabla}) \Phi = \vec{\nabla} \cdot (K \vec{\nabla} \Phi) \quad (5)$$

An implicit time discretization of equation (5) gives

$$\frac{\Phi^{k+1} - \Phi^k}{\Delta t} + (\vec{V}^k \cdot \vec{\nabla})\Phi^{k+1} = \vec{\nabla} \cdot (K \vec{\nabla} \Phi^{k+1}) \quad (6)$$

Where k is the time iteration and Δt the time step. The conservative form of equation (6) under the assumption $\vec{\nabla} \cdot \vec{V}^k = 0$ is

$$\frac{\Phi^{k+1} - \Phi^k}{\Delta t} + \vec{\nabla} \cdot (\Phi^{k+1} \vec{V}^k) = \vec{\nabla} \cdot (K \vec{\nabla} \Phi^{k+1}) \quad (7)$$

For a more comprehensible approach, we put $K = 0$ and we assume space steps are constant. The aim is to determine the unknown Φ on a node P_L . By applying the finite volume method to the control volume Ω_{P_L} (hachured area in figure 5) centered in P_L to equation (7), we can write

$$\int_{\Omega_{P_L}} (\Phi^{k+1} - \Phi^k) d\Omega_{P_L} + \Delta t \int_{\Omega_{P_L}} \vec{\nabla} \cdot (\Phi^{k+1} \vec{V}^k) d\Omega_{P_L} = 0 \quad (8)$$

The space step of a node P_L along with the horizontal (respectively vertical) axis x (respectively z) is noted Δx_L (respectively Δz_L). \vec{n} is a unit vector normal to a considered face of the control volume. By applying the Green formula to equation (8), we obtain

$$\frac{1}{\Delta t} (\Phi_{P_L}^{k+1} - \Phi_{P_L}^k) \Delta x_L \Delta z_L + F_{E,P_L} + F_{W,P_L} + F_{N,P_L} + F_{S,P_L} = 0 \quad (9)$$

where $F_{I,P_L} = \int_{\Sigma_{I,P_L}} \Phi^{k+1} \vec{V}^k \cdot \vec{n} d\Sigma_{I,P_L}$, is the advection flux of Φ through the face Σ_{I,P_L} , $I \in \{N, S, E, W\}$.

We note in figure 5, that the west face Σ_{W,P_L} coincides with the East face $\Sigma_{E,W_{L-1}}$ and is on the interface of levels $G_L - G_{L-1}$. By assuming there is a continuity of the fluxes through the interface $G_L - G_{L-1}$, we can write the inward flux F_{W,P_L} is equal to $\frac{1}{raf}$ of the outward flux $F_{E,W_{L-1}}$, i.e :

$$F_{W,P_L} = -\frac{1}{raf} \frac{\Phi_{W_{L-1}}^{k+1} + \Phi_{P_{L-1}}^{k+1}}{2} u_{\Sigma_{E,W_{L-1}}}^k \Delta z_{L-1} \quad (10)$$

The fluxes F_{E,P_L} , F_{N,P_L} and F_{S,P_L} are determined using the classic method because the faces Σ_{E,P_L} , Σ_{S,P_L} and Σ_{N,P_L} do not belong to the interface $G_L - G_{L-1}$. If a centered scheme is applied, we find

$$F_{E,P_L} = \frac{\Phi_{E_L}^{k+1} + \Phi_{P_L}^{k+1}}{2} u_{\Sigma_{E,P_L}}^k \Delta z_L \quad (11)$$

$$F_{N,P_L} = \frac{\Phi_{N_L}^{k+1} + \Phi_{P_L}^{k+1}}{2} w_{\Sigma_{N,P_L}}^k \Delta x_L \quad (12)$$

$$F_{S,P_L} = -\frac{\Phi_{P_L}^{k+1} + \Phi_{S_L}^{k+1}}{2} w_{\Sigma_{S,P_L}}^k \Delta x_L \quad (13)$$

We inject (10-13) into equation (9). Then we can determine the contribution of each discretized term of Φ which represents the coefficients of the discretization matrix and these of the second member. Hence it is possible to calculate $\Phi_{P_L}^{k+1}$ thanks to an implicit solver.

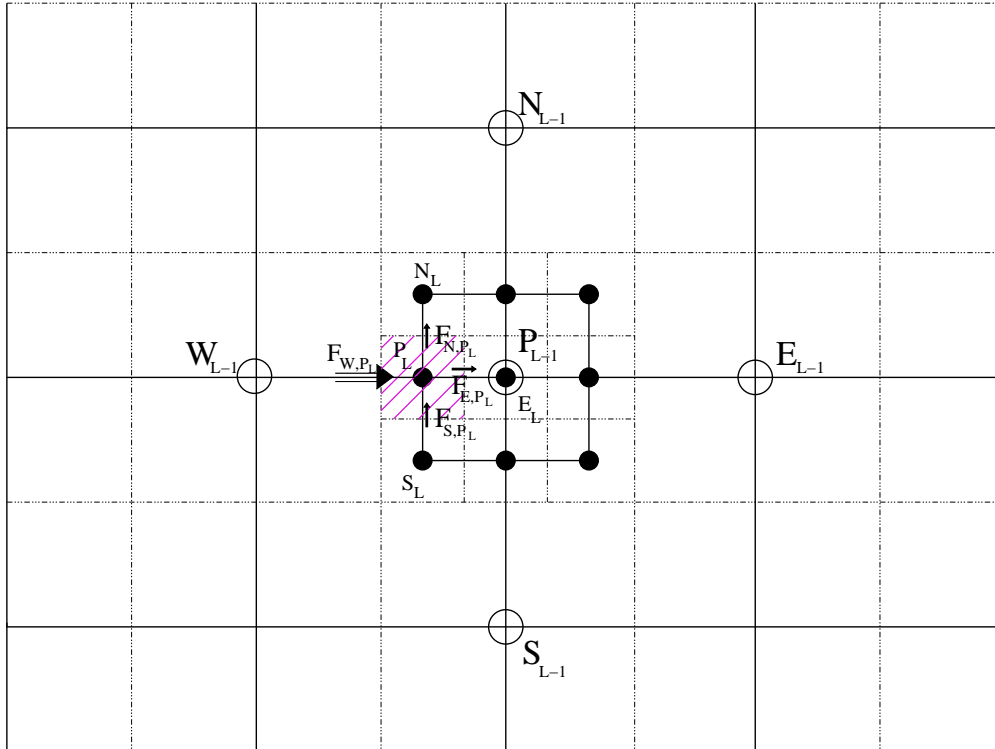


Fig. 5. Principle of the conservative interpolation on nodes of the pressure grid.

2.3 Refinement criteria

Several kinds of refinement criteria have been developed to manage this moving mesh strategy efficiently. A first method consists in calculating a posteriori errors to refine a mesh (Berger, 1989). A second one, recently developed, consists in coupling the solution accuracy and the maximum memory cost accepted (Bellenger, 2005). We have chosen to use the physical variables as refinement criteria as in (Vincent, 2000).

Here, the physical variables are the concentration S_p in order to track the

species in a monophasic incompressible flow and the phase function C_i to track the interface in a two-phase incompressible flow. For both S_p and C_i , several parameters can be evaluated, namely their magnitude or their variation relative to the distance. For each parameter, a threshold is chosen to manage the refinement. Let Φ represent S_p or C_i .

The criterion based on the value of Φ consists in imposing a minimum threshold Val_{min} and a maximum threshold Val_{max} . Given a node $P_{L-1} \in G_{L-1}$, $l \in [1, niv]$, if $\Phi(P_{L-1}) \in [Val_{min}, Val_{max}]$, then the node P_{L-1} generates an *AMR* cell. This criterion is efficient for slow variations of Φ as it is explained in (Delage Santacreu, 2006). So it is not adapted to track fronts.

Now, we describe the approach of the criterion based on the variations of Φ relative to the distance. Let P_{L-1} be a pressure node of level G_{L-1} . As shown in figure 6 left, its neighbours are noted W_{L-1} , E_{L-1} , S_{L-1} , N_{L-1} , SW_{L-1} , SE_{L-1} , NW_{L-1} and NE_{L-1} . The distance between two nodes, P_{L-1} and W_{L-1} for instance, is noted $dist(P_{L-1} - W_{L-1})$. The procedure consists in imposing a minimum threshold Grd_{min} and evaluating the variations of Φ relative to the distance along with every grid direction:

$$\begin{aligned} & \left| \frac{\Phi(S_{L-1}) - \Phi(P_{L-1})}{dist(S_{L-1} - P_{L-1})} \right|, \left| \frac{\Phi(P_{L-1}) - \Phi(N_{L-1})}{dist(P_{L-1} - N_{L-1})} \right|, \left| \frac{\Phi(E_{L-1}) - \Phi(P_{L-1})}{dist(E_{L-1} - P_{L-1})} \right|, \\ & \left| \frac{\Phi(P_{L-1}) - \Phi(W_{L-1})}{dist(P_{L-1} - W_{L-1})} \right|, \left| \frac{\Phi(P_{L-1}) - \Phi(NW_{L-1})}{dist(P_{L-1} - NW_{L-1})} \right|, \left| \frac{\Phi(SE_{L-1}) - \Phi(P_{L-1})}{dist(SE_{L-1} - P_{L-1})} \right|, \\ & \left| \frac{\Phi(P_{L-1}) - \Phi(NE_{L-1})}{dist(P_{L-1} - NE_{L-1})} \right|, \left| \frac{\Phi(SW_{L-1}) - \Phi(P_{L-1})}{dist(SW_{L-1} - P_{L-1})} \right| \end{aligned}$$

If the maximum value of the previous calculated variations is superior or equal to Grd_{min} , then the node P_{L-1} creates an *AMR* cell.

A demonstration is made for a 1D problem. We consider a domain $\Omega = [0, 1]$ and a function Φ verifying $\forall x \in \Omega$, if $x \in [0.2, 0.4]$ then $\Phi(x) = 1$, else $\Phi(x) = 0$. We represent in figure 6 right this function as well as the nodes $P_{L-1} \in G_{L-1}$. By assuming a correct value of $Grd_{min} > 0$ has been chosen, we note the variations of Φ on the black nodes of figure 6 right are equal to zero whereas its variations on the grey nodes are different from zero. Therefore the grey nodes of figure 6 right, situated on and close to the front, can generate *AMR* cells. This means that this criterion is efficient in determining the front accurately.

We specify that the derefinement procedure is the same as the refinement procedure. Moreover, we chose the same criterion for both refinement and derefinement management.

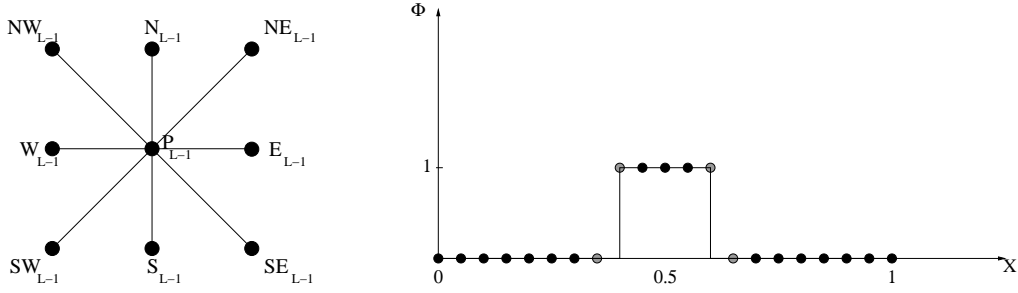


Fig. 6. Nodes of level G_{L-1} used to calculate the variations of Φ on P_{L-1} relative to the distance (left). Representation of the function Φ and detection of the nodes liable to create *AMR* cells (right).

3 Application of the method to scalar equations

3.1 Two dimensional diffusion case

In this subsection, we aim at determining the effects of the non-conservative and conservative interpolations on the solution when concentration gradients are present everywhere in the domain Ω .

We chose the same conditions as those described by (Whitaker, 1983) for a conduction case in order to have an analytical solution for the steady state. The difference is that the temperature in (Whitaker, 1983) has been replaced by the species concentration S_p .

Let Ω be a two dimensional domain, $H = 1$ meter in height and $b = 0.5$ meters in width. The horizontal direction \vec{x} is noted x whereas the vertical is noted z . Equation (2) is considered with $\vec{V} = \vec{0}$. The diffusion coefficient is equal to $D = 1m^2.s^{-1}$. A constant concentration $S_{p_0} = 10g.m^{-3}$ is imposed on the left, right and lower boundaries. As for the boundary on the top of the domain, the following concentration profile S_{p_f} is imposed with $S_{p_1} = 100g.m^{-3}$ (see figure 7 left):

$$S_{p_f} : [0, b] \times [H, H] \rightarrow \mathfrak{R}$$

$$(x, z) \mapsto S_{p_0} - \frac{16}{b^4}(S_{p_0} - S_{p_1})x^2(x - b)^2 \quad (14)$$

Using formula (14), the analytical solution $S_{p_{an}}$ can be written as (see also figure 7 (right)):

$$S_{p_{an}} : \Omega \rightarrow \mathfrak{R} \quad (15)$$

$$(x, z) \mapsto S_{p_0} + 16(S_{p_1} - S_{p_0}) \sum_{n=1}^{\infty} \frac{1 - (-1)^n}{(n\pi)^3 \sinh(\frac{n\pi H}{b})} \sin(\frac{n\pi x}{b}) \sinh(\frac{n\pi z}{b})$$

Three cases are studied and then compared to achieve our objective.

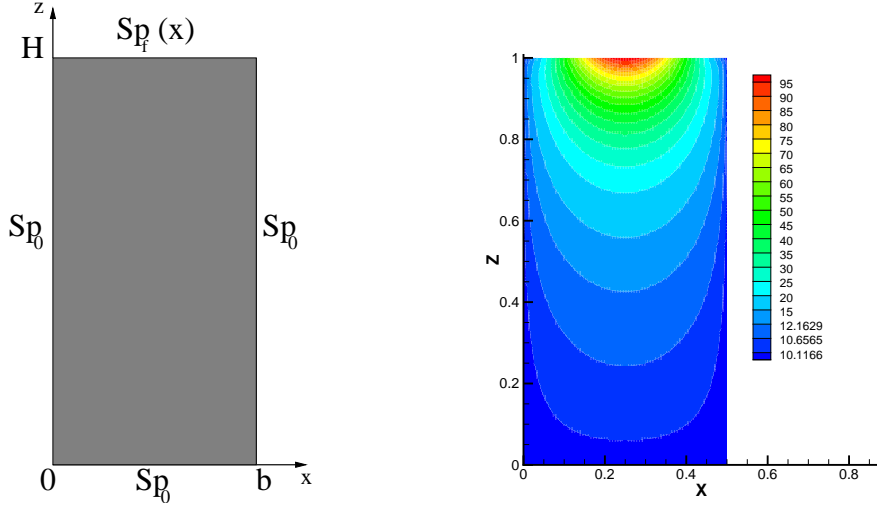


Fig. 7. Description of the diffusion case (left). Analytical concentration distribution for the steady state (right).

- *case*₁: study of the concentration distribution using a classic mesh.
- *case*₂: study of the concentration distribution using one level of refinement in the area $[0, 0.5][0.8, 1]$, and a refinement coefficient $raf = 3$. The solution is regular, therefore a Q_2 interpolation is used for the nodes belonging to the interface $G_0 - G_1$, (Delage, 2006). A *FWCV* injection has been chosen as a restriction operator (Hackbush, 1984).
- *case*₃: this case is the same as *case*₂. The difference comes from the use of a conservative interpolation (see subsection 2.2) instead of the Q_2 interpolation.

Let $S_{p_{num}}$ be the numerical concentration obtained when the steady state is reached. The number of nodes needed to calculate $S_{p_{num}}$ is noted N . For *case*₁, N represents the number of the nodes of the monogrid G and the error ϵ in the L^2 norm can be defined as follows:

$$\begin{aligned} \epsilon &= \frac{\|S_{p_{num}} - S_{p_{an}}\|_2}{\|S_{p_{an}}\|_2} \\ \epsilon &= \frac{[\int_{\Omega} |S_{p_{num}}(x, z) - S_{p_{an}}(x, z)|^2 dx dz]^{\frac{1}{2}}}{[\int_{\Omega} |S_{p_{an}}(x, z)|^2 dx dz]^{\frac{1}{2}}} \\ \epsilon &\sim \frac{[\sum_{i=1, N} |S_{p_{num}_i} - S_{p_{an}_i}|^2 \Delta x_i \Delta z_i]^{\frac{1}{2}}}{[\sum_{i=1, N} |S_{p_{an}_i}|^2 \Delta x_i \Delta z_i]^{\frac{1}{2}}}. \end{aligned}$$

For *case*₂ and *case*₃, we define $C_{amr} = \{M \text{ belonging to the interior of an } AMR \text{ cell}\}$ and $V_{amr} = \{M \in G_0 \cup C_{amr} \text{ such as } M \text{ has not created an } AMR\}$

cell }. Here, N represents the number of nodes so that $M \in V_{amr}$. Three errors ϵ_{G_0} , ϵ_{G_1} and $\epsilon_{composite}$ are defined as

$$\begin{aligned}\epsilon_{G_0} &\sim \frac{[\sum_{i \in G_0} |S_{p_{num_i}} - S_{p_{an_i}}|^2 \Delta x_i \Delta z_i]^{\frac{1}{2}}}{[\sum_{i \in G_0} |S_{p_{an_i}}|^2 \Delta x_i \Delta z_i]^{\frac{1}{2}}}, \\ \epsilon_{G_1} &\sim \frac{[\sum_{i \in G_1 \cap C_{amr}} |S_{p_{num_i}} - S_{p_{an_i}}|^2 \Delta x_i \Delta z_i]^{\frac{1}{2}}}{[\sum_{i \in G_1 \cap C_{amr}} |S_{p_{an_i}}|^2 \Delta x_i \Delta z_i]^{\frac{1}{2}}}. \\ \epsilon_{composite} &\sim \frac{[\sum_{i \in V_{amr}} |S_{p_{num_i}} - S_{p_{an_i}}|^2 \Delta x_i \Delta z_i]^{\frac{1}{2}}}{[\sum_{i \in V_{amr}} |S_{p_{an_i}}|^2 \Delta x_i \Delta z_i]^{\frac{1}{2}}}.\end{aligned}$$

mesh G	N	ϵ
8 * 8	81	1.2998 10 ⁻²
16 * 16	289	4.2915 10 ⁻³
32 * 32	1089	1.2316 10 ⁻³
64 * 64	4225	3.2712 10 ⁻⁴
128 * 128	16641	8.3961 10 ⁻⁵
256 * 256	65536	2.1340 10 ⁻⁵

Table 1
Evolution of the error ϵ along with the number of nodes for *case*₁

Table 1 shows the evolution of the error ϵ when the number of the nodes N of the mesh G increases. We compare in table 2 the evolution of the different errors along with the number of nodes, when a conservative interpolation (*FIC*) or a non-conservative interpolation Q_2 is used. We note that ϵ_{G_0} is similar in the two cases whereas ϵ_{G_1} is lower in *case*₃ than in *case*₂.

Given that the error is linked to the number of nodes N by the formula $\epsilon = k(\sqrt{\frac{L}{N}})^p$ (ϵ corresponds to $\epsilon_{composite}$ for *case*₂ and *case*₃), where k is a constant, $L = \sqrt{Hb}$ and p the convergence order, we compare the convergence for the three cases. Figure 8 shows the line $\ln(\epsilon)$ along with $\ln(\frac{L}{N})$ for each case. *Case*₁ is represented by a dash-dot-dot line and square symbols, *case*₂ by a dash-dot line and triangular symbols, and *case*₃ by a solid line and circular symbols. We note in figure 8 that the lines for *case*₁ and *case*₃ are linear if the first value is not taken into account and the slopes are very close. This means that *case*₁ and *case*₃ give similar results in term of convergence. As for *case*₂, the absolute value of the slopes between two successive points increases with N , but hardly reach the slope of the straight line of *case*₁. So we can deduce that the method used in *case*₂ is less efficient than that of *case*₁ in terms of convergence. To confirm these preliminary results, we calculated the slopes p_1 ,

G_0	Interpolation	N	ϵ_{G_1}	ϵ_{G_0}	$\epsilon_{composite}$
16 * 16	<i>case₂</i>	907	9.61 10 ⁻⁴	3.8 10 ⁻²	1.29 10 ⁻³
	<i>case₃</i>	828	2.148 10 ⁻³	3.8 10 ⁻²	2.133 10 ⁻³
32 * 32	<i>case₂</i>	3188	5.762 10 ⁻⁴	1.48 10 ⁻²	6.561 10 ⁻⁴
	<i>case₃</i>	3029	5 10 ⁻⁴	1.48 10 ⁻²	5 10 ⁻⁴
48 * 48	<i>case₂</i>	6845	3.3 10 ⁻⁴	8.33 10 ⁻³	3.67 10 ⁻⁴
	<i>case₃</i>	6606	2.174 10 ⁻⁴	8.31 10 ⁻³	2.168 10 ⁻⁴
64 * 64	<i>case₂</i>	11878	2.09 10 ⁻⁴	5.498 10 ⁻³	2.306 10 ⁻⁴
	<i>case₃</i>	11559	1.211 10 ⁻⁴	5.48 10 ⁻³	1.208 10 ⁻⁴
96 * 96	<i>case₂</i>	26939	1 10 ⁻⁴	3.042 10 ⁻³	1.092 10 ⁻⁴
	<i>case₃</i>	26460	5.05 10 ⁻⁵	3.037 10 ⁻³	5.049 10 ⁻⁵

Table 2

Evolution of the errors ϵ_{G_0} , ϵ_{G_1} and $\epsilon_{composite}$ along with the number of nodes for *case₂* and *case₃*, when different extension operators are used. The coarse mesh G_0 is variable.

p_2 , p_3 and p_4 between two successive points for each case. These slopes are put in table 3. They show that for each case, the limit order of convergence is $p = 2$. This value is quickly reached in *case₁* since $p_2 = 2$. As it has been already noticed in figure 8, table 3 shows that *case₂* damages the convergence order in so far as the limit value seems to be $p = 2$ but $p_4 < 2$. Therefore, the use of fine mesh G_0 is essential for *case₂* to reach an order of convergence $p = 2$. Table 3 confirms the efficiency of the method used in *case₃* because the limit order of convergence is reached from p_1 .

So we can conclude the use of a conservative interpolation as an extension

	p_1	p_2	p_3	p_4
<i>case₁</i>	1.9	2	2	2
<i>case₂</i>	1	1.5	1.7	1.8
<i>case₃</i>	2.2	2.1	2.1	2.1

Table 3

successive local slopes of figure for the three cases.

operator (*case₃*) is more efficient than *case₂*, compared with *case₁*, in terms of convergence.

We are also interested in the ability of the method to conserve the species mass M defined as:

$$M = \int_{\Omega} S_p d\Omega \quad (16)$$

As a consequence, we decide to compare the mass M_{an} deduced from the analytical solution S_{pan} and the use of formula (16) with the mass M_{num}

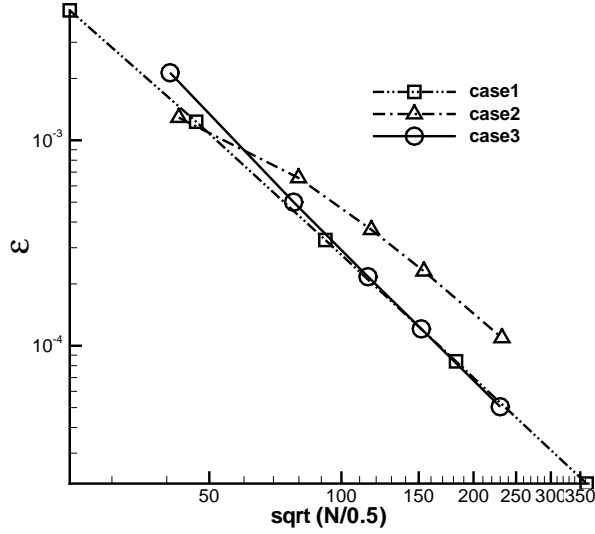


Fig. 8. Convergence study for *case1*, *case2* and *case3*.

calculated thanks to the numerical concentration field $S_{p_{num}}$ (steady state reached) and using formula (16). M_{an} is evaluated on finer and finer meshes until a constant value of M_{an} considered as a reference is found. We find a constant value $M_{an} = 8.9734g$ from a fine mesh $G = 400 * 400$. The mass M_{num} is evaluated for *case2* and *case3* on the coarse mesh $G_0 = 32 * 32$ when the steady state is reached. Table 4 shows that for *case2*, $M_{num} = 8.9811g$, that is to say an error between M_{an} and M_{num} of 0.09%. For *case3*, we obtain in table 4 $M_{num} = 8.9732g$, i. e. an error between M_{an} and M_{num} of 0.001%. Hence the error between M_{an} and M_{num} is 90 times higher in *case2* than in *case3*. This means that the mass is better conserved for *case3* than for *case2*. Nevertheless, the error ϵ_{G_0} obtained for $G_0 = 32 * 32$ in *case2* is the same as in *case3* (see table 2). We decide to analyse the error distribution field ϵ_0 , in Ω , defined as $\epsilon_0(x, z) = \frac{|S_{p_{num}}(x, z) - S_{p_{an}}(x, z)|}{|S_{p_{an}}(x, z)|}$, $\forall (x, z) \in G_0$ to understand why the mass conservation depends on the cases whereas the error ϵ_{G_0} is the same for both cases. Figure 9 compares ϵ_0 obtained on level G_0 in *case2* with this in *case3*. We notice ϵ_0 is different along with the treated case. This means that both extension operators give the same error ϵ_{G_0} to within about 10^{-2} , but they differ from their conservation properties.

	mesh G_0	$M_{num}(g)$	error
<i>case2</i>	$32 * 32$	8.9811	0.09%
<i>case3</i>	$32 * 32$	8.9732	0.001%

Table 4

Comparison of M_{num} with M_{an} for *case2* and *case3*.

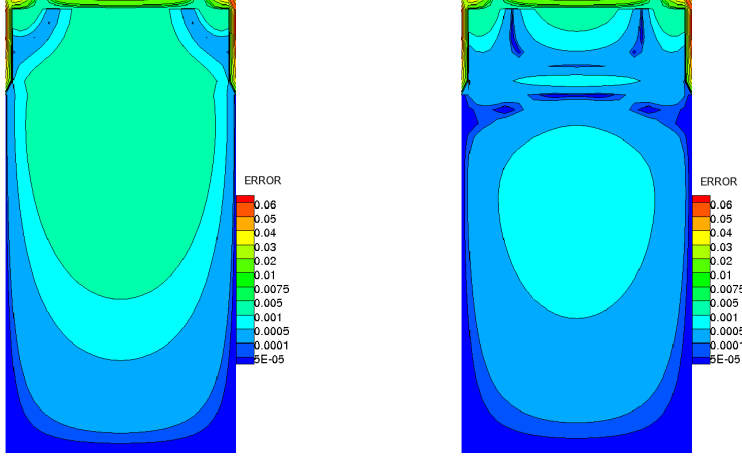


Fig. 9. Distribution of the error ϵ_0 on the coarse mesh $G_0 = 32 * 32$ with a Q_2 interpolation on the left and a conservative interpolation on the right. *FWCV* restriction operator.

3.2 Sheared disk in a two dimensional one-phase flow

A two dimensional square domain Ω of characteristic length $L = 0.1m$, initially full of a pure fluid (concentration equal to zero), is considered. A disk of a polluted fluid of concentration $1g.m^{-3}$ is introduced in Ω . It is initially centered on $(x = 0.075m, z = 0.05m)$ and its radius is $r = 0.015m$ (see figure 10 left). The disk is sheared by a rotating velocity field (see figure 10 right) $\vec{V}(x, z) = (u(x, z), w(x, z))$ of averaged amplitude $V_m = 0.1m.s^{-1}$,

$$\begin{aligned} u(x, z) &= -V_m \cos\left(\frac{\pi(x - x_0)}{L}\right) \sin\left(\frac{\pi(z - z_0)}{L}\right) \\ w(x, z) &= V_m \sin\left(\frac{\pi(x - x_0)}{L}\right) \cos\left(\frac{\pi(z - z_0)}{L}\right) \end{aligned} \quad (17)$$

The velocity field is continuous but not the concentration field. The governing equation is (2) with a diffusion coefficient $D = 10^{-8}m^2.s^{-1}$.

Given the results obtained in subsection 2.3, we use the criterion based on the variation of S_p relative to the distance. The efficiency of the method in reducing the numerical diffusion has been demonstrated in (Delage, 2006) for a similar case. As a result, we restrict the study to the efficiency of the *AMR* approach in preserving a satisfying accuracy of the solution as well as optimizing time and memory costs.

3.2.1 Accuracy of solutions

As it has been shown in (Delage, 2006), the Q_1 interpolation used as an extension operator can avoid the oscillation phenomenon. However, this operator

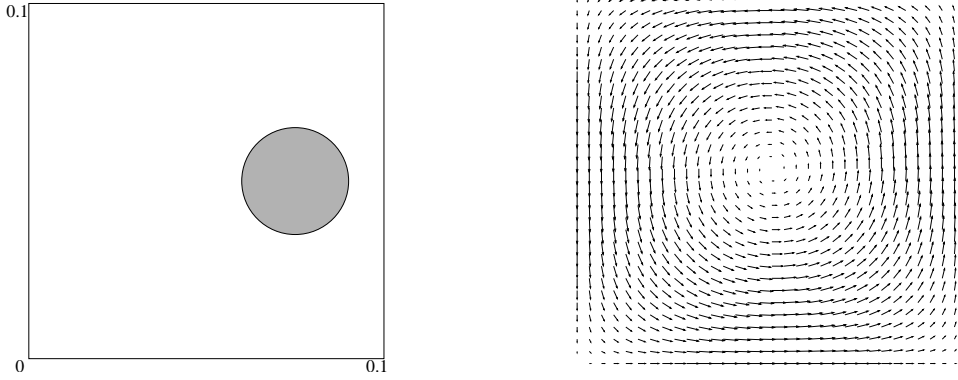


Fig. 10. Polluted fluid disk in a pure fluid (left). Imposed velocity field (right).

introduces a small numerical diffusion when it is used for nodes too close to the front. As a consequence, a more appropriate non-conservative interpolation, called binary interpolation, was used in this case.

We compare in figure 11 left the concentration contour line $S_p = 0.5g.m^{-3}$ of the sheared disk, for binary and Q_1 interpolations. The calculation was made on a coarse mesh $G_0 = 20 * 20$ and two refinement levels. The time step is $\Delta t = 0.01s$ and verifies the CFL condition $max(\frac{V_m \Delta t}{\Delta x_2}, \frac{V_m \Delta t}{\Delta z_2}) \leq 0.5$, where Δx_2 (respectively Δz_2) is the space step of level G_2 along with the axis x (respectively z). We can see in figure 11 left that the use of the Q_1 interpolation (solid line) adds an extra numerical diffusion on the head and the tail of the sheared disk, in relation to the use of the binary interpolation (dotted line). We specify that a similar numerical diffusion is observed when we compare the solution obtained on a coarse mesh $G_0 = 16 * 16$, two refinement levels and a Q_1 interpolation with the solution calculated on a single mesh $G = 144 * 144$. As for the binary interpolation, the solution $S_{p_{amr}}$ is similar to the solution $S_{p_{wamr}}$ obtained on a single mesh $G = 144 * 144$. Besides, the calculation of the error ϵ_{amr} defined as, $\forall P \in V_{amr}, \epsilon_{amr} = \frac{\|S_{p_{amr}} - S_{p_{wamr}}\|_2}{\|S_{p_{wamr}}\|_2}$ gives $\epsilon_{amr} = 6.17 * 10^{-5}$.

This means the use of the binary interpolation as an extension operator for S_p enables the AMR method to preserve an accurate representation of the concentration field even when S_p is strongly stretched or distorted, as physically observed in turbulent flows for instance.

As in subsection 4.1, we decide to analyse the convergence of the AMR approach. Given the fact that an analytical solution does not exist, a solution calculated on a very fine mesh $G = 1296 * 1296$ is chosen as a reference solution $S_{p_{ref}}$. By adopting the same approach as in subsection 4.1, we obtain a limit convergence order $p = 2$ with and without AMR . The accuracy and the convergence order of the standard schemes are not damaged by their non conforming AMR extension. Figures 12 and 13 show the efficiency of this method in reducing the numerical diffusion while the number of refinement levels increases. We specify the connections between the refinement levels are

not structured only for the graphic application.

In figure 14, we compare the concentration field obtained on a monogrid $G = 16 * 16$ at time $t = 8s$ with the one for the coarse mesh $G_0 = 16 * 16$ when three refinement levels and a *FWCV* injection are used. We note that the maximum value of S_p is about $0.28g.m^{-3}$ in the left figure. So the concentration field has been greatly diffused. In the right figure, the concentration field observed on $G_0 = 16 * 16$ has a maximum value $S_p = 0.8g.m^{-3}$. Therefore, it is less diffused than in the left figure. This result shows that the *FWCV* injection enriches S_p on the coarse mesh in such a way as to limit the numerical diffusion.

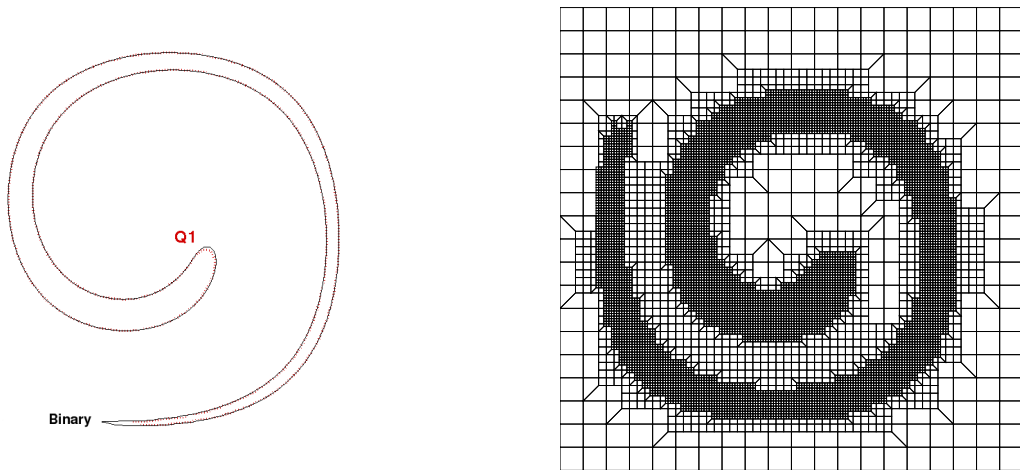


Fig. 11. Concentration contour line $S_p = 0.5g.m^{-3}$ obtained with a Q_1 and a binary interpolations at time $t = 8s$. Mesh used for the simulation on the right

3.2.2 Memory cost analysis

We compare the memory size needed to simulate the shearing of the disk during $8s$ using the *AMR* method with the memory size required for the same case without *AMR*. This study is essential to estimate the memory gains obtained with the approach.

We note mem_{amr} (respectively mem) the memory size (in *Mo*) required for the *AMR* simulation (respectively simulation without *AMR*), rap the ratio $rap = \frac{mem_{amr}}{mem}$. NBG_{niv} represents the number of nodes needed to calculate the solution with niv levels of refinement. Let KKT be the number of nodes needed for the simulation on a mesh G , without *AMR*. This mesh is equivalent to the finest grid used in the case with *AMR*. rgl is defined as $rgl = \frac{NBG_{niv}}{KKT}$. For the case with *AMR*, we use a coarse mesh $G_0 = 20 * 20$, a time step $\Delta t = 3.08 \cdot 10^{-4}s$ and simulate the sheared disk from one to three refinement levels. Table 5 shows the evolution of the previously defined parameters when

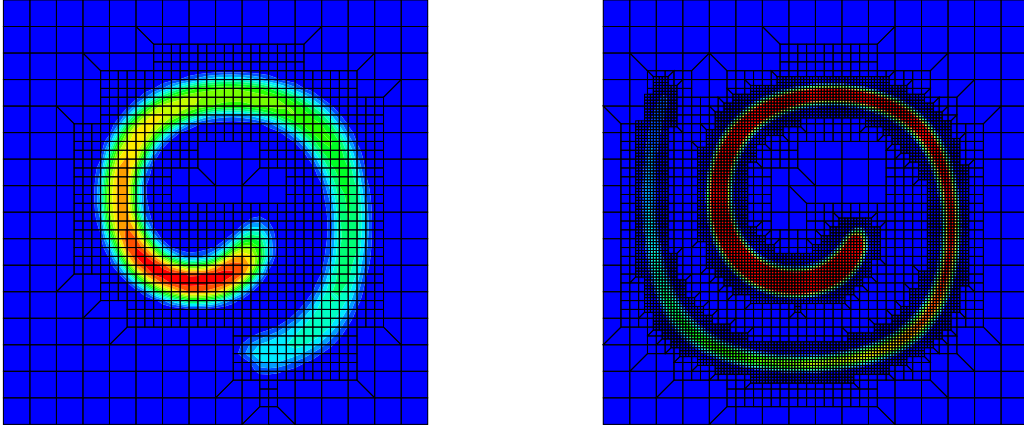


Fig. 12. Concentration obtained with a binary interpolation at time $t = 8s$ from a coarse mesh $G_0 = 16 * 16$. There is one refinement level on the left and two on the right.

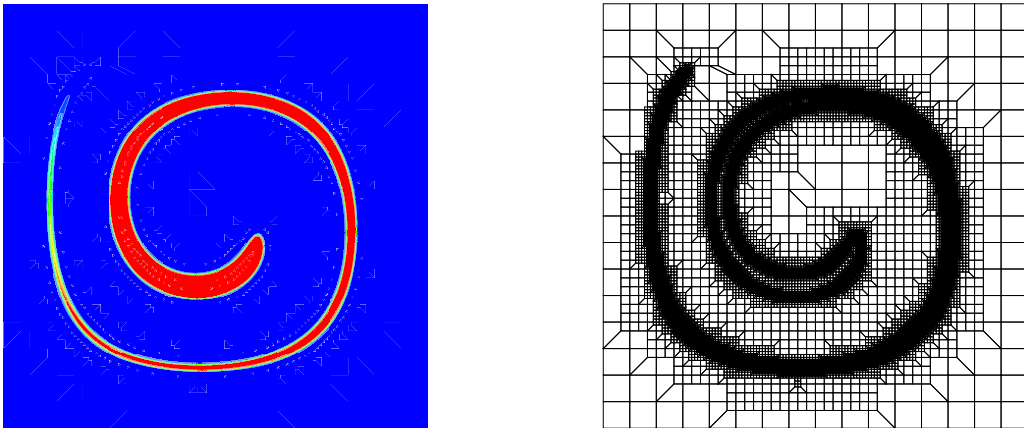


Fig. 13. Concentration field obtained with a binary interpolation at time $t = 8s$ from a coarse mesh $G_0 = 16 * 16$ and three refinement levels (left). Mesh associated to the concentration field (right).

the number of refinement levels increases for the *AMR* case as well as when the monogrid G becomes finer and finer for the case without *AMR*. We can see on table 5 that the ratio $rap > 1$ until two refinement levels. This means that the *AMR* approach is more expensive than a classic solving on a fine cartesian grid. For three refinement levels our approach is interesting because a memory size gain of 30% has been achieved with the *AMR* method, compared with the classic one. If we pay attention to the evolution of the ratio rgl , we realize rgl is linked to rap . We aim at evaluating a critical value of rgl , noted rgl_c . For this, we deem the *AMR* algorithm cost becomes interesting when the

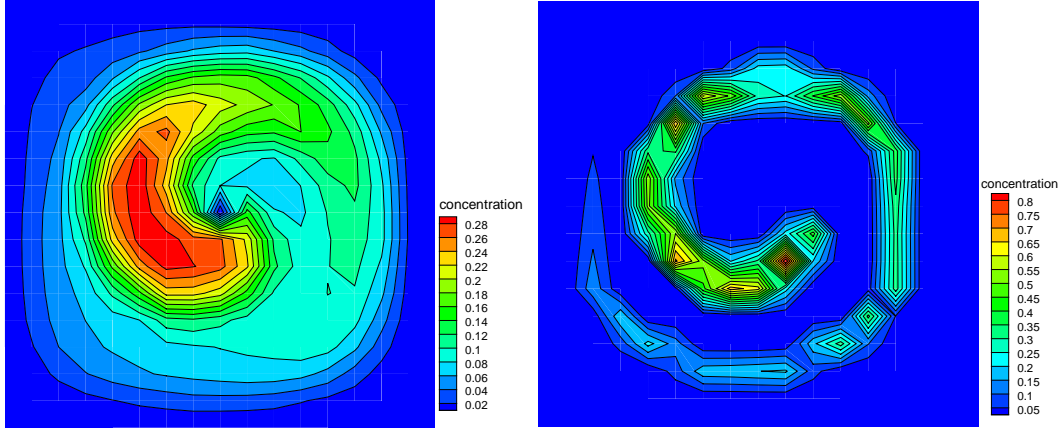


Fig. 14. Concentration field obtained with a binary interpolation at time $t = 8s$ on a monogrid $G = 16 * 16$ (left) and on the coarse mesh $G_0 = 16 * 16$ when three refinement levels are used (right).

minimum memory profit is superior or equal to 25%, i. e.: $rap_c = 0.75$. A linear regression between rap and rgl enables us to obtain a straight line of equation: $rap = 4.605 * rgl - 0.7376$ with a correlation coefficient equal to 0.995. We deduce $rgl_c = 32\%$ i. e. that this approach is efficient in minimizing the memory size cost as long as the number of nodes used with the *AMR* method is inferior to 32% of the nodes which would have been used with a similar fine monogrid. This is very satisfying in *2D*.

mesh	mem_{amr}	mem	NBG_{niv}	KKT	rap	rgl
$60 * 60$	18	7	2650	3721	2.6	0.71
$180 * 180$	37	34	14018	32761	1.1	0.43
$540 * 540$	82	281	61170	292681	0.3	0.21

Table 5

Memory size required to simulate the case of the sheared disk on different meshes.

3.2.3 CPU time cost analysis

The purpose of this subsection is to estimate the efficiency of the *AMR* method concerning the simulation time t_{amr} . Let t_{tot} be the average *CPU* time of simulation for one iteration. t_{tot} includes the refinement management time t_{ref} and the solving time t_{sol} . t_{ref} , t_{sol} and t_{tot} are defined as: $\frac{1}{NT} \sum_{i=1,NT} t_i$, where NT is the number of time iterations and t_i the considered *CPU* time per iteration.

Table 6 represents the different average times t_{ref} , t_{sol} and t_{tot} from one to three refinement levels. The coarse mesh is $G_0 = 20 * 20$ and the time step is

$\Delta t = 3.08 \cdot 10^{-4} s$. We have also calculated the ratios $\frac{t_{ref}}{t_{tot}}$ and $\frac{t_{sol}}{t_{tot}}$ in order to evaluate the distribution of the costs of refinement and solving in one iteration. We notice on table 6 that $\frac{t_{ref}}{t_{tot}}$ increases while $\frac{t_{sol}}{t_{tot}}$ decreases when the number of refinement levels increases.

For three refinement levels, the refinement management algorithm is more expensive in terms of *CPU* time than the solving algorithm. This can be explained by the fact that the equation (2) is easy to solve.

So we ask ourselves if the *CPU* time of simulation t_{amr} is smaller than that obtained without *AMR*, t_{wamr} . We note $rt = \frac{t_{amr}}{t_{wamr}}$.

As for the memory cost of the *AMR* method, the *CPU* time cost becomes interesting from three refinement levels ($rt = 0.75$ in table 7). The results are satisfying since a *CPU* time profit of 25% has been made for three refinement levels (see table 7). However, the *CPU* time cost depends on the complexity of the equation to solve. So we cannot estimate a priori the time gains.

number of levels	t_{ref}	t_{sol}	t_{tot}	$\frac{t_{ref}}{t_{tot}}$	$\frac{t_{sol}}{t_{tot}}$
1	0.002499	0.01193	0.02001	0.1249	0.5962
2	0.04609	0.07379	0.1343	0.3432	0.5494
3	0.4509	0.2476	0.7481	0.6027	0.3310

Table 6

Comparison of averaged *CPU* time at $NT = 26000$ for different refinement levels.

mesh	t_{amr}	t_{wamr}	rt
$60 * 60$	522.1	475.3	1.10
$180 * 180$	3495	3144	1.11
$540 * 540$	19460	25860	0.75

Table 7

Comparison of simulation time with and without *AMR*.

3.3 Sheared disk in a two dimensional two-phase flow

The aim of this subsection is to show the ability of the method to treat the same case as that of subsection 3.2 with two immiscible fluids, $fluid_1$ and $fluid_2$.

As a result, we consider the same situation as in subsection 3.2. The polluted fluid is replaced by $fluid_1$ and the pure fluid by $fluid_2$. Let C_i be the phase function of $fluid_i$, $i \in [1, 2]$. Equation (3) is solved to track the interface. We detail the *AMR* contribution for the *VOF - PLIC* method. As it is known, the *VOF - PLIC* method is efficient in obtaining an accurate position of the interface. However, the interface tends to be fragmented when the fluid is too

sheared compared with the mesh used.

Figure 15 represents the interface position at time $t = 6.3s$, on three meshes $G = 20*20$, $G = 180*180$ and $G = 540*540$. The time step is $\delta t = 0.0009s$. We notice that the coarser the mesh, the more fragmented the disk. The problem is that if the shearing is so strong that the number of nodes needed to avoid fragmentation is superior to the available memory size, the *VOF – PLIC* method becomes unusable.

As a consequence, we aim at coupling the *AMR* approach to the *VOF – PLIC* method so that it becomes usable for finer meshes. The results concerning the memory and *CPU* time costs are the same as those in subsection 3.2. That is the reason why we have restricted our study to the behaviour of this coupling.

We first calculate the error ϵ_{amr} defined as $\forall P \in V_{amr}, \epsilon_{amr} = \frac{\|C_{1_{amr}} - C_{1_{wamr}}\|_2}{\|C_{1_{wamr}}\|_2}$.

It gives $\epsilon_{amr} = 6 \cdot 10^{-14}$, which means that both solutions $C_{1_{amr}}$ and $C_{1_{wamr}}$ are identical. Secondly, we impose the velocity field (17) for 6.3s to the disk and then the opposite velocity field of (17) for 6.3s to reverse the shearing. We compare the phase function C_1 calculated on a monogrid $G = 540 * 540$ with C_1 obtained on a coarse mesh $G_0 = 20 * 20$ and three refinement levels, at times $t = 6.3s$ and $t = 12.6s$ (see figures 16 and 17). Results are similar in both cases, which is very satisfying. As in section 3.2, the same conclusion

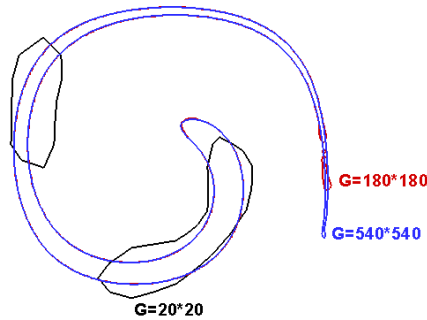


Fig. 15. Rebuilt of sheared fluid interface at time $t = 6.3s$ on three meshes.

applies when a *VOF – TVD* method is used for treating interfaces, i. e. the *AMR* method is efficient in reducing the numerical diffusion while the number of refinement levels increases.

3.4 Sheared sphere in a three dimensional one-phase flow

We consider a cubic domain Ω of characteristic length $L = 0.1m$, initially full of a pure fluid. A sphere of polluted fluid ($S_p = 1g.m^{-3}$) whose centre is $(x_0 = 0.075m, z_0 = 0.05m, y_0 = 0.05m)$ and radius is $r = 0.015m$, is initially introduced in Ω . The sphere is sheared around the y axis by a rotating velocity field (see 18) $\vec{V}(x, z, y) = (u(x, z, y), w(x, z, y), v(x, z, y))$ of averaged amplitude $V_m = 0.1m.s^{-1}$,

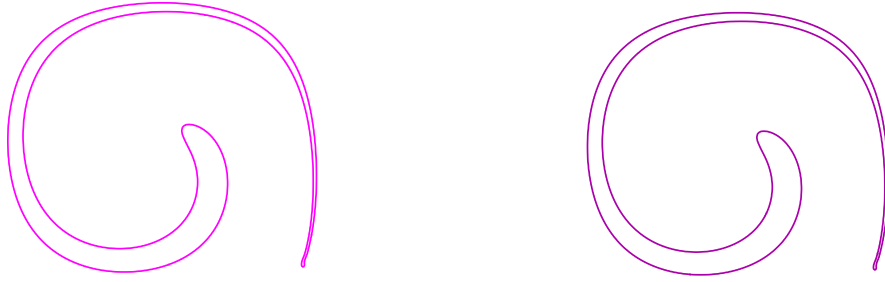


Fig. 16. Comparison of the interface position on three refinement levels ($G_0 = 20 \times 20$) (left) with the interface position obtained without *AMR* (right), at time $t = 6.3s$.



Fig. 17. Comparison of the interface position on three refinement levels ($G_0 = 20 \times 20$) (left) with the interface position obtained without *AMR* (right), at time $t = 12.6s$. The dash-dot line circle represents the initial position of the interface.

$$\begin{aligned}
 u(x, z) &= -V_m \cos\left(\frac{\pi(x - x_0)}{L}\right) \sin\left(\frac{\pi(z - z_0)}{L}\right) \\
 w(x, z) &= V_m \sin\left(\frac{\pi(x - x_0)}{L}\right) \cos\left(\frac{\pi(z - z_0)}{L}\right) \\
 v(x, z) &= 0
 \end{aligned} \tag{18}$$

The governing equation is (2) and a diffusion coefficient $D = 10^{-8} m^2 s^{-1}$ has been chosen. As in subsection 3.2, a criterion based on the variation of the concentration relative to the distance is used. Given the results obtained in subsection 3.2, the binary interpolation is utilized as the extension operator. The goal of this subsection is to evaluate the efficiency of the *AMR* method in reducing the numerical diffusion, in preserving the accuracy of solutions and in obtaining interesting memory and time costs.

For this, we first present the evolutions of the masses M_0 , M , M_1 , M_2 and M_3 obtained on a monogrid $G = 48 \times 48 \times 48$ (see figure 19 left) with these

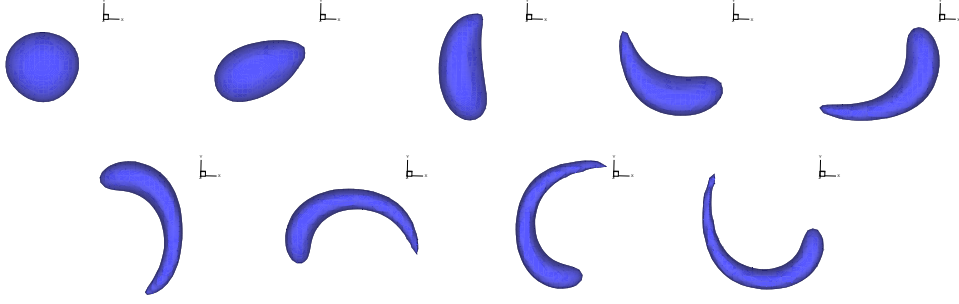


Fig. 18. Sketch concentration contour line $S_p = 0.5g.m^{-3}$ on the refined mesh $G_1 = 48^3$ until $t = 4s$. $D = 10^{-8}m^2.s^{-1}$.

obtained on a coarse mesh $G_0 = 16 * 16 * 16$ and one refinement level (see figure 19 right) during $t = 4s$. These masses are defined as follows: M_0 is the theoretical mass calculated from the sphere volume $\frac{4}{3}\Pi r^3$ and concentration $S_p = 1g.m^{-3}$. M is the total mass of the sheared sphere calculated numerically from the discretized form of equation (2). M_1 (respectively M_2 , M_3) is the mass corresponding to the concentration values superior or equal to $0.9g.m^{-3}$ (respectively $0.95g.m^{-3}$, $0.99g.m^{-3}$). We note the same evolution of each mass M , M_1 , M_2 and M_3 for both cases. We consider the monogrid case as the reference. Figure 19 shows the mass M is conserved for both cases. We notice that for each mass the evolution is similar whatever the case. This means the *AMR* approach is efficient in reducing the numerical diffusion and gives similar results to a classic method. Table 8 represents the error $e_r = \frac{|M_{num} - M_{ref}|}{|M_{ref}|}$ where M_{num} is the mass M , M_1 , M_2 or M_3 calculated from the *AMR* method and M_{ref} the mass M , M_1 , M_2 or M_3 obtained on the monogrid. Given the fact that e_r depends on time, an approximate value has been put in table 8. Whatever the mass, e_r is inferior to 2%, which is due to the fact that we assume the concentration is equal to zero outside the refinement levels. This result reinforces the previous conclusion.

Figure 20 shows the concentration contour lines $S_p = 0.2g.m^{-3}$, $S_p = 0.4g.m^{-3}$, $S_p = 0.6g.m^{-3}$ and $S_p = 0.8g.m^{-3}$ of the sheared sphere for both cases at time $t = 4s$. We can see that each contour line is superposed for both cases. The calculated error $\epsilon = 1.6 \cdot 10^{-2}$ reinforces the fact that the *AMR* approach does not damage the accuracy of the solution.

Finally, we analyse the memory cost of this method by adopting the same approach as that used in subsection 3.2. The results are written in table 9. We determine a critical value of rgl , $rgl_c = 57\%$ associated to $rap_c = 75\%$, i. e. this approach is efficient in minimizing the memory size cost as long as the number of nodes used with the *AMR* method is inferior to 57% of the nodes

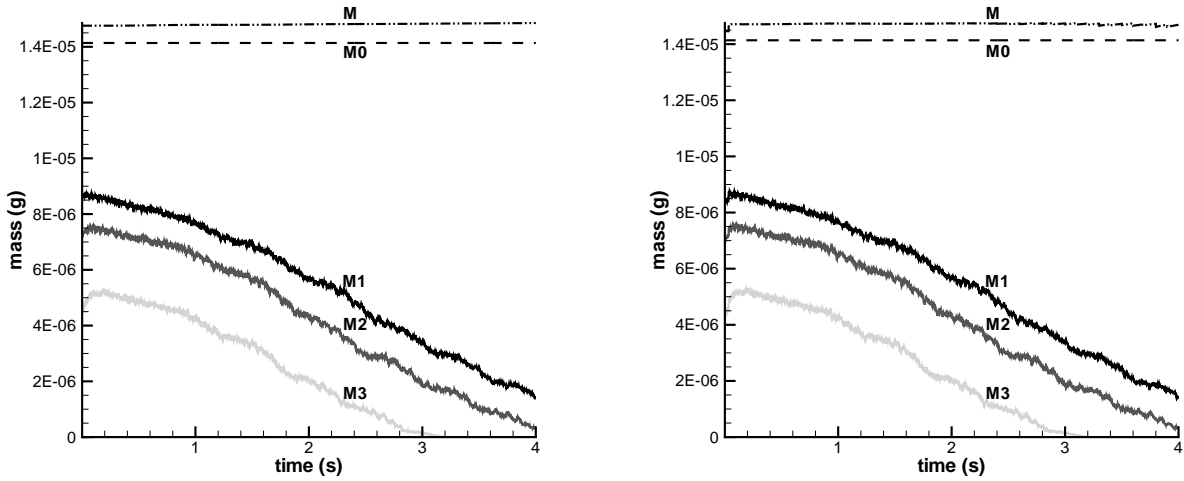


Fig. 19. Evolutions of the masses during $t = 4s$, obtained on a monogrid $G = 48 * 48 * 48$ (left) and with one refinement level from $G_0 = 16 * 16 * 16$ (right).

	M	M_1	M_2	M_3
e_r	1.28%	1%	1.24%	1.78%

Table 8

Approximative error of masses between the *AMR* and the monogrid cases.

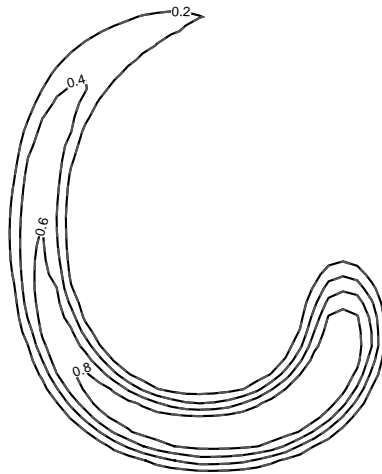


Fig. 20. Concentration contour lines in the $x-z$ plane of the sheared sphere obtained on a monogrid $G = 48 * 48 * 48$ (black solid lines) and with one refinement level (grey dashed lines) at time $t = 4s$, on the right.

which would have been used with a similar fine monogrid. We repeat that for the similar case in two dimensions of subsection 3.2, $rgl_c = 32\%$. This means that the efficiency of our approach concerning the gains in memory cost is much more interesting in $3D$ than in $2D$ (Benefits multiplied by 1.8).

mesh	mem_{amr}	mem	NBG_{niv}	KKT	rap	rgl
$48 * 48 * 48$	185	200	85413	117649	0.93	0.7
$108 * 108 * 108$	1400	2100	616072	1259712	0.67	0.5

Table 9

Memory size required to simulate the case of the sheared sphere on different meshes.

4 Application of the method to the Navier-Stokes equations in 2D

The purpose of this section is to study the ability of the *AMR* method to preserve the accuracy of the solution, to minimize the memory and *CPU* time costs while solving the Navier-Stokes equations.

4.1 Green-Taylor vortex

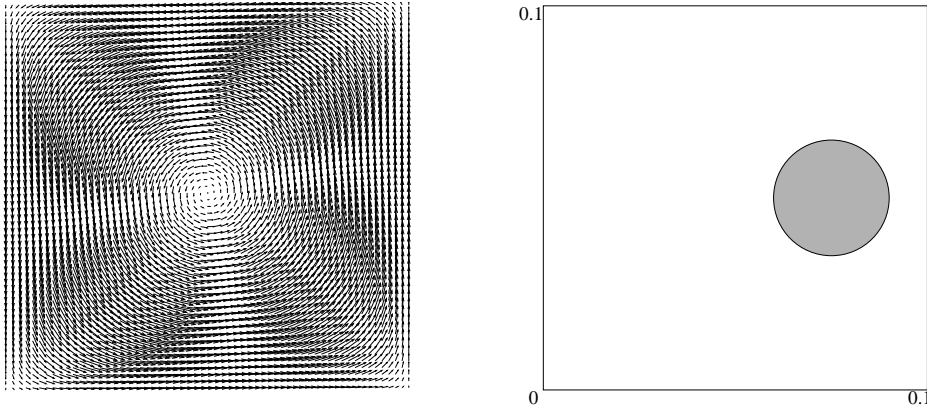


Fig. 21. Velocity field at time $t = 0.001s$ on a mesh $G = 90*90$ (left). Introduction of a polluted fluid to generate one refinement level from $G_0 = 90*90$ at time $t = 0.001s$ (right).

We consider a square domain $\Omega = [-0.1m, 0.1m] \times [-0.1m, 0.1m]$ of characteristic length $H = 0.2m$, in which an unsteady flow called the Green-Taylor vortex is induced. This vortex has been modified by introducing a source term $\vec{S}_0 = (u_0, w_0)$ in the Navier-Stokes equations in order to obtain a steady solution different from zero. More details are available in (Caltagirone, 1999). \vec{S}_0 can be written as:

$$\begin{aligned}
 u_0(x, z) &= -\frac{\pi^2 \mu}{2H^2} \cos\left(\frac{\pi x}{2H}\right) \sin\left(\frac{\pi z}{2H}\right) \\
 w_0(x, z) &= \frac{\pi^2 \mu}{2H^2} \sin\left(\frac{\pi x}{2H}\right) \cos\left(\frac{\pi z}{2H}\right)
 \end{aligned} \tag{19}$$

This gives the velocity field (see figure 21 left):

$$\begin{aligned} u(x, z, t) &= -\cos\left(\frac{\pi x}{2H}\right) \sin\left(\frac{\pi z}{2H}\right) \left(1 - \exp\left(\frac{-\pi^2 \mu t}{2\rho H^2}\right)\right) \\ w(x, z, t) &= \sin\left(\frac{\pi x}{2H}\right) \cos\left(\frac{\pi z}{2H}\right) \left(1 - \exp\left(\frac{-\pi^2 \mu t}{2\rho H^2}\right)\right) \end{aligned} \quad (20)$$

The initial velocity field is equal to zero. The analytical velocity field is imposed on the limits of the domain Ω . A disk of a second fluid is initially introduced to manage the refinement-derefinement procedure (figure 21 right). The disk is initially centered at $(x_0 = 0.05m, z_0 = 0m)$ and its radius is $r = 0.015m$. This disk is sheared by the velocity field (20). We specify the Navier-Stokes equations and equation (3) are solved. A *TVD* scheme is used to solve equation (3). A refinement criterion based on the variations of the phase function relative to the distance is used.

4.1.1 Accuracy of solutions

The aim of this subsection is to evaluate the ability of the method to solve the Navier-Stokes equations by using the augmented Lagrangian approach. This test case has been chosen as it provides an analytical solution (cf equation (20)) which enables us to study the space convergence. However the interest of the *AMR* method is limited given the fact that the velocity field does not present sharp local variations.

Table 10 represents the errors on the velocity field, ϵ_{G_1} , ϵ_{G_2} and ϵ_{G_3} , obtained for several coarse meshes G_0 . We notice the different errors decrease when the number of refinement levels or the number of nodes on G_0 increase.

These results enable us to evaluate a space convergence order $q = 2$ which demonstrates that the *AMR* approach does not damage the convergence order of the discretization schemes.

mesh G_0	ϵ_{G_1}	ϵ_{G_2}	ϵ_{G_3}
20 * 20	6.880 10 ⁻⁵	1.628 10 ⁻⁵	4.495 10 ⁻⁶
40 * 40	1.387 10 ⁻⁵	3.763 10 ⁻⁶	1.083 10 ⁻⁶
60 * 60	3.143 10 ⁻⁶	9.323 10 ⁻⁷	2.737 10 ⁻⁷

Table 10

Evolution of the error according to the refinement levels at time $t = 6s$. $\Delta t = 0.01s$.

4.1.2 Memory cost analysis

We aim at evaluating the gains of memory obtained with the *AMR* method. For this, we compare the memory size required to simulate the shearing of the

disk during 0.5s using the *AMR* approach with the memory size used for the same case without *AMR*.

We have chosen to simulate the *AMR* case with the following parameters: $G_0 = 20 * 20$, from one to three refinement levels and $\Delta t = 0.001s$.

The same approach as in subsection 3.2 is adopted and the same notations as in subsection 3.2 are used. Given the fact that we solve the Navier-stokes equations and equation (3) on a staggered cartesian grid (cf (Harlow, 1965)), the number of nodes required to solve these equations is the sum of the pressure and velocity nodes. Let NBG_{niv} (respectively KKT) be this sum of nodes for the *AMR* case (respectively the classic case without *AMR*).

We notice in table 11 that $rap < 1$ whatever the number of refinement levels. If we link the variations of rap with those of rgl , a linear regression gives a straight line of equation $rap = 2.55 * rgl - 0.015$ with a correlation coefficient equal to 0.9996. We estimate $rgl_c = 30\%$ for $rap_c = 75\%$, i. e. this approach is efficient in minimizing the memory size cost as long as the number of nodes used with the *AMR* method is inferior to 30% of the nodes which would have been used with a similar fine monogrid. So we obtain a similar result as that of subsection 3.2. This means that the gain in memory cost, obtained by the *AMR* approach compared with a classic method, depends on the number of nodes required to solve the equation and not on the number of equations solved.

mesh	mem_{amr}	mem	NBG_{niv}	KKT	rap	rgl
60 * 60	7	8	3898	11285	0.9	0.35
180 * 180	15	50	12143	98645	0.3	0.12
540 * 540	38	420	38782	879125	0.1	0.02

Table 11

Memory size required to simulate the case of the Green-Taylor vortices on different meshes.

4.1.3 CPU time cost analysis

We aim at studying the efficiency of the *AMR* approach in minimizing the *CPU* time cost of simulation t_{amr} compared with the *CPU* time cost of simulation t_{wamr} . We have chosen to simulate the *AMR* case with the following parameters: $G_0 = 20 * 20$, from one to three refinement levels and $\Delta t = 0.001s$. The same notations as in subsection 3.2 are used. Given the fact that we solve the Navier-stokes equations and equation (3), we note t_{sol-NS} (respectively t_{sol-ad}) the average *CPU* time of simulation per iteration to solve the Navier-Stokes equations (respectively equation (3)). $t_{sol} = t_{sol-NS} + t_{sol-ad}$.

We can observe on table 12 that the solving of the Navier-stokes equations is very expensive compared with the solving of equation (3) and with the refinement management. In fact, its cost represents about 99% of the average

CPU time t_{tot} . This means that the CPU time cost of the AMR algorithm is insignifiant compared with the CPU time cost of the Navier-stokes equations solving.

Table 13 shows a benefit in the simulation time t_{amr} ranging from 30% to 60% in relation to the simulation time t_{wamr} . However, it is difficult to evaluate this benefit in advance because the parameter rt cannot be linked to another one except the complexity of the solving which is impossible to estimate a priori.

number of levels	t_{ref}	t_{sol-NS}	t_{sol-ad}	t_{tot}	$\frac{t_{ref}}{t_{tot}}$	$\frac{t_{sol}}{t_{tot}}$
1	0	0.0523	0.0001	0.0526	0	1
2	0.001	0.6471	0.0013	0.6502	0.0023	0.997
3	0.0452	5.9895	0.008	6.0542	0.0075	0.991

Table 12

Comparison of averaged CPU time at $NT = 500$ for different refinement levels.

mesh	t_{amr}	t_{wamr}	rt
$60 * 60$	26.44	36.98	0.7
$180 * 180$	325.2	565.2	0.6
$540 * 540$	3027	7274	0.4

Table 13

Comparison of simulation time with and without AMR .

4.2 Collapse of a liquid column on a rigid horizontal plane

The purpose of this subsection is to analyse the ability of the AMR approach to treat a real two-phase flow case. We consider the situation described in (Martin, 1952), namely a $2D$ rectangular domain $\Omega = [0m, 1.2m] \times [0m, 0.14m]$. A water column, whose base dimension is $a = 0.057m$ and height $H_0 = n^2a$, is initially present on the left of the domain Ω . The rest of the domain is full of air. We specify n is a real number verifying $n^2 = 2$ for the treated case. The governing equations are the Navier-stokes and advection equations. A TVD scheme is used to solve equation (3).

According to (Martin, 1952), when the water column, initially at rest on the domain Ω , collapses on to the rigid plane, the fluid spreads out and the height of the column, h , falls.

The following notations are used: $T_a = nt\sqrt{\frac{g}{a}}$, $Z = \frac{z}{a}$ and $H = \frac{h}{an^2}$, where z is the distance of surge front and g the acceleration due to gravity.

We study the evolution of Z and H on a coarse mesh $G_0 = 100 * 30$ and one refinement level for the dimensionless time $T_a = 3.3$. $\Delta t = 10^{-4}s$. The results

obtained using the *AMR* method are compared with the experimental results of (Martin, 1952) and the numerical results of (Ishii, in press).

Figure 22 left represents the evolutions of H along with T_a . We notice a very slow fall at the beginning ($T_a < 0.5$). Then the speed of the H fall is intensified and is linear for $0.5 \leq T_a \leq 2.2$. Finally, the fall slows down to $H < 0.4$. We cannot fail to note the numerical results obtained using the *AMR* method are very close to those obtained by (Martin, 1952) and (Ishii, in press).

As for the evolution of Z , figure 22 right shows that the *AMR* approach and the method used by (Ishii, in press) give satisfying numerical results given the fact that the maximum error between each numerical method and the experimental results is about 6%. The only difference is that the *AMR* method under-estimates Z whereas the (Ishii, in press) approach over-evaluates it.

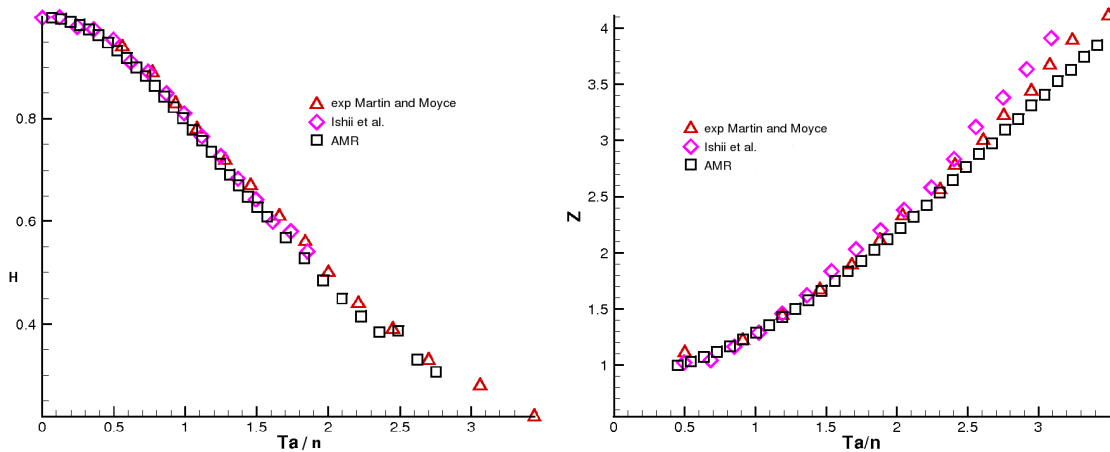


Fig. 22. Collapse of a water column: evolution of H (left) and Z (right).

4.3 Collapse of a liquid column on a dampened horizontal plane

As for in subsection 4.2, we aim at studying the ability of the *AMR* approach in treating the collapse of a water column on a dampened horizontal plane.

We consider the situation described in (Stansby, 1998), namely a 2D rectangular domain $\Omega = [0m, 1.2m] \times [0m, 0.14m]$. A water column, whose base dimension is $a = 0.06m$ and height $H_0 = 0.1m$, is initially present on the left of the domain Ω . The horizontal plane is initially covered with water $h_f = 0.01m$ high. The rest of the domain is full of air.

We compare the numerical results obtained on a coarse mesh $G_0 = 400 * 47$ with the experiment results of (Stansby, 1998) and the numerical ones of (Vincent, 1999) (see figure 23).

The water column is initially at rest on the domain Ω . When it collapses, the

fluid spreads out and the height H of the column falls on the dampened plane. Given the fact that the plane is initially covered with water, a jet forms during the collapse of the column (see figures 24 and 25) and breaks. A sketch of the column collapse and break is represented in figure 24.

We note in figure 23 that our approach and that of (Vincent, 1999) give a good estimation of the break zone around $x = 0.8m$, compared with the experimental result: the differences between numerical and experimental results vary from 1% to 3%.

The numerical results concerning the estimation of the parameter H in the break zone seems to be correct (the error between numerical and experimental results is about 16%). We assume the gap is partially due to the fact this parameter is very difficult to evaluate experimentally (see images in (Stansby, 1998)). In fact, the presence of a lot of foam makes it difficult to locate the free surface accurately.

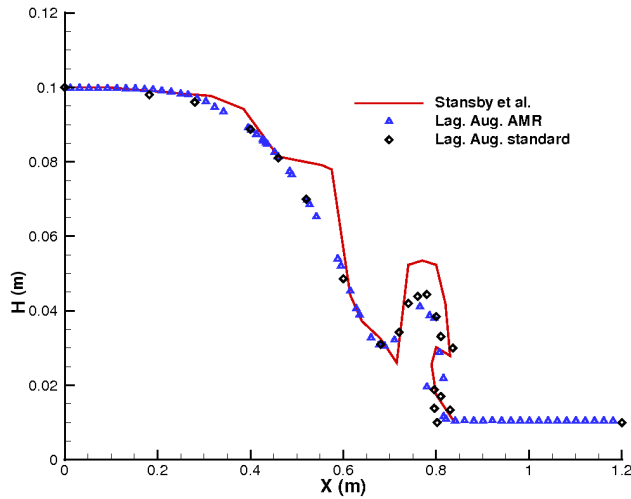


Fig. 23. Collapse of a water column: Residual height H at time $t = 0.24s$, $h_f = 0.01m$, $\Delta t = 2.10^{-4}s$. For the AMR case $G_0 = 400 * 47$ with one refinement level.

5 Conclusion

An adaptive mesh refinement method has been developed to track interfaces in two-phase incompressible flows and species concentrations in one-phase incompressible flows. The main originality of this approach consists in the implementation of an implicit method and in the fact that each refinement level is made of a series of linked AMR cells. This approach turns out to be efficient in treating this sort of problem in $2D$ in so far as it enables us to obtain accurate solutions, second convergence order for both Navier-Stokes and transport equations and numerical solutions close to experimental and numerical results

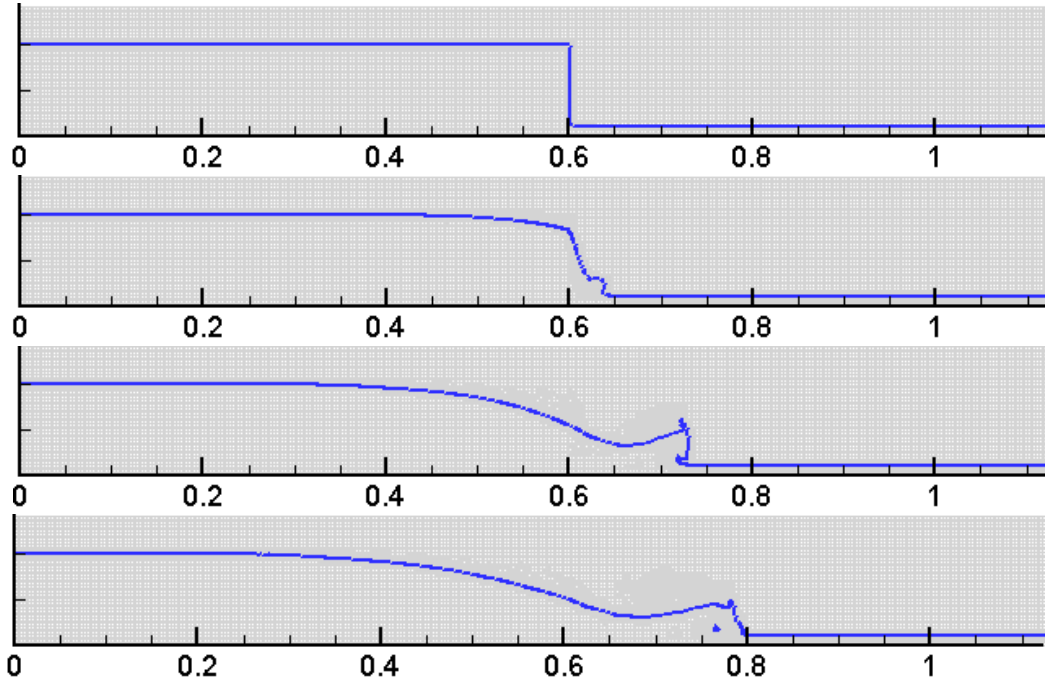


Fig. 24. Evolution of the interface at times $t = 0.01s$, $t = 0.06s$, $t = 0.16s$ and $t = 0.22s$ (from the top to the bottom). $G_0 = 400 * 47$ with one refinement level, $\Delta t = 2 \cdot 10^{-4}s$ and $h_f = 0.01m$.

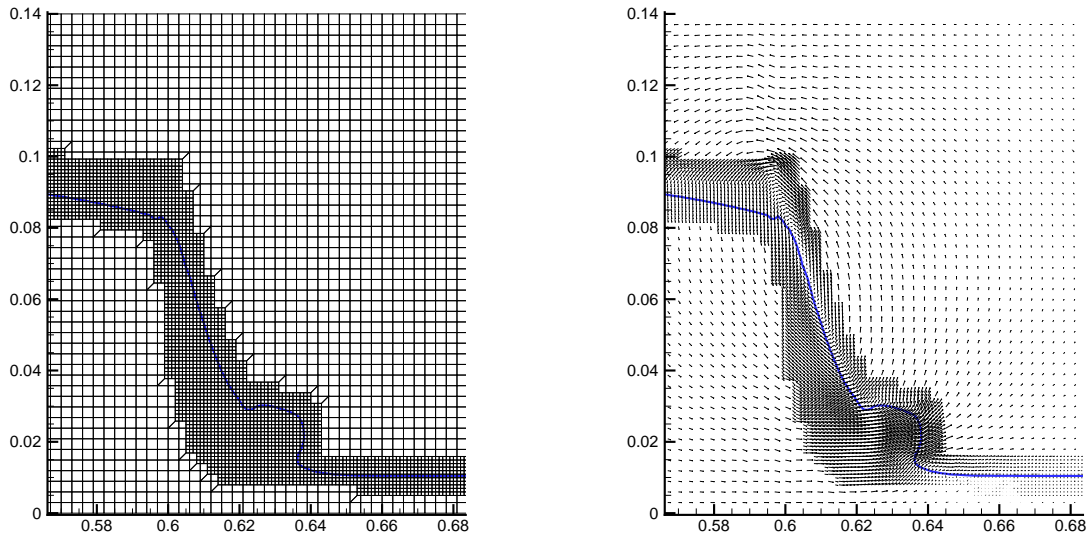


Fig. 25. Forming of the jet at $t = 0.06s$ during the collapse of a water column. Mesh used on the left: $G_0 = 400 * 47$ with one refinement level. Velocity field on the right.

from the literature. The accuracy is likely to be improved by the development of conservative interpolations to treat the velocity field.

The extension of the method to $3D$ gives good results concerning its ability to track fronts in an imposed velocity field. Its adaptation to the $3D$ Navier-Stokes equations is in progress.

The *AMR* approach has been optimized with respect to memory requirements

and *CPU* time. It results in a memory profit superior or equal to 25% as long as the number of nodes used with the *AMR* method is inferior to around $rgl_c = 30\%$ of the nodes which would have been used with a similar fine monogrid, for both 2*D* Navier-Stokes and transport equations. The memory benefits are increased for 3*D* transport equations in so far as rgl_c reaches a value of 57%.

Its efficiency in reducing *CPU* time has been demonstrated in 2*D* for both Navier-Stokes and transport equations. In fact, *CPU* time profits increase when the complexity of the equations solving increases. As it has been explained, this is due to the fact that the dynamic management of the *AMR* approach becomes negligible compared with the solving of complex equations.

References

- P. Angot, J. P. Caltagirone and K. Khadra, An adaptive method for local mesh refinement: the Flux Interface Correction *C. R. Acad. Sci. Paris, Serie I*, vol. 315, 739-745, 1992.
- E. Bellenger and P. Coorevits, Adaptive mesh refinement for the control of cost and quality in finite element analysis, *Finite Elements in Analysis and Design*, vol. 41, 1413-1440, 2005.
- M. J. Berger and P. Colella, Local adaptive mesh refinement for hyperbolic partial differential equations, *J. Comput. Phys.*, vol. 82, 64-84, 1989.
- J. P. Caltagirone and J. Breil, A vector projection method for solving the Navier-Stokes equations, *C. R. Acad. Sci. Paris, Serie II b, Mecanique des fluides numrique*, vol. 327, 1179-1184, 1999.
- S. Delage, S. Vincent, J.P. Caltagirone and J.P. Heliot, A hybrid linking approach for solving the conservation equations with an Adaptive Mesh Refinement method, *Journal of Computational and Applied Mathematics*, vol. 191, issue 2, 280-296, 1 july 2006.
- S. Delage-Santacreu, A hybrid Adaptive Mesh Refinement method to track interfaces in incompressible flows, *Phd thesis*, Universite Bordeaux1, 2006.
- M. Fortin and R. Glowinsky, Methodes de Lagrangien augmente. Application a la resolution numerique de problemes aux limites. *Methodes mathematiques de l'informatique*. **1** Dunod, 1982.
- I. Gustafsson, On first and second order symmetric factorization methods for the solution of elliptic difference equations, *Chalmers* **1** University of technology, 1978.
- W. Hackbush, Local defect correction method and domain decomposition techniques, in defect correction methods: Theory and applications, *K. Bhmer and H. J. Stetter, eds computation supplementation, Springer-Verlag, Wien*, vol. 5, 89-113, 1984.
- H. Harlow and J. E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *Phys. Fluids*, vol. 8, 2182-2189,

- 1965.
- C. Hirsh, Numerical computation of internal and external flows, volume 2: computational methods for inviscid and viscous flows, *Lectures in mathematics, Landford, Birkhauser, Zurich*, 1992.
- C. W. Hirt and B. D. Nichols, Volume of fluids (VOF) method for the dynamics of free boundaries, *Journal of Computational Physics*, vol. 39, issue 1, 201-225, 1981.
- E. Ishii, T. Ishikawa and Y. Tanabe, Hybrid Particle/Grid method for predicting motion of micro and macro-free surfaces, *Journal of Fluids Engineering*, in press.
- I. Kataoka, Local instant formulation of two phase flow, *Int. J. Multiphase Flow*, vol. 12, 745-758, 1986.
- R. J. Leveque, Numerical methods for conservation laws, *John Wiley and Sons*, Chichester, 1990.
- X. D. Liu, S. Oscher and T. Chan, Weighted Essentially Non Oscillatory Schemes, *J. Comput. Phys.*, vol. 115, 200-212, 1994.
- J. C. Martin and W. J. Moyce, An experimental study of the collapse of liquids columns on a rigid horizontal plane, *Phys. Trans. Serie A, Math. Phys. Sci.*, vol. 244, 312-325, 1952.
- S. V. Patankar, Numerical heat transfer and fluid flow, *Series in computational methods in mechanics and thermal sciences. 1* Hemisphere Publishing Corporation, 1980.
- R. Peyret and T. D. Taylor, Computational methods for fluid flows, *Springer series in computational physics*. Springer, verlag, New York, 1983.
- P. K. Stansby, A. Chegini and T. C. D. Barnes, The initial stages of dam-break flow, *Journal of Fluid Mechanics*, vol. 374, 407-424, 1998.
- U. Trottenberg, C. Oosterlee and Schuler, Multigrid, *Academic Press*, 2001.
- T. Yabe, F. Xiao and T. Utsumi, The Constrained Interpolation Profile method for multiphase analysis, *J. Comput. Phys.*, vol. 169, issue 2, 556-593, 20 may 2001.
- H.A. Van Der Vorst, A fast and smoothly converging variant of BI-CG for the solution of non-symmetric linear systems, *SIAM J. Sci. Stat. Comput.* **1** vol. 13, number 2, 631-644, 1992.
- S. Vincent, Modelling incompressible flows of non-miscible fluids, *Phd thesis*, Universite Bordeaux1, 1999.
- S. Vincent and J.P. Caltagirone, Efficient solving method for unsteady incompressible interfacial flow problems, *Int. J. Numer. Meth. Fluids*, vol. 30, 795-811, 1999.
- S. Vincent and J.P. Caltagirone, One Cell Local Multigrid method for solving unsteady incompressible multi-phase flows, *J. Comput. Phys.*, vol. 163, 172-215, 2000.
- S. Vincent, J. P. Caltagirone, P. Lubin and T. N. Randrianarivelo, An adaptive augmented Lagrangien method for three dimensional multimaterial flows, *Computers and Fluids*, vol. 33, 1273-1289, 2004.
- S. Whitaker, Fundamental Principles of Heat Transfer, *Robert E. Krieger Pub-*

lishing company, Malabar, Florida, 1983.
D. L. Youngs, Time-dependent multimaterial flow with large fluid distortion,
K. W. Morton and M. J. Baines, New York (USA), 1982.