A Two-Dimensional Second Order Conservative Front-Tracking Method with an Original Marker Advection Approach Based on Jump Relations

Mathilde Tavares, Désir-André Koffi Bi, Eric Chénier, Stéphane Vincent

To cite this version:

Mathilde Tavares, Désir-André Koffi Bi, Eric Chénier, Stéphane Vincent. A Two-Dimensional Second Order Conservative Front-Tracking Method with an Original Marker Advection Approach Based on Jump Relations. Communications in Computational Physics, 2020, 27 (5), pp.1550-1589. ⟨10.4208/cicp.OA-2019-0028⟩. ⟨hal-04891078⟩

HAL Id: hal-04891078
https://hal.science/hal-04891078v1

Submitted on 16 Jan 2025

A two-dimensional second order conservative front-tracking method with an original marker advection approach based on jump relations

Mathilde Tavares[1], **Désir-André Koffi-Bi, Eric Chénier, Stéphane Vincent**

Université Paris-Est Marne-la-Vallée, Lab. Modélisation et Simulation Multi Echelle, MSME, UMR 8208 CNRS 5 bd Descartes, 77454 Marne-la-Vallée, France

Abstract:

A two-dimensional front-tracking method is developed for handling complex shape interfaces satisfying the volume conservation. In order to validate the proposed front-tracking method, a complete convergence study is carried out on several analytical test cases for which the interface is widely stretched and deformed. Comparisons to different existing approaches show that our front-tracking method is second order accurate in space with lower errors than existing interface tracking techniques of the literature.
We also propose an original marker advection method which takes into account the jump relations valid at interface in order to deal with the contrast of physical properties encountered in two-phase flow simulations. The conservative front-tracking method computed in this work is shown to be able to describe interfaces with high accuracy even for poorly resolved Eulerian grids.

Key words : Front-tracking, marker velocity reconstruction based on jump relations, multiphase flow, volume conservation, second order accuracy

1 Introduction

In past two decades, major progress has been achieved in fluid interface tracking when multiphase flow simulations are tackled with. Various tracking techniques have been investigated in the literature to handle these flow simulations. Depending on how the pressure and velocity variables are computed and how the interface is taken into account in the conservation equations, either explicitly with a moving interfacial mesh or indirectly by capturing the presence of the interface through an auxiliary variable, these tracking techniques can be divided in two major classes:

- On one hand, moving unstructured body fitted meshes, pure Lagrangian [17] or arbitrary Eulerian-Lagrangian methods [14, 13] are techniques that conform to the interfacial shape. They use the two-fluid flow equations for simulating multiphase flows, *i.e.* the Navier-Stokes equations in each phase and jump relations on the mesh elements of the interface [8]. Indeed, the deformations of the interface are explicitly tracked over time with deformable meshes which follow the front surface. Quite popular for fluid structure interaction problems, ALE approaches are widely used in this class of tracking methods for free surface flow problems since they avoid the fast mesh distortion known from pure Lagrangian methods. Thanks to the explicit representation of

---

[1] mathilde.tavares@u-pem.fr

the interface, surface characteristics such as the surface tension of the material properties can be computed accurately. However, the management of the moving deformable mesh involves high complexity and limits the application of this approach to two dimensional configurations or weakly deformable interfaces in three dimensions. In the range of body fitted approaches with mesh points describing the interface, the adaptive curvilinear orthogonal grid techniques developed in Jadim code are also interesting to mention [28]. For example, they allow the accurate simulation of single droplet or bubble flows in axisymmetric coordinates.

- On the other hand, another class of approach to track interfaces, generally on fixed Eulerian meshes, consists in capturing directly the presence of the front by using an auxiliary Eulerian variable. The pioneer work of Harlow and Welch [18] deals with the tracking of Lagrangian volume parcels of fluid in order to describe the evolution of one of the two existing phases on a structured grid. In these approaches, the interface is not *a priori* known, only its presence in a given cell is reported. Among the huge amount of works devoted to these interface capturing techniques, we can cite the Volume Of Fluid (VOF) method [59], for which the interface is represented by means of a phase function. This is one of the oldest but still popular front-capturing approach as it provides naturally volume conservation of phases. Another approach that is very often used by the scientific community is the level-set method [50]. With this method, the interface is located by a signed distance function which is continuous across the interface and allows an accurate estimate of the curvature and normal at the interface. A combination of both level-set and VOF methods has been proposed by various authors [49, 10, 5] to ensure a mass conservative tracking approach while preserving the simple definition and implementation of the level-set method. Even if the research has been focused on fixed regular grids, more recent developments [6, 9, 22, 58, 32] allowed the extension of these methods to three dimensional unstructured meshes for VOF or level-set methods so that they can handle problems with strong deformations and topological changes of the interface while refining the mesh locally around interface to capture vorticity or dynamic boundary layers. The phase field approach is a last class of implicit description of the interface. It relies on the definition of a physical interfacial thickness that belongs to diffuse interface models [23, 44, 1]. Although these methods are widely used to solve complex multiphase flows, their main drawback is that the size of smaller resolved interfacial structures is controlled by the Eulerian mesh. For multiphase flows involving subgrid size bubbles, droplets or ligaments, few VOF and Level Set methods have been extended to compute the multi-scale character of the multiphase flows. Among them, we can cite [31, 29, 21, 4, 56, 27] for VOF-like methods with submesh interface description, [57, 39, 60] for adaptive mesh refinement around interfaces, [61, 51] for VOF and level set coupling with point-particle tracking of small drops and bubbles, [20] for a dual mesh level set approach for which the interface is always resolved on a very fine mesh and the Moment of Fluid technique [25] that allows a sub-mesh reconstruction of the interface by using the centroid of the phases in a given cell. Even in their most recent improvements, the main difficulty of the front capturing approaches is to maintain the sharpness of the front during the flow motion while accounting at the same time for flow structures whose characteristic size is smaller or equal to the mesh cell length. To prevent the drawbacks of previously presented front capturing techniques, another class of methods for handling interface evolutions on a fixed Eulerian mesh is Lagrangian. It consists in building a surface mesh that follows the interface. These front-tracking approaches are accurate and allows intrinsically keeping subgrid scale interfacial structures. However, they involve high complexity mesh management in two- and three-dimensional flows. Despite the difficulty of dealing with two grids, the sharpness of the front is always maintained even for under-resolved Eulerian grids making the front-tracking approach one of the most accurate methods to track interfaces and deal with capillary effects.

We focus in this paper on this last approach, the Front-Tracking (FT) method, which basically couples two non-conforming meshes:

- A specific mesh of the surface separating one phase to the rest of the flow motion is used to track explicitly the interface. This surface mesh is sliding over the Eulerian mesh, which is chosen fixed in our developments. It deforms according to the flow motion and capillary effects exerted on it.
- An Eulerian fixed mesh is considered for solving the conservation equations. A coupling between the front tracking mesh and the flow variable mesh is required. Most of the subtlety of the accuracy and the physical relevancy of the method is hidden in this step.

The FT method has been extensively developped and studied in literature. Many authors [11, 55, 40, 47, 52] have shown in their works the strength of the front-tracking compared to most popular Eulerian interface capturing methods. We can detailed the front tracking works as follows. Glimm and *al.* [16] first represents the interface by connected markers for shock waves applications and then extended the use of the front-tracking method to various flow problems [15, 11]. Unverdi and Tryggvason [55] generalized the application of the front-tracking to viscous, incompressible multiphase flows for the simulation of a rising bubble. An improvement of the capillary forces calculation in the front-tracking method was performed by Popinet and Zaleski [40] in particular by modifying the projection step of time splitting methods for incompressible two-phase flows. The simulation of homogeneous bubbly flows was also performed by Tryggvason and *al.* [53] with a front-tracking method. A complete review of the use of front-tracking method to track interfaces in multiphase flows is beyond the scope of this paper and all details can be found in [54]. We can emphasizes that the front-tracking method is most generally used for applications with small interfacial scale ranges such as bubbly flows or shock waves. The main difficult aspect of front tracking is tackling with coalescence and rupture that have to be imposed with a physical length scale that is not *a priori* known.

Our main goal is to provide a detailed analysis of front tracking numerical performances in order to improve as much as possible the accuracy of the method while obtaining a second order convergence in space. This point is very important as coarsely or under resolved interfacial structures observed in multiscale multiphase flows will be described, on the Eulerian flow mesh, by few (at least one) cells and so the intrinsic accuracy of the method is of paramount importance. We propose in this work a preliminary path with an improvement of the front-tracking method. We have studied different existing solutions in order to obtain the most accurate advection of the interface as possible and we have set-up algorithms to keep the volume constant after the interface advection step. A convergence study has been performed on several test cases and our front-tracking method has been compared to other classical interface capturing and front tracking techniques of the literature. We have also devoted a part of this work to the calculation improvement of specific characteristics of the interface such as the local curvature and the normal. In multiphase flow simulations, overcoming the loss of accuracy of marker velocity calculation during the advection step of the Lagrangian interfacial mesh is of major importance as soon as the ratio of physical properties between different phases is large. An original marker advection method is then proposed, that accounts for velocity and stress jump relations at the interface. The idea is to deal with the jump of the physical properties directly in the solving step devoted to the Lagrangian interfacial mesh instead of using kinematic interpolation procedures.

This paper is structured as follows: the second section is dedicated to the improvement and validation of the front tracking method for kinematic problems. The first subsection is devoted to describing the front-tracking method with the advection step and the reseeding operations required to maintain a constant resolution of the interface. A validation test case is also proposed to show the ability of the front-tracking method to accurately locate interfaces. The second subsection discusses volume conservation algorithms used to keep the volume constant after each advection step. We have focused in the third subsection on the calculation of interfacial characteristics, namely local curvature and normal computations. Two different methods have been studied and compared to assess the most accurate one. In the fourth subsection, we have compared our proposed front-tracking method to classical interface tracking methods of the literature in order to confirm the idea that the front-tracking is the most accurate interface tracking method as previously stated by Trontin and *al.* [52] and Glimm and *al.* [11]. In the third section, we present an original method devoted to the advection of the front-tracking markers in real multiphase flows. The jumps of physical properties such as pressure, velocity and stress are used to account for viscosity ratio at the interface and capillary forces directly in the marker advection step. The jump marker velocity calculation is validated against Hadamard-Rybczynski analytical solution. This new marker velocity estimate prevents the use of artificial interpolation procedures. Conclusions and perspectives are finally drawn in the fourth section.

2 Front-tracking method

The front-tracking method is based on the Tryggvason and *al.* work [53] for multiphase and isothermal flows. The interface consists of a chain of markers connected by segments. Contrary to the Shin and Juric description of the front [47], the connectivity between neighbouring elements is also stored in order to efficiently compute the surface tension force, namely its magnitude and its direction.

Let us introduce the notations used in this paper (see Fig. 1 for a graphical illustration). We denote by $\mathcal{D} = (\mathcal{E}, \mathcal{M}, \mathcal{T})$ a space discretization for two-dimensional problems.

- $\mathcal{E}$ is a set of $N_{\mathcal{E}}$ elements (or segments) all joined two by two through their endpoints to produce a closed curve. Any component of $\mathcal{E}$ is usually noted e.
- $\mathcal{M}$ is the set of $N_{\mathcal{E}}$ points, called markers, which connect two adjacent elements. Any component of $\mathcal{M}$ is usually noted m. A specific notation is also used to point out the link between any element $e \in \mathcal{E}$ with its two endpoints noted in that case $(m_{[e,1]}, m_{[e,2]}) \in \mathcal{M}^2$.

For all $e \in \mathcal{E}$, the neighbouring element sharing the marker $m_{[e,1]}$ (resp. $m_{[e,2]}$) is noted $e_{[1]}$ (resp. $e_{[2]}$). The local numbering of the markers is such that $m_{[e,1]}$ and $m_{[e_{[1]},2]}$ are related to a unique component in $\mathcal{M}$, because they stand for the same endpoint connecting the elements e and $e_{[1]}$. The same remark is applied for $m_{[e,2]}$ and $m_{[e_{[2]},1]}$. To emphasize the equivalence between the different notations, the symbol "$\equiv$" is used ($m_{[e,1]} \equiv m_{[e_{[1]},2]}$ and $m_{[e,2]} \equiv m_{[e_{[2]},1]}$). We also define:

- $\mathcal{T}$ the set of $N_{\mathcal{E}}$ 2-tuples such that for all $e \in \mathcal{E}$ and $(m_{[e,1]}, m_{[e,2]}) \in \mathcal{M}^2$, we have $(m_{[e,1]}, m_{[e,2]}) \in \mathcal{T}$. The order of the two markers is guaranteed because, for all $e \in \mathcal{E}$, $m_{[e,2]} \equiv m_{[e_{[2]},1]}$ (the second marker of the element e is identical to the first marker of its neighbor $e_{[2]}$). By convention, we decide that the two markers $m_{[e,1]}$ and $m_{[e,2]}$ of any element $e \in \mathcal{E}$ of a closed interface are described counter-clockwise.

For any marker $m \in \mathcal{M}$, we designate by $\bar{x}_m$ and $\bar{V}_m$ its coordinate and velocity. Similarly, we define $\bar{n}_m$, the unit normal vector pointing outward to the volume delimited by the discrete closed interface and located at the marker $m \in \mathcal{M}$. The mathematical expression of the vector $\bar{n}_m$ will be presented and discussed later in Sec. 2.3. It is worth to point out that the size and the component values of the different sets evolve as a function of time. For example, $\mathcal{E}^{(n)}$ indicates the elements set at time $t^{(n)}$.

Lastly, to clarify the vocabulary in the rest of the paper, from now on we denote by "grid(s)", the Eulerian staggered mesh devoted to the resolution of the fluid flow equations (Navier-Stokes equations for example). The term "mesh" refers exclusively to the discretization of the Lagrangian interface which consists of elements and markers as described here-above.

In the following subsections are presented the basic features of our front-tracking method. This includes the comparison of different second order advection methods accounting for a reseeding procedure for highly stretched fronts. Since the simple advection of the interface does not guarantee the volume conservation, an original method has been developed to overcome this drawback and ensure this fundamental property for incompressible flows. Then, some attention is paid both on the calculation of the interface curvature and the direction of the surface tension force.

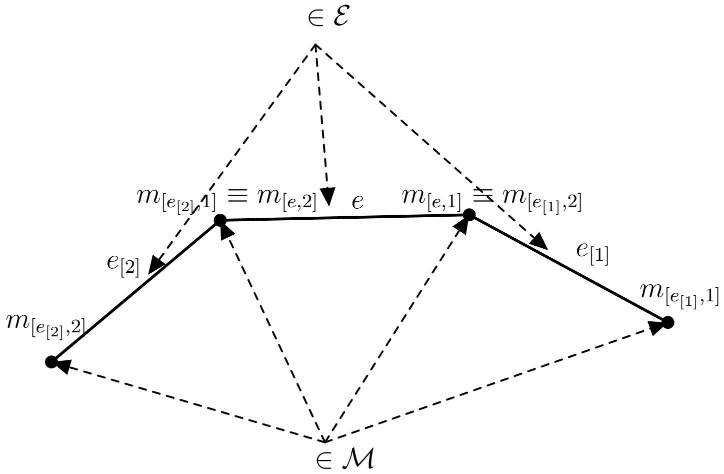


Figure 1: Discretization of the interface and notations

2.1 Base components of the front-tracking method

2.1.1 Lagrangian advection of the interface markers

The interface is advected by solving the following Lagrangian equation: $\forall m \in \mathcal{M}$,

$$\frac{d\bar{x}_m}{dt} = \bar{V}_m(t) \tag{1}$$

where $\bar{V}_m(t)$ stands for the Lagrangian velocity of marker m. A second order Runge-Kutta scheme is used for time integration: for all $m \in \mathcal{M}^{(n,n+1)}$ (same marker at $t^{(n)}$ and $t^{(n+1)}$),

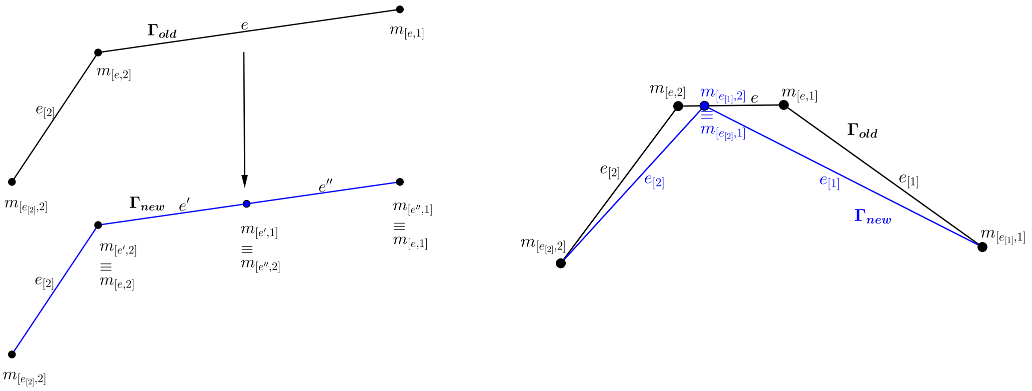
$$\bar{x}_m^{\star} = \bar{x}_m^{(n)} + \Delta t \bar{V}_m^{(n)} \tag{2a}$$

$$\bar{x}_m^{(n+1)} = \frac{1}{2}\left(\bar{x}_m^{(n)} + \bar{x}_m^{\star}\right) + \frac{\Delta t}{2}\bar{V}_m^{(\star,n+1)} \tag{2b}$$

with $\Delta t \overset{\text{def}}{=} t^{(n+1)} - t^{(n)}$ the time step, and $\bar{x}_m^{(n)}$ (resp. $\bar{x}_m^{(n+1)}$) the position of the marker m at $t^{(n)}$ (resp. $t^{(n+1)}$). The intermediate Lagrangian velocity $\bar{V}_m^{(\star,n+1)}$ at position $\bar{x}_m^{\star}$ and time $t^{(n+1)}$ is calculated with a second order Adams-Bashforth extrapolation method $\bar{V}_m^{(\star,n+1)} \overset{\text{def}}{=} 2\bar{v}^{(n)}(\bar{x}_m^{\star}) - \bar{v}^{(n-1)}(\bar{x}_m^{\star})$ where $\bar{v}^{(n)}(\bar{x})$ and $\bar{v}^{(n-1)}(\bar{x})$ are the Eulerian flow velocity at times $t^{(n)}$ and $t^{(n-1)}$. Since $\bar{x}_m$ and $\bar{x}_m^{\star}$ are not located at the velocity grid nodes, different interpolation procedures are investigated to get the velocities at these points (see Sec. 2.1.3 for the numerical tests). Two of them are quite common in the framework of the front-tracking methods, namely the bilinear interpolation method [34] and the Peskin technique [38]. The third method that is considered here is the Parabolic Edge Reconstruction Method (PERM) by McDermott and *al.* [33] which consists in a divergence free approximation at the nodes and cells centers of the Eulerian velocity grids. The reconstructed velocity is continuous and piecewise parabolic in the direction of the considered velocity component and piecewise linear in the transverse direction. For a given cell, the reconstructed divergence is thus bilinear and consistent with the discrete cell divergence.

2.1.2 Management of the interface elements size

In two-phase fluid flows, the interface can be highly stretched and deformed, inducing the appearance of regions with different marker densities. A too small marker density with respect to the Eulerian grid size h may lead to an overall loss of accuracy. In contrast, a too large marker density is CPU and memory consuming and may create spurious oscillations during the advection step [30]. To maintain a good accuracy at each time step, markers are locally inserted or suppressed in order to keep the instantaneous size of each interface element almost constant. Let us note the size of any element $e \in \mathcal{E}^{(n)}$ by $h_e^{(n)}$. Basically, the interface elements are initially set to about half the Eulerian grid size ($h_e \overset{\text{def}}{=} h_e^{(0)} \approx h/2$, see Sec. 2.1.3) and during the advection step, they are controlled so that their length remains mainly bounded between $h_e/2$ and $2h_e$ [53].



(a) Splitting of element e

(b) Deletion of element e

Figure 2: Reseeding operation of the front, Γ_{old} represents the front before the reseeding and Γ_{new} the front after the reseeding.

At $t^{(n)}$, if the size $h_e^{(n)}$ of element $e \in \mathcal{E}^{(n)}$ is too large, the mesh is enriched: the size $N_{\mathcal{E}}$ of the sets $\mathcal{E}$, $\mathcal{M}$ and $\mathcal{T}$ is increased by one. The element e is split into two twice smaller elements (Fig. 2a), a new element e' and an element e'' which substitutes for the former element e. A new interface marker $m_{[e',1]} \equiv m_{[e'',2]}$ is also introduced at the center of the original element e, $\bar{x}_{m_{[e'',2]}} \equiv \bar{x}_{m_{[e',1]}} = (\bar{x}_{m_{[e,1]}} + \bar{x}_{m_{[e,2]}})/2$, and a new 2-tuple is defined for the new element e': $(m_{[e',1]}, m_{[e',2]}) \in \mathcal{T}$. The pointers associated with the markers and neighbors are updated:

- New Element e': $m_{[e',2]} \equiv m_{[e,2]}$, $e'_{[1]} = e''$ and $e'_{[2]} = e_{[2]}$. Reciprocally, the first neighbor of element $e_{[2]}$ is ${e_{[2]}}_{[1]} = e'$.
- Element e'', formerly element e: $m_{[e'',1]} \equiv m_{[e,1]}$, $e''_{[2]} = e'$ and $e''_{[1]} = e_{[1]}$. Reciprocally, the second neighbor of element $e_{[1]}$ is ${e_{[1]}}_{[2]} = e''$.

The velocity of the new marker $m_{[e',1]} \equiv m_{[e'',2]}$ is approximated by a linear combination of the velocities of the adjacent markers: $\bar{V}_{m_{[e'',2]}} \equiv \bar{V}_{m_{[e',1]}} = (\bar{V}_{m_{[e,1]}} + \bar{V}_{m_{[e,2]}})/2$.

At $t^{(n)}$, the deletion of a too small element $e \in \mathcal{E}^{(n)}$ depends on the dimensionless curvatures calculated at the markers $m_{[e,1]}$ and $m_{[e,2]}$ in order to avoid the local deterioration of the interface shape. If $\kappa_{[e,1]} h_e^{(n)}$ and $\kappa_{[e,2]} h_e^{(n)}$ are less than the threshold value $\kappa_{th} = 0.1$ (see Sec. 2.3 for the curvature approximation), the two markers of the element e are merged together to form a unique marker located between the two previous ones (Fig. 2b). The coordinate of this new marker is given by the arithmetic mean of the two parents coordinates weighted by the curvatures:

$$\bar{x}_{m_{[e_{[1]},2]}} \equiv \bar{x}_{m_{[e_{[2]},1]}} = \frac{\kappa_{[e,1]} \bar{x}_{m_{[e,2]}} + \kappa_{[e,2]} \bar{x}_{m_{[e,1]}}}{\kappa_{[e,1]} + \kappa_{[e,2]}}$$

The components $e \in \mathcal{E}^{(n)}$, $m_{[e,2]} \in \mathcal{M}^{(n)}$ and $(m_{[e,1]}, m_{[e,2]}) \in \mathcal{T}^{(n)}$ are suppressed, leading to a decrease of the size of the different sets: $N_{\mathcal{E}} \leftarrow N_{\mathcal{E}} - 1$. Again, the local connectivity must be updated.

2.1.3 Test case: Circle deformation in a vortex flow

In the present section, we consider the test case by Bell *et al.* [2] which consists in the advection of a circle, initially located at $(0.5\,m; 0.75\,m)$ with a $0.15\,m$ radius, in a vortex flow given by the velocity field $\bar{v} = u\bar{e}_x + v\bar{e}_y$ with

$$u(x,y) = -2\sin(\pi x)^2\cos(\pi y)\sin(\pi y) \qquad [u] = m/s \tag{3a}$$

$$v(x,y) = 2\sin(\pi y)^2\cos(\pi x)\sin(\pi x) \qquad [v] = m/s \tag{3b}$$

In Fig. 3 is illustrated the interface shape at times $t = 0$ and $4\,s$ for an Eulerian grid $h = 1/512\,m$, a time step $\Delta t = h/(2V_0)\,s$ with $V_0 = 1\,m/s$, and a marker velocity based on the PERM interpolation method. In this simulation, the interface is widely stretched and deformed to finally give rise to a long ligament much thinner than the Eulerian grid size h.

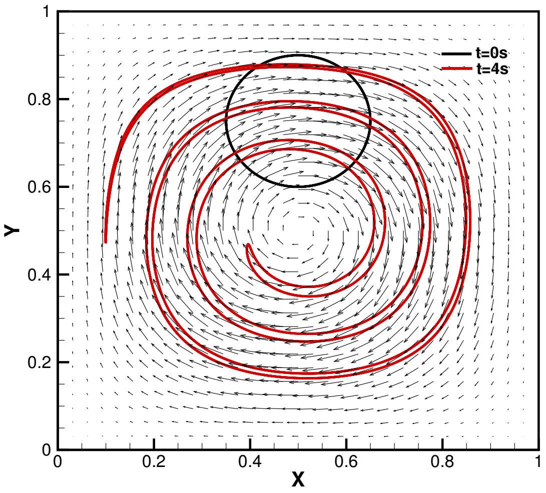


Figure 3: Circle advection by a vortex flow (Eq. 3a-3b). Initial solution in black, $t = 4\,s$ in red and velocity vector field are plotted.

Since no analytical solution exists for this problem at maximum interface deformation, a reference interface $\Gamma_{h_e^{\text{ref}}}(t)$ has been computed using a $4th$ order Runge-Kutta scheme combined with an exact velocity field for advecting the markers, a time step $\Delta t = h^{\text{ref}}/2$ and an initial interface element size such that for any element $e \in \mathcal{E}^{(0)}$, $h_e^{\text{ref}} \overset{\text{def}}{=} h_e^{\text{ref}}(0) = h^{\text{ref}}/100$, with $h^{\text{ref}} = 1/4096$ (equivalent to a very fine Eulerian grid size). Concerning the coarse interface meshes that will be compared to the reference interface, they are in the order of the Eulerian grid size h, with an initial element size $h_e = h/2$, $\forall e \in \mathcal{E}^{(0)}$, a time step $\Delta t = h/2$, a $2nd$ order Runge-Kutta scheme and the use of different interpolation algorithms to get the velocity of the markers from the Eulerian staggered grids. The measure of the error is defined by the distance between the markers of the interface coarse mesh $\Gamma_{h_e}^{(n)}$ to the reference one $\Gamma_{h_e^{\text{ref}}}^{(n)}$ at time $t \overset{\text{def}}{=} n\Delta t = 4\,s$ (see Fig. 3):

$$\|E_{h_e}\|_1 \overset{\text{def}}{=} \frac{1}{N_{\mathcal{E}}^{(n)}} \sum_{m\in\mathcal{M}^{(n)}} \min_{p\in\Gamma^{(n)}_{h_e^{\text{ref}}}} \|\bar{x}[p]-\bar{x}_m\|_2 \tag{4}$$

$$\|E_{h_e}\|_\infty \overset{\text{def}}{=} \max_{m\in\mathcal{M}^{(n)}} \left(\min_{p\in\Gamma^{(n)}_{h_e^{\text{ref}}}} \|\bar{x}[p]-\bar{x}_m\|_2 \right) \tag{5}$$

where $\|.\|_2$ is the Euclidian norm, $N_{\mathcal{E}^{(n)}}$ is the number of elements in the set $\mathcal{M}^{(n)}$, p is any point of the interface $\Gamma^{(n)}_{h_e^{\text{ref}}}$ and $\bar{x}[p]$ stands for its coordinates. We also define the convergence order of a quantity $X(h)$ between two grid or mesh sizes h_1 and h_2 by the ratio

$$\text{Order} \overset{\text{def}}{=} \frac{\log\left(\|X(h_1)\|_{ni} \,/\, \|X(h_2)\|_{ni}\right)}{\log\left(h_1/h_2\right)} \tag{6}$$

with $ni = 1$ or ∞.

Table 1 provides, for different grid sizes h and interpolation methods to approximate the marker velocity from the Eulerian grids, the errors and the local convergence order of the front-tracking method for initial element sizes $h_e = h/2$. Except for the coarsest grid size with the Peskin interpolation scheme, the three methods converge with a second order accuracy for $\|E_{h_e}\|_1$. The PERM algorithm turns out to be the most accurate interpolation method, with an error up to six to sixteen times smaller than the bilinear and Peskin interpolations, respectively. If the infinity norm $\|E_{h_e}\|_\infty$ is now considered, it is more difficult to discriminate the methods even if the PERM interpolation scheme has, overall, the lowest errors. In terms of convergence order, the Peskin and bilinear methods tends to be second order accurate. For the PERM algorithm, the local convergence order looks erratic but it is globally second order accurate if the whole grid size interval $h \in [1/512; 1/32]$ is taken into account. Generally, the PERM interpolation scheme performs better than the other methods.

Interpolation scheme	PERM [33]		BILINEAR		PESKIN [38]	
$h \equiv 2h_e$	$\|E_{h_e}\|_1$	Order	$\|E_{h_e}\|_1$	Order	$\|E_{h_e}\|_1$	Order
1/16	8.8E-04	-	4.6E-03	-	7.6E-03	-
1/32	1.9E-04	2.2	1.2E-03	2.0	3.0E-03	1.3
1/64	4.7E-05	2.0	2.9E-04	2.0	7.6E-04	2.0
1/128	1.2E-05	2.0	7.3E-05	2.0	1.9E-04	2.0
1/256	2.9E-06	2.0	1.8E-05	2.0	4.8E-05	2.0
1/512	7.2E-07	2.0	4.6E-06	2.0	1.2E-05	2.0
$h \equiv 2h_e$	$\|E_{h_e}\|_\infty$	Order	$\|E_{h_e}\|_\infty$	Order	$\|E_{h_e}\|_\infty$	Order
1/16	2.6E-02	-	1.2E-02	-	2.9E-02	-
1/32	2.4E-03	3.5	4.6E-03	1.4	9.9E-03	1.6
1/64	1.2E-03	1.0	1.6E-03	1.5	3.4E-03	1.5
1/128	1.6E-04	2.9	6.4E-04	1.4	1.4E-03	1.3
1/256	3.9E-05	2.0	1.9E-04	1.8	5.4E-04	1.4
1/512	3.5E-05	0.2	5.1E-05	1.9	1.5E-04	1.9

Table 1: Error $\|E_{h_e}\|_1$ and $\|E_{h_e}\|_\infty$ (Eq. 4 and Eq. 5) and local convergence order (Eq. 6) for different grid or mesh sizes and interpolation methods.

The accuracy of the front-tracking method is now studied as a function of the size of the initial element size h_e, for different Eulerian grid sizes h and the PERM interpolation method (Tab. 2). Whatever the grid size, the error decreases with a second order convergence as long as $h_e \geq h/2$. Once h_e is smaller than $h/2$, no improvement is measured, or even a slight lose of accuracy is observed. Therefore, the best choice for the initial length of the interface elements is $h_e = h/2$, namely elements the size of which evolves roughly in the range $h_e(t) \in [h/4, h]$.

In this section, we showed that the PERM interpolation algorithm is the most accurate method to track the markers position thanks to the velocity staggered grids of the Eulerian field. Likewise, the best discretization choice for the initial interface is to satisfy a two-ratio between the Eulerian grid h and the front-tracking elements h_e.

In the remaining of the paper, the front tracking method based on (i) the PERM algorithm, (ii) initial interface element sizes equal to $h/2$, and (iii) the reseeding algorithm is called in a shortened

h $\Downarrow$	h_e $\Rightarrow$	$h_e = 2h$	$h_e = h$	$h_e = h/2$	$h_e = h/10$	$h_e = h/100$
1/16	$\|E_{h_e}\|_1$	1.7E-02	4.2E-03	8.8E-04	1.2E-03	1.3E-03
	Order	-	2.0	2.2	*-0.2*	*-0.0*
1/32	$\|E_{h_e}\|_1$	4.3E-03	1.5E-03	1.9E-04	3.1E-04	3.2E-04
	Order	-	1.5	3.0	*-0.3*	*-0.0*
1/64	$\|E_{h_e}\|_1$	1.3E-03	3.0E-04	4.7E-05	7.9E-05	8.1E-05
	Order	-	2.1	2.7	*-0.3*	*-0.0*
1/128	$\|E_{h_e}\|_1$	3.6E-04	5.3E-05	1.2E-05	2.0E-05	2.0E-05
	Order	-	2.7	2.2	*-0.3*	*-0.0*
1/256	$\|E_{h_e}\|_1$	7.0E-05	1.7E-05	2.9E-06	4.9E-06	5.1E-06
	Order	-	2.1	2.5	*-0.3*	*-0.0*
1/512	$\|E_{h_e}\|_1$	1.9E-05	3.4E-06	7.2E-07	1.2E-06	1.3E-06
	Order	-	2.5	2.2	*-0.3*	*-0.0*

Table 2: Accuracy of the front-tracking method versus the initial element size h_e - the local convergence order is presented for different element size h_e.

form FTR-PERM method or even simply FTR-PERM. Most of the simulations performed in the rest of the paper are carried out with this FTR-PERM method.

2.2 Volume conservation

Since the interface markers are only advected by the velocity field, the front-tracking method cannot geometrically conserve the volume delimited by the connected elements for incompressible flows as the interface is approximated by linear elements. This issue is inherent to this method, even if the advection algorithm was exact. The calculation of the volume $\mathcal{V}^{(n)}$ is straightforward. Whatever the point O of coordinate $\bar{x}_O$, we have

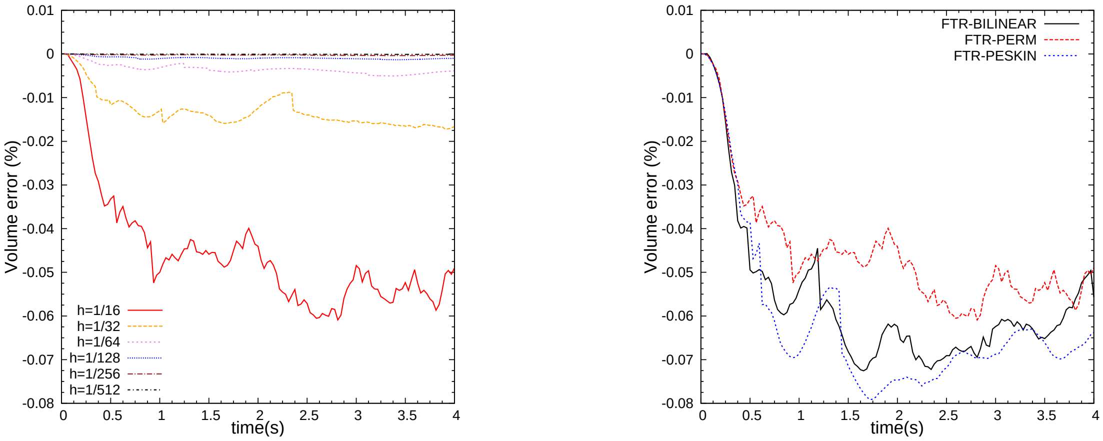
$$\mathcal{V}^{(n)} = \frac{1}{2} \left(\sum_{(m_1,m_2)\in\mathcal{T}^{(n)}} (\bar{x}_{m_1} - \bar{x}_O) \wedge (\bar{x}_{m_2} - \bar{x}_O) \right) .\bar{e}_z \tag{7}$$

where $\bar{e}_z = \bar{e}_x \wedge \bar{e}_y$. Figure 4a presents the evolution of the volume error for the circle advection by a vortex flow for different grid sizes (see Fig. 3 for interface snapshots and (Eq. 3a-3b) for the velocity field) when the PERM algorithm is used for the marker velocity interpolation. We first notice that the volume variation is small, less than 1% for the coarsest grid $h = 1/16$, and it decreases with the grid refinement. Fig. 4b, which compares the evolution of the volume error for the circle advection by a vortex flow for the coarsest grid size when the bilinear, Peskin or PERM methods are used for the calculation of the markers velocity, shows that the volume variation is slightly smaller when the PERM method is used. This assessment confirms the interest of the PERM method versus the two other methods.

Despite the small volume variation induced by our front-tracking approach, it is always better to control the volume at each time step to avoid a too large volume lose or gain for long integration times or very complex incompressible flow fields. In the next subsections, two volume conservation algorithms are presented for incompressible flows and then discussed to show their respective advantages and drawbacks.

2.2.1 Homothetic Rescaling (HR) method – FTR-PERM-HR

The homothetic rescaling method is purely geometric and does not depend on the fluid flow. It consists in performing an homothety of the markers position after the advection step in order to



(a) (b)

Figure 4: Evolution of the volume error for the advection of a circle in a vortex flow (Eq. 3a-3b) by the front tracking method: (a) Volume error as a function of the grid size h when the PERM method is used for the calculation of the velocity marker.(b) Volume error for the grid size $h = 1/16$ when PERM, bilinear or Peskin methods are used.

manage accurately the volume delimited by the interface. For incompressible fluid flows, the volume between two time steps must be preserved, that is $\mathcal{V}^{(n+1)} = \mathcal{V}^{(n)}$ with $\mathcal{V}^{(n)}$ the known volume at time $t^{(n)}$. Once the homothety center $\bar{x}_O$ is chosen, the homothety coefficient $k_{\bar{x}_O}$ is expressed by:

$$k_{\bar{x}_O} = \sqrt{\frac{\mathcal{V}^{(n)}}{\mathcal{V}^{(n+1,adv)}}}$$

where $\mathcal{V}^{(n)}$ and $\mathcal{V}^{(n+1,adv)}$ are expressed by relation (7), for 2-tuples respectively defined at time $t^{(n)}$ and $t^{(n+1)}$, the end of the advection step. Here, index $.^{(n+1,adv)}$ denotes a given quantity at time $n+1$ after Lagrangian advection. Then, for all $m \in \mathcal{M}^{(n,n+1)}$, the coordinate $\bar{x}_m$ of the marker m is adjusted to satisfy the volume conservation as follows:

$$\bar{x}_m^{(n+1)} = \bar{x}_O + k_{\bar{x}_O}(\bar{x}_m^{(n+1,adv)} - \bar{x}_O) \tag{8}$$

At this stage, the location of the homothety center is still free. Obviously, its position plays an important role in the resulting interface shape. One natural choice could be to identify $\bar{x}_O$ to the center of the volume delimited by the interface at the end of the advection step:

$$\bar{x}_O = \frac{\displaystyle\sum_{(m_1,m_2)\in\mathcal{T}^{(n+1)}} [(\bar{x}_{m_1} - \bar{x}_{O'}) \wedge (\bar{x}_{m_2} - \bar{x}_{O'})] \cdot \bar{e}_z \ (\bar{x}_{m_1} + \bar{x}_{m_2} + \bar{x}_{O'})}{\displaystyle\sum_{(m_1,m_2)\in\mathcal{T}^{(n+1)}} 3\,[(\bar{x}_{m_1} - \bar{x}_{O'}) \wedge (\bar{x}_{m_2} - \bar{x}_{O'})] \cdot \bar{e}_z} \tag{9}$$

with $\bar{x}_{O'}$ the coordinate of any point O' and $\bar{e}_z = \bar{e}_x \wedge \bar{e}_y$. Another very simple choice is to use the centroid of the whole markers $\bar{x}_O = \sum_{m\in\mathcal{M}^{(n+1)}} \bar{x}_m / N_{\mathcal{E}}^{(n+1)}$. This latter calculation is adopted in our simulations to locate the homothety center.

The front-tracking method based on the PERM algorithm for the velocity node interpolation and the homothetic rescaling method for the volume conservation is called FTR-PERM-HR.

2.2.2 Velocity Based Correction (VBC) method – FTR-PERM-VBC

The second method differs significantly from the previous one in the sense that it is based on the fluid flow, and more specifically the markers displacement over a time step ($\approx \bar{V}\Delta t$). The first step is to write the volume conservation (see volume calculation (7)) between the two time steps $t^{(n)}$ and $t^{(n+1)}$:

$$\sum_{(m_1,m_2)\in\mathcal{T}^{(n,n+1)}} \left((\bar{x}_{m_1}^{(n+1)} - \bar{x}_O) \wedge (\bar{x}_{m_2}^{(n+1)} - \bar{x}_O)\right).\bar{e}_z = 2\mathcal{V}^{(n)} \tag{10}$$

We can then rewrite relation (10) using the position of the markers at the time $t^{(n+1,adv)}$:

$$\sum_{(m_1,m_2)\in\mathcal{T}^{(n,n+1)}} \left(\left[(\bar{x}_{m_1}^{(n+1)} - \bar{x}_{m_1}^{(n+1,adv)}) + (\bar{x}_{m_1}^{(n+1,adv)} - \bar{x}_O)\right] \wedge \left[(\bar{x}_{m_2}^{(n+1)} - \bar{x}_{m_2}^{n+1,adv}) + (\bar{x}_{m_2}^{(n+1,adv)} - \bar{x}_O)\right] \right).\bar{e}_z = 2\mathcal{V}^{(n)} \tag{11}$$

At this stage, the position of the markers at $t^{(n+1)}$ must be determined in order to maintain the volume $\mathcal{V}^{(n)}$. After the advection step, each marker $m \in \mathcal{M}^{(n+1)}$ is slightly moved in the normal direction to the interface, according to the unit vector $\bar{n}_m^{(n+1)}$: for all $m \in \mathcal{M}^{(n,n+1)}$ (same marker at $t^{(n)}$ and $t^{(n+1)}$),

$$\bar{x}_m^{(n+1)} = \bar{x}_m^{(n+1,adv)} + k\bar{p}_m \tag{12}$$

with $\bar{p}_m \overset{\text{def}}{=} \left|(\bar{x}_m^{(n+1,adv)} - \bar{x}_m^{(n)}) \cdot \bar{n}_m^{(n+1)}\right| \bar{n}_m^{(n+1)}$. The relation (11) is then written as a function of the coefficient k and the displacement $\bar{p}_m$:

$$\sum_{(m_1,m_2)\in\mathcal{T}^{(n,n+1)}} \left(\left[k\bar{p}_{m_1} + (\bar{x}_{m_1}^{(n+1,adv)} - \bar{x}_O)\right] \wedge \left[k\bar{p}_{m_2} + (\bar{x}_{m_2}^{(n+1,adv)} - \bar{x}_O)\right] \right).\bar{e}_z = 2\mathcal{V}^{(n)} \tag{13}$$

The coefficient k is solution of the second order polynomial

$$\begin{aligned} \Big[k^2 & \sum_{(m_1,m_2)\in\mathcal{T}^{(n,n+1)}} \bar{p}_{m_1} \wedge \bar{p}_{m_2} \\ & + k \sum_{(m_1,m_2)\in\mathcal{T}^{(n,n+1)}} (\bar{x}_{m_1}^{(n+1,adv)} - \bar{x}_O) \wedge \bar{p}_{m_2}) + \bar{p}_{m_1} \wedge (\bar{x}_{m_2}^{(n+1,adv)} - \bar{x}_O) \\ & \qquad + \sum_{(m_1,m_2)\in\mathcal{T}^{(n,n+1)}} (\bar{x}_{m_1}^{(n+1,adv)} - \bar{x}_O) \wedge (\bar{x}_{m_2}^{(n+1,adv)} - \bar{x}_O)\Big] \cdot \bar{e}_z = 2\mathcal{V}^{(n)} \end{aligned} \tag{14}$$

The front-tracking method based on the PERM algorithm for the velocity node interpolation and the velocity based correction method for the volume conservation is called FTR-PERM-VBC.

2.2.3 Comments on the volume conservation methods

To study the influence of the volume conservation algorithms, we again consider the advection of a circle in a vortex flow defined by the velocity field given by Eqs. (3a)-(3b) (see Fig. 3). Table 3 provides, for the front-tracking method FTR-PERM, different grid sizes and volume conservation algorithms, the position error and the local convergence order. For the two proposed methods, accounting for the volume conservation does not modify substantially the accuracy of the solution on the coarsest grid: the error $\|E_{h_e}\|_1$, defined by Eq. (4), increases only from 8.8E-04 to 9.2E-04 with the FTR-PERM-HR method whereas it remains unchanged when the FTR-PERM-VBC is used. This is also valid for the refined grid size. The same comments yield also for the errors computed with the infinity-norm, $\|E_{h_e}\|_\infty$. This result is not really surprising because, although the volume is not strictly conserved in time, the front-tracking method turns out to be very accurate with a volume conservation error less than 1% when the volume conservation is deactivated (Fig. 4b). The same conclusion holds for all grids from 1/16 to 1/512.

Figure 5 presents the total computational time required for the advection of a circle in a vortex flow with respect to the grid size h for the FTR-PERM-HR and FTR-PERM-VBC methods. Whereas

FTR method	FTR-PERM		FTR-PERM-HR		FTR-PERM-VBC	
$h \equiv 2h_e$	$\|E_{h_e}\|_1$	Order	$\|E_{h_e}\|_1$	Order	$\|E_{h_e}\|_1$	Order
$1/16$	8.8E-04	-	9.2E-04	-	8.8E-04	-
$1/32$	1.9E-04	2.2	1.9E-04	2.2	1.9E-04	2.2
$1/64$	4.7E-05	2.0	4.7E-05	2.0	4.7E-05	2.0
$1/128$	1.2E-05	2.0	1.2E-05	2.0	1.2E-05	2.0
$1/256$	2.9E-06	2.0	2.8E-06	2.0	2.9E-06	2.0
$1/512$	7.2E-07	2.0	7.2E-07	2.0	7.2E-07	2.0
$h \equiv 2h_e$	$\|E_{h_e}\|_\infty$	Order	$\|E_{h_e}\|_\infty$	Order	$\|E_{h_e}\|_\infty$	Order
$1/16$	2.6E-02	-	2.8E-02	-	2.7E-02	-
$1/32$	2.4E-03	3.5	2.8E-03	3.3	2.5E-03	3.5
$1/64$	1.2E-03	1.0	1.3E-03	1.1	1.3E-03	0.9
$1/128$	1.6E-04	2.9	1.7E-04	2.9	1.7E-04	2.9
$1/256$	3.9E-05	2.0	3.6E-05	2.3	3.9E-05	2.1
$1/512$	3.5E-05	0.2	3.4E-05	0.1	3.5E-05	0.2

Table 3: Error $\|E_{h_e}\|_1$ and $\|E_{h_e}\|_\infty$ (Eq. 4 and Eq. 5) and local convergence order (Eq. 6) for different grid or mesh sizes and without (FTR-PERM) or with the activation of the volume conservation algorithms (FTR-PERM-HR and FTR-PERM-VBC).

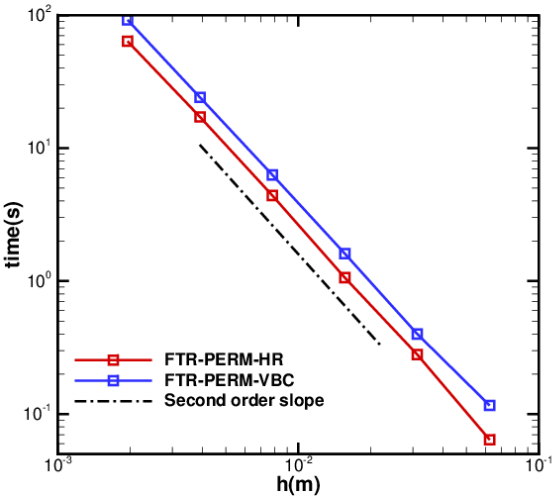


Figure 5: Required total computational time ($t = 4\,s$) for the advection of a circle in a vortex flow (Eq. 3a-3b) when the volume conservation is activated. Comparison between the FTR-PERM-HR and FTR-PERM-VBC

both methods show an increasing computational time proportional to the square of the grid size h, the FTR-PERM-VBC method is slightly more expensive than the FTR-PERM-HR method.

Although both volume conservation methods seem to be similarly efficient, the homothetic rescaling method remains purely geometric, which could lead to different difficulties. For example, if a drop/bubble is close to a solid wall, the homothety may move the interface markers beyond the physical boundaries. A similar behaviour may be encountered between two drops/bubbles, which could induce an unexpected merging process. On the other hand, the velocity based correction method is less sensitive to such a drawback since the markers displacement depends on the local advection velocity. The choice of the homothety center is also a sensitive issue. If the interface is circular and the volume center O is adopted for the homothety center, the correction is the same for all the markers. But for elongated structures where the distance between the markers and O are very different, the homothetic rescaling method may dramatically increase and/or deform the initial shape.

From the above discussion, we prefer to adopt the velocity based correction method to ensure the volume conservation for the advection of the interface, namely the FTR-PERM-VBC method.

2.3 2D topological properties of the interface

One of the challenges in the two-phase flows simulation is the evaluation of the surface tension force which corresponds to the stress jumps accross the interface. The intensity of this vector is proportional to the local curvature of the interface and is directed according to the unit vector normal to the interface.

In this section, two methods are described to calculate geometrical properties of the interface required to obtain the capillary force. Their accuracy is also discussed. The first method relies on the Frenet method which expresses the curvature as the derivative of the tangential unit vector with respect to the curvilinear coordinate and then transforms an integral along a curve into a difference between values expressed at both curve endpoints. The second method consists in looking for circles passing through specific markers. Each of these methods will be compared with respect to an analytical solution.

2.3.1 Frenet based method

Let us define Δs a curve in the 2d plan, with s the curvilinear coordinate, then in the Frenet-Serret frame $(\bar{t}, \bar{n})$, the equality $\partial\bar{t}/\partial s = \kappa\bar{n}$ holds. We denote p_1 and p_2 two points of Δs such that $s(p_2) > s(p_1)$, then the surface tension force along the piece $\Delta s_{p_1p_2}$ of the curve Δs writes [47] with the Frenet based method:

$$\bar{f}_{\Delta s_{p_1p_2}} = \int_{\Delta s_{p_1p_2}} \left(\sigma\kappa\bar{n} + \frac{\partial\sigma}{\partial s}\bar{t}\right) ds = \int_{\Delta s_{p_1p_2}} \frac{\partial\sigma\bar{t}}{\partial s}\, ds = (\sigma\bar{t})_{p_2} - (\sigma\bar{t})_{p_1}$$

where σ is the surface tension coefficient. Assuming a constant σ, we get

$$\int_{\Delta s_{p_1p_2}} \kappa\bar{n}\, ds = \bar{t}_{p_2} - \bar{t}_{p_1} \tag{15}$$

To evaluate the discrete surface tension force at the marker $m \in \mathcal{M}$, a point belonging to $\Delta s_{p_1p_2}$, we consider the two adjacent elements e and $e_{[1]}$ such that $m_{[e,1]} \equiv m_{[e_{[1]},2]} \equiv m$, and the elements center C_e and $C_{e_{[1]}}$ (Fig. 6). Let us choose the point p_2 (resp. p_1) such that its orthogonal projection

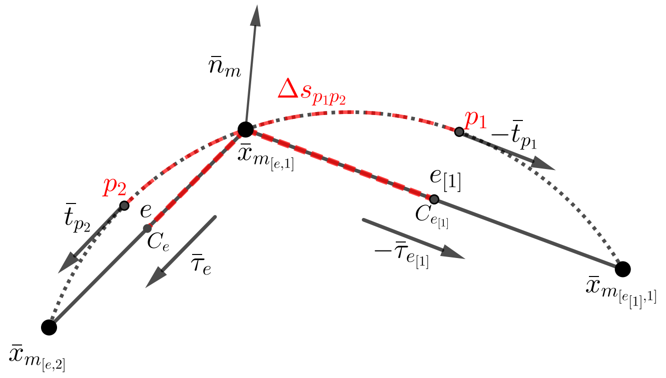


Figure 6: Notations for local curvature and normal estimate using the Frenet based method

over the element e (resp. $e_{[1]}$) is C_e (resp. $C_{e_{[1]}}$), then with notations

$$\bar{\tau}_e \overset{\text{def}}{=} \frac{\bar{x}_{m_{[e,2]}} - \bar{x}_{m_{[e,1]}}}{h_e}, \qquad \bar{\tau}_{e_{[1]}} \overset{\text{def}}{=} \frac{\bar{x}_{m_{[e_{[1]},2]}} - \bar{x}_{m_{[e_{[1]},1]}}}{h_{e_{[1]}}}$$

we get the approximations

$$\bar{t}_{p_2} \approx \bar{\tau}_e, \qquad \bar{t}_{p_1} \approx \bar{\tau}_{e_{[1]}}, \qquad \int_{\Delta s_{p_1 p_2}} \kappa \bar{n} \, ds \approx \kappa_m \bar{n}_m \frac{h_e + h_{e_{[1]}}}{2}$$

Assuming the above approximations mimic the continuous relation (15), we obtain the estimates of the curvature κ_m and unit normal vector $\bar{n}_m$ at the marker m:

$$\kappa_m = 2\frac{\| \bar{\tau}_e - \bar{\tau}_{e_{[1]}} \|}{h_e + h_{e_{[1]}}}, \qquad \bar{n}_m = \frac{\bar{\tau}_e - \bar{\tau}_{e_{[1]}}}{\| \bar{\tau}_e - \bar{\tau}_{e_{[1]}} \|}$$

Here, the tangential vectors $\bar{\tau}_e$ and $\bar{\tau}_{e_{[1]}}$ have been directly calculated on elements e and $e_{[1]}$. However, Febres and Legendre [12] propose a different approach for the calculation of the tangential vectors in the framework of the front-tracking method to compute the surface tension force at the interface which almost leads to the same results.

2.3.2 Circumscribed circle method

The principle of the evaluation of the geometric parameters at the marker $m \in \mathcal{M}$ is based on geometric considerations (Fig. 7). The circumscribed circle (O_m, R_m) passing through three successive markers of two adjacent elements e and $e_{[1]}$, namely $m \equiv m_{[e,1]} \equiv m_{[e_{[1]},2]}$, $m_{[e,2]}$ and $m_{[e_{[1]},1]}$ gives the estimated curvature κ_m and the unit vector normal $\bar{n}_m$ to an approximated interface:

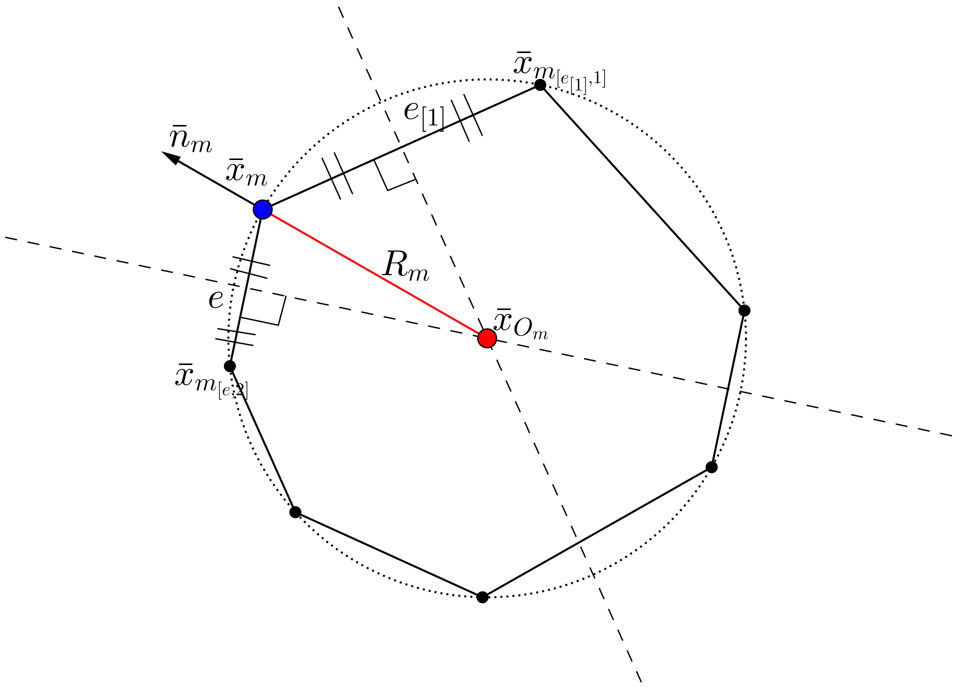


Figure 7: Lagrangian interface points used for the estimate of the curvature at each marker using the circumscribed circle method

$$\bar{n}_m = \frac{\bar{x}_m - \bar{x}_{O_m}}{R_m} \operatorname{sign}[(\bar{x}_m - \bar{x}_{O_m}) \cdot \bar{n}], \qquad \kappa_m = \frac{\operatorname{sign}[(\bar{x}_m - \bar{x}_{O_m}) \cdot \bar{n}]}{R_m}$$

where $R_m \overset{\text{def}}{=} \|\bar{x}_m - \bar{x}_{O_m}\|_2$ is the circumscribed circle radius, sign$[x]$ stands for the sign of the real value x, $\bar{n} \overset{\text{def}}{=} \bar{n}_e + \bar{n}_{e_{[1]}}$ is the sum of the two unit outward vectors normal to e and $e_{[1]}$, and the coordinates $\bar{x}_{O_m}$ are solutions of the linear system:

$$(\bar{x}_{O_m} - (\bar{x}_{m_{[e_{[1]},1]}} + \bar{x}_m)/2) \cdot (\bar{x}_{m_{[e_{[1]},1]}} - \bar{x}_m) = 0$$
$$(\bar{x}_{O_m} - (\bar{x}_{m_{[e,2]}} - \bar{x}_m)/2) \cdot (\bar{x}_{m_{[e,2]}} - \bar{x}_m) = 0$$

It is worth to point out that, with the circumscribed circle approach, κ_m and $\bar{n}_m$ are always both exact as soon as the interface is circular, whatever the size of the elements.

2.3.3 Test case: static curve

To validate the curvatures and normals estimated by the Frenet based method and the circumscribed circle method, we consider an analytical function $f(x) = -x\sin(3\pi x^2/2)$ defined over the range $x \in [0, 1]$. Let us assume a space discretization $\mathcal{D} = (\mathcal{E}, \mathcal{M}, \mathcal{T})$ of the curve $\Gamma = \{M(x, f(x)); x \in [0, 1]\}$. Since Γ is an open curve, the number of markers is no longer $N_{\mathcal{E}}$, but $N_{\mathcal{E}} + 1$. For any marker $m \in \mathcal{M}$, we define $\kappa_{m_{\text{ref}}}$ and $\bar{n}_{m_{\text{ref}}}$ the exact curvature and unit normal vector pointing upward. To measure the error between the estimates and the references, two norms are introduced for the curvature

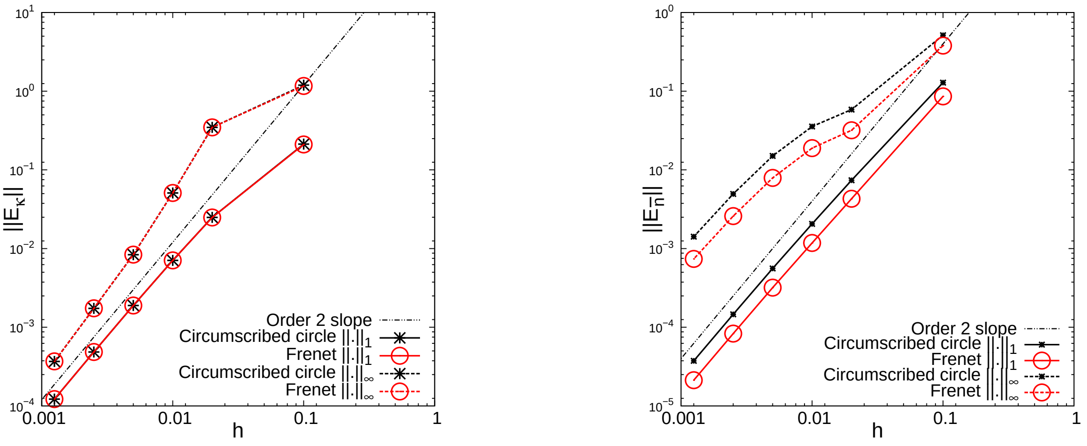
$$\|E_\kappa\|_1 = \frac{\sum_{m\in\mathcal{M}^*} |\kappa_{m_{\text{ref}}} - \kappa_m|}{\sum_{m\in\mathcal{M}^*} |\kappa_{m_{\text{ref}}}|}, \qquad \|E_\kappa\|_\infty = \frac{\max_{m\in\mathcal{M}^*} |\kappa_{m_{\text{ref}}} - \kappa_m|}{\max_{m\in\mathcal{M}^*} |\kappa_{m_{\text{ref}}}|} \tag{16}$$

and the unit normal vector to the interface

$$\|E_{\bar{n}}\|_1 = \sum_{m\in\mathcal{M}^*} \|\bar{n}_{m_{\text{ref}}} - \bar{n}_m\|, \qquad \|E_{\bar{n}}\|_\infty = \max_{m\in\mathcal{M}^*} \|\bar{n}_{m_{\text{ref}}} - \bar{n}_m\| \tag{17}$$

where $\mathcal{M}^* = \mathcal{M} - \{m_0; m_{N_{\mathcal{E}}}\}$, with m_0 and $m_{N_{\mathcal{E}}}$ the two markers of $\mathcal{M}$ of absissa $x_{[m_0]} = 0$ and $x_{[m_{N_{\mathcal{E}}}]} = 1$.

In the first case, the space discretization $\mathcal{D}$ is such that the projection of the whole elements of the set $\mathcal{E}$ provides a uniform partition of the segment $[0; 1]$ located on the x-axis. Figure 8 illustrates the variations of the errors for the curvature and the normal vector, as a function of the size of the projected elements $h = 1/N_{\mathcal{E}}$. For the curvature assessment (Fig. 8a), it is impossible to discriminate between



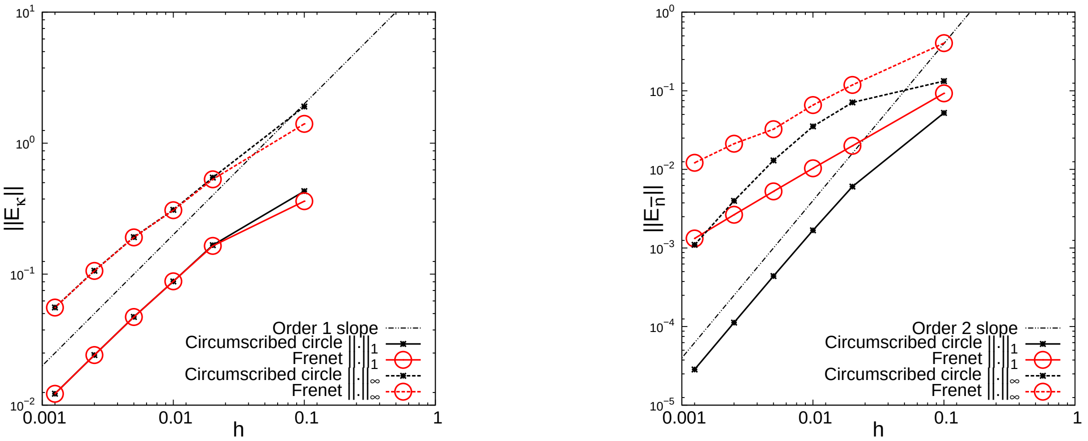
(a) $\|E_\kappa\|_1$ and $\|E_\kappa\|_\infty$

(b) $\|E_{\bar{n}}\|_1$ and $\|E_{\bar{n}}\|_\infty$

Figure 8: Convergence of the curvature and normal vector for the discrete approximation of the curve $f(x) = -x\sin(3\pi x^2/2)$, $x \in [0, 1]$, as a function of the constant grid size $h = 1/N_{\mathcal{E}}$. The sizes of the projected elements on the x-axis are all equal to h.

the two methods, whatever the considered norm. For small enough h-values, both methods turn out to be second order accurate in $\|.\|_1$ and $\|.\|_\infty$ norms. Concerning the normal vector calculation (Fig. 8b), the Frenet based method seems to perform slightly better than the circumscribed circle method. Again, the asymptotic convergence of the errors is second order, as it was for the curvature. To end this test case, it is worth noticing that the mesh of the curve Γ is not uniform, only its projection is. The size of the elements varies discretely as $\sqrt{h^2 + (f(ih) - f((i+1)h))^2}$, with $i = 0, N_{\mathcal{E}} - 1$. Consequently, the sizes of two adjacent elements will be all the more close since $N_{\mathcal{E}}$ will increase.

In the second case, the definition of the space discretization $\mathcal{D}$ for the curve Γ is similar to previously, except that the projection of the elements on the x-axis forms a succession of segments of size $h/2$ and $3h/2$, with $h = 1/N_{\mathcal{E}}$. Therefore, the average size of the elements remains h. Figure 9a shows that the method to estimate the curvature does not matter, except eventually for the coarsest mesh where the Frenet approach is slightly better. For this kind of irregular meshes, the order of convergence is smaller than for the previous test case, since it falls from two to one, only. The major changes are highlighted on the convergence curves for the normal vector estimates (Fig. 9b). Whereas for the circumscribed circle method, the errors are slightly modified between the two test cases, the errors provided by the Frenet based method become significantly larger. With this latter method, the convergence order is then reduced to one.



(a) $\|E_\kappa\|_1$ and $\|E_\kappa\|_\infty$

(b) $\|E_{\bar{n}}\|_1$ and $\|E_{\bar{n}}\|_\infty$

Figure 9: Convergence of errors on the curvature and normal vector for the discrete approximation of the curve $f(x) = -x\sin(3\pi x^2/2)$, $x \in [0,1]$, as a function of the average grid size $h = 1/N_{\mathcal{E}}$. The sizes of the projected elements on the x-axis are successively $h/2$ and $3h/2$.

The two cases indicate that the circumscribed circle method is less sensitive to the mesh discretization of the interface, especially for the calculation of the unit normal vector which seems remaining second order accurate for fine enough meshes. Concerning the curvature, the accuracy of the algorithm is first order, or even second order when the sizes of adjacent elements are close enough. Thus, accounting for the non uniform mesh used in real physical problems, the circumscribed circle method is now adopted to evaluate the surface tension forces.

2.3.4 Test case: circle deformation by a potential flow impinging a flat plane

The aim of this section is to assess the performance of the curvature calculation during the advection of an interface by the FTR-PERM-VBC method. To this end, we consider the deformation of an initial circle, of radius R_0 and center $(x_0; y_0)$, by a potential flow impinging a flat plane:

$$u(x,y) = 2\alpha(x - x_a) \qquad [u] = m/s \tag{18a}$$

$$v(x,y) = -2\alpha(y - y_a) \qquad [v] = m/s \tag{18b}$$

with x_a and y_a the stagnation point coordinates and α the characteristic frequency. Any point $(x(t), y(t))$ of the interface is solution of the equation

$$\left(\frac{x(t)-x_a}{a(t)}\right)^2 + \left(\frac{y(t)-y_c(t)}{b(t)}\right)^2 = 1 \tag{19}$$

with $a(t) = R_0 \exp(2\alpha t)$, $b(t) = R_0 \exp(-2\alpha t)$ and $y_c(t) = y_a + (y_0 - y_a)\exp(-2\alpha t)$. Relation (19) is the equation of an ellipse centered at $(x_a, y_c(t))$ with $a(t)$ and $b(t)$ the half major and minor axes. The following parameters are chosen $x_0 = 0\,m$, $y_0 = 0.75\,m$, $R_0 = 0.15\,m$, $x_a = 1\,m$, $y_a = 0\,m$ and $\alpha = 1\,s^{-1}$.

Table 4 presents the relative difference between the curvature and normal vector estimates with the analytical orders of convergence. At $t = 0.3\,s$, the circle is moderately deformed (Fig. 10) and the solutions are second order accurate with low error levels. For a longer time, at $t = 0.75\,s$, the ellipse is much more flattened with higher curvature zones. At this time, the second order accuracy is lost and the error values are much larger, especially for the curvature approximation and the infinity norm of the normal vector. To understand the reason of these bad results, we now focus our attention on

	Curvature error				Normal error			
t = 0.3 s								
h	$\|E_\kappa\|_1$	Order	$\|E_\kappa\|_\infty$	Order	$\|E_{\bar{n}}\|_1$	Order	$\|E_{\bar{n}}\|_\infty$	Order
1/50	4.9E-03	-	1.1E-02	-	2.2E-03	-	7.9E-03	-
1/100	1.2E-03	2.0	2.8E-03	2.0	5.5E-04	2.0	2.0E-03	2.0
1/200	3.1E-04	2.0	7.1E-04	2.0	1.4E-04	2.0	5.0E-04	2.0
1/400	7.8E-05	2.0	1.8E-04	2.0	3.4E-05	2.0	1.3E-04	2.0
1/800	1.9E-05	2.0	4.4E-05	2.0	8.5E-06	2.0	3.1E-05	2.0
t = 0.75 s								
h	$\|E_\kappa\|_1$	Order	$\|E_\kappa\|_\infty$	Order	$\|E_{\bar{n}}\|_1$	Order	$\|E_{\bar{n}}\|_\infty$	Order
1/50	3.0E-01	-	2.9E-01	-	9.5E-03	-	4.7E-01	-
1/100	2.0E-01	0.6	2.5E-01	0.2	2.9E-03	1.7	2.2E-01	1.1
1/200	1.8E-01	0.1	2.1E-01	0.3	7.5E-04	2.0	1.1E-01	1.0
1/400	1.0E-01	0.8	1.5E-01	0.4	2.5E-04	1.6	3.9E-02	1.5
1/800	6.9E-02	0.6	7.3E-02	1.1	8.1E-05	1.6	1.7E-02	1.2

Table 4: Convergence of the errors on curvatures (Eq. 16) and normal vectors (Eq. 17) with the convergence order (see Eq. 6) for the circle deformation by a potential flow at times $t = 0.3\,s$ and $t = 0.75\,s$

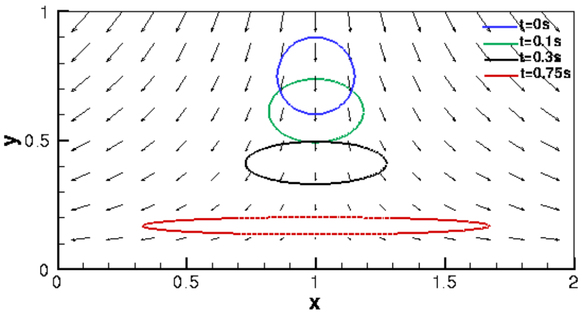


Figure 10: Deformation of a circle by a potential flow (Eq. 18): Initial circular interface (blue) and solutions at $t = 0.1\,s$ (green), $t = 0.3\,s$ (black) and $t = 0.75\,s$ (red).

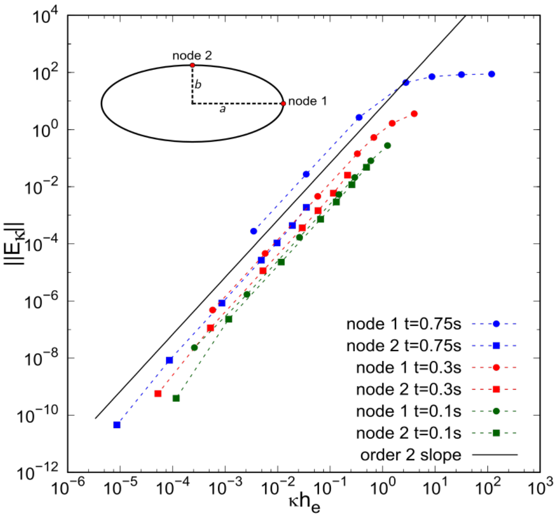


Figure 11: Relative curvature error variation for different local element sizes h_e times the local curvature κ for different ellipses defined by their flattening at different times of simulation ($t = 0.1\,s$, $t = 0.3\,s$ and $t = 0.75\,s$) on two specific points: the node 1 is located in alignment with the major axis and the node 2 is in the alignment of the minor axis, using respectively the circumscribed circle method.

two specific points located at the extremity of the major and minor axes of the ellipse in order to consider both large and small curvatures. The relative curvature errors evaluated at these points, at times $t = 0.1\,s$, $t = 0.3\,s$ and $t = 0.75\,s$, are drawn in Fig. 11 as a function of the dimensionless element size κh_e. This graph clearly shows that the second order accuracy is recovered if $\kappa h_e < 1$, that is if the size of the adjacent elements is at least smaller than the local radius of curvature to the curve. Furthermore, the curvature error is nearly insensitive to the curvature itself from the moment that the element size h_e is scaled by the local curvature κ.

2.4 Comparison with other classical interface tracking methods

The front-tracking method FTR-PERM-VBC developed in the previous sections is now compared to other interface tracking techniques of the literature. The analytical test case is similar to that presented in Sec. 2.1.3. It consists in the tracking of a $0.15\,m$ radius circle initially located at $(0.5\,m; 0.75\,m)$ and advected by a time periodic vortex flow

$$u(x,y) = -2\sin(\pi x)^2 \cos(\pi y) \sin(\pi y) \cos(\pi t/T) \qquad [u] = m/s \qquad (20a)$$

$$v(x,y) = 2\sin(\pi y)^2 \cos(\pi x) \sin(\pi x) \cos(\pi t/T) \qquad [v] = m/s \qquad (20b)$$

The initial circle is then first deformed and stretched by the velocity field up to the half period $T/2$, and then the interface evolves reversely over the next half period until it recovers its initial shape at $t = T$ (Fig. 12).

To perform quantitative comparisons with other tracking methods, generally based on the advection of an indicator function, the exact volume fraction from the discrete interface is calculated. Tables 5 and 6 indicate the average volume fraction errors

$$\|E_C\|_1 = \sum_{i,j} h^2 |C^i(i,j) - C^f(i,j)| \qquad (21)$$

at the end of the time period for $T = 2\,s$ and $T = 8\,s$, for different interface tracking methods of the literature, with $C^f(i,j)$ and $C^i(i,j)$ the final and initial volume fraction values at cell (i,j). Whatever the T-period, the FTR-PERM-VBC method provides always the most accurate volume fraction with

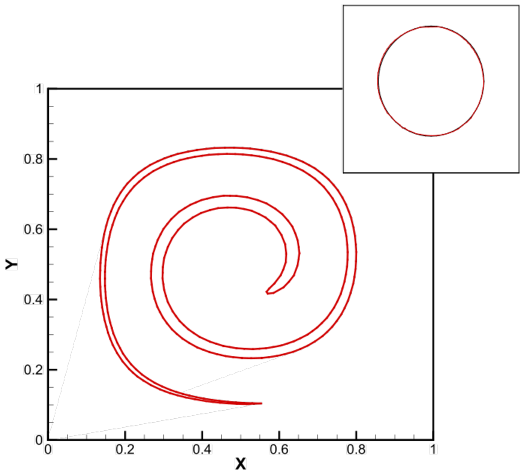


Figure 12: Circle deformation by a periodic vortex flow for $h = 1/32$ and $T = 8\,s$, at times $t = T/2$ (main figure), $t = 0$ and $t = T$ (upper right figure). An enlargement of the interface at $t = 0$ and $t = T$ is provided in Fig. 13a.

Methods	h=1/32	Order	h=1/64	Order	h=1/128
ELVIRA [46]	2.52E-03	1.96	6.46E-04	2.16	1.45E-04
Quadratic fit continuity [46]	1.09E-03	1.96	2.80E-04	2.29	5.72E-05
Quadratic fit/EI-LE [46]	1.88E-03	2.09	4.42E-04	2.24	9.36E-05
Rider-Kothe/Puckett [42]	2.36E-03	2.01	5.85E-04	2.16	1.31E-04
Stream/Puckett [19]	2.37E-03	2.07	5.65E-04	2.10	1.32E-04
FTR-FRONTIER [11]	1.03E-03	2.11	2.39E-04	1.32	9.54E-05
2D conservative FTR [7]	5.67E-03	1.68	1.77E-03	1.26	7.43E-04
FTR-PERM-VBC	2.26E-04	2.05	5.45E-05	1.98	1.38E-05

Table 5: Average errors on the volume fraction (Eq. 21) and convergence order (Eq. 4) for several tracking techniques of the literature – Reversed circle deformation by a vortex flow (Eq. 20) for $T = 2\,s$.

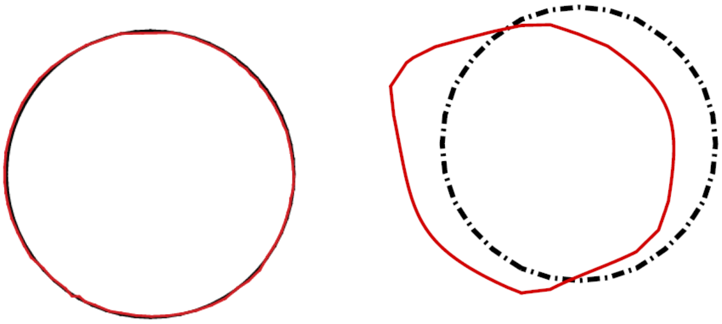
an error often at least one decade smaller than the other approaches. With the grid refinement, our method turns out to be nearly second order accurate. The accuracy of our tracking method is also graphically confirmed by comparing, for the coarse grid $h = 1/32$, the discrepancy between the interface at the end of the time period $t = T = 8\,s$ and at the initial time (Figs. 13a-13d).

In this paragraph, the efficiency of the FTR-PERM-VBC method is studied for the reversed vortex flow, over a complete time period $T = 8\,s$, and compared to the Stream/Youngs [19] and DS-CLVOF [49] methods. Due to the lack of data in these references, the computation times of the Stream/Youngs and DS-CLVOF methods have been estimated equal to the usual VOF-PLIC and LSET-WENO methods we developed some years ago (see Tab. 7 for details). These choices are justified by the following comments: (i) the Stream/Youngs method is based on the VOF-PLIC scheme and (ii) the computational time required for the DS-CLVOF method is essentially driven by the level-set part of the algorithm. All calculations are single-threaded and performed on a Linux system with a Intel Xeon CPU E7-8890 V4 2.2GHz.

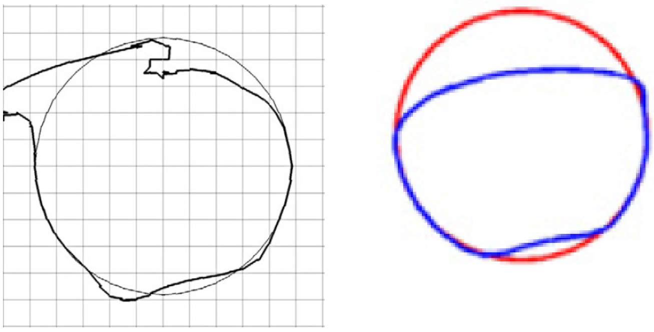
Figure 14a gives, for the FTR-PERM-VBC, Stream/Youngs (VOF-PLIC) and DS-CLSVOF (LSET-WENO) methods, the CPU time required to track the interface from the initial time to $t = T = 8\,s$, as a function of the grid size h. The Stream/Youngs (VOF-PLIC) method reveals to be the cheaper scheme, whatever the grid size. For the coarse grids, the FTR-PERM-VBC method is more expensive than DS-CLSVOF (LSET-WENO) method. However, the computational cost of the FTR-PERM-VBC method grows quadratically with $1/h$, whereas the cost of the two other methods are nearly $\mathcal{O}(1/\mathrm{h}^3)$.

Methods	h=1/32	Order	h=1/64	Order	h=1/128
DS-CLSVOF [49]	5.45E-02	2.38	1.05E-02	2.59	1.74E-03
DS-CLSMOF [24]	2.92E-02	2.41	5.51E-03	2.01	1.37E-03
Stream/Youngs [19]	3.61E-02	1.85	1.00E-02	2.21	2.16E-03
Rider-Kothe/Puckett [42]	4.78E-02	2.78	6.96E-03	2.27	1.44E-03
Stream/Puckett [19]	3.72E-02	2.45	6.79E-03	2.52	1.18E-03
MMCUSL [37]	1.04E-02	2.95	1.34E-03	1.94	3.49E-04
THINC/LS [41]	2.85E-02	3.07	3.39E-03	2.32	6.79E-04
LFRM [48]	-	-	1.23E-02	1.91	3.28E-03
FTR-PERM-VBC	1.14E-03	1.66	3.59E-04	1.77	1.05E-04

Table 6: Average errors on the volume fraction (Eq. 21) and convergence order (Eq. 4) for several tracking techniques of the literature – Reversed circle deformation by a vortex flow (Eq. 20) for $T = 8\,s$.



(a) FTR-PERM-VBC (b) THINC/LS

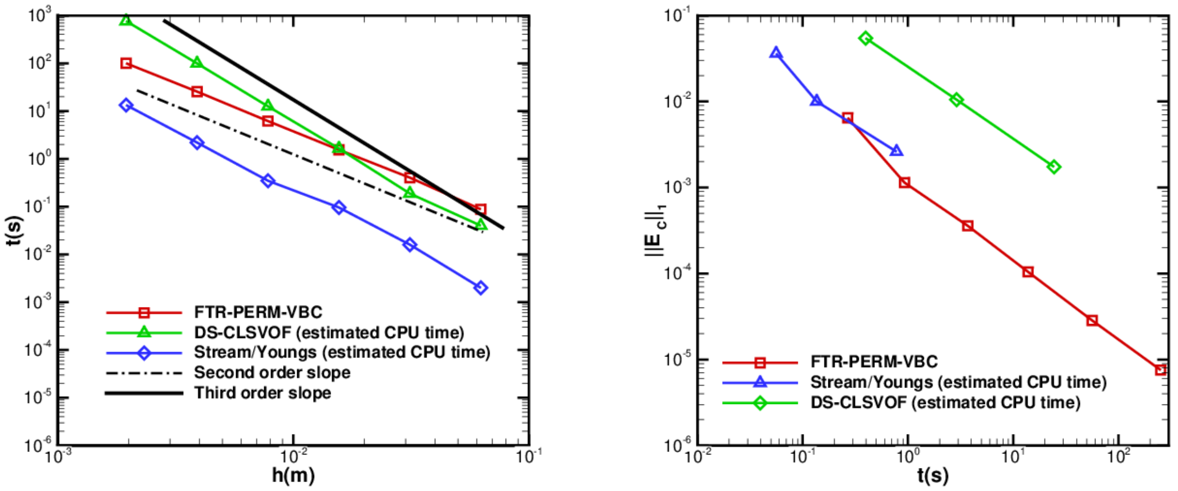


(c) MMCUSL (d) LFRM

Figure 13: Comparison of the initial and final shape of the reversed circle deformation by a vortex flow for the FTR-PERM-VBC and different tracking methods of the literature, $h = 1/32$ and $T = 8\,s$. Methods: (a) FTR-PERM-VBC (b) Coupled THINC and level set method (THINC/LS) [41] (c) Mass and momentum conserving unsplit semi-Lagrangian (MMCUSL) [37] (d) The Local Front Reconstruction Method (LFRM) with the resolution factor $nc = 2$ $i.e$ the ratio between the reconstruction grid and the Eulerian grid resolution [48].

Thus, the computational cost of the FTR-PERM-VBC method becomes all the more competitive that the grid becomes finer.

The efficiency of a numerical method links both the computation cost and the accuracy of the numerical solution. Figure 14b shows the volume fraction error (Eq. 21) as a function of the estimated CPU time for the FTR-PERM-VBC, Stream/Youngs (VOF-PLIC) and DS-CLSVOF (LSET-WENO) methods. Whereas the computation times are always estimates (see here-above), the errors for the



(a) Computational times vs the Eulerian grid size h

(b) Average errors on the volume fraction as a function of the total computational time

Figure 14: Calculation of the reversed circle in a vortex flow (Eq. 20a-20b) over the period $T = 8\,s$ for the FTR-PERM-VBC, Stream/Youngs (VOF-PLIC) and the DS-CLSVOF (LSET-WENO) methods.

Stream/Youngs and DS-CLSVOF methods are respectively extracted from [19] and [49] (and also reported in Tab. 6). The Stream/Youngs (VOF-PLIC) seems to be the most efficient method when a rough solution is simply expected. As soon as a better precision is searched, the FTR-PERM-VBC method becomes the less expensive approach in comparison with the DS-CLSVOF (LSET-WENO) and Stream/Youngs (VOF-PLIC) methods. We can also notice that the CPU time increases linearly with the decrease of the average errors on volume fraction for the FTR-PERM-VBC and the DS-CLSVOF (LSET-WENO) approaches.

Method	Abbreviation	Reference
Volume of Fluid method with PLIC reconstruction	VOF-PLIC	[59, 45]
Level-Set with conservative WENO scheme	LSET-WENO	[36, 35]
Hybrid method of particle/Level-Set coupling	LSET-PART	[52]
Front-tracking without connectivity level contour reconstruction method (and volume conservation every 100 iterations)	FTR-LCRM-VC	[47]
Front-tracking method with PERM reconstruction for velocity nodes calculation (and volume conservation)	FTR-PERM-VBC	Present method

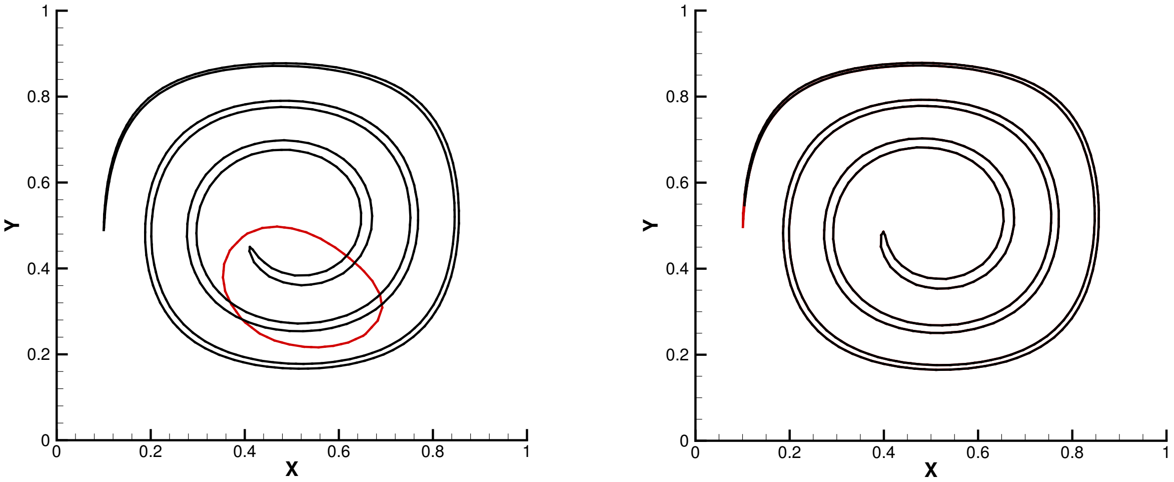
Table 7: Different methods used for comparison tests

Since the Lagrangian front tracking methods are usually considered as the most accurate schemes, the FTR-PERM and FTR-LCRM (resp. FTR-PERM-VBC and FTR-LCRM-VC) methods without (resp. with) the volume conservation algorithms are compared on the steady vortex flow at $t = 4\,s$ (see Eqs. 3a-3b in Sec. 2.1.3). Table 8 sums up the different results. Contrary to the FTR-LCRM method, which loses a lot of accuracy in its volume conservative version FTR-LCRM-VC, the FTR-PERM and FTR-PERM-VBC methods provide identical results. Likewise, although the FTR-LCRM approach is second order accurate beyond $h = 1/32$ ($h = 1/128$ for FTR-LCRM-VC), the errors remain at

	FTR-PERM(-VBC)		FTR-LCRM		FTR-LCRM-VC	
h	$\|E_{h_e}\|_1$	Order	$\|E_{h_e}\|_1$	Order	$\|E_{h_e}\|_1$	Order
$1/16$	8.8E-04	-	7.7E-03	-	3.3E-02	-
$1/32$	1.9E-04	2.2	3.0E-03	1.3	3.1E-02	0.1
$1/64$	4.7E-05	2.0	7.6E-04	2.0	2.4E-02	0.4
$1/128$	1.2E-05	2.0	1.9E-04	2.0	1.7E-02	0.5
$1/256$	2.9E-06	2.0	4.8E-05	2.0	4.4E-03	1.9
$1/512$	7.2E-07	2.0	1.2E-05	2.0	1.1E-03	2.0

Table 8: Average error on the position (Eq. 4) and convergence order (Eq. 6) for the FTR-PERM(-VBC) or FTR-LCRM(-VC) methods (Tab. 7) - Circle deformation by a steady vortex flow (Eqs. 3a-3b)

least one order of magnitude larger than those achieved with FTR-PERM-VBC. The huge worsening of the error when the volume conservation is accounted for in the FTR-LCRM method, namely the FTR-LCRM-VC method, is illustrated by the drawing of the interface shapes of the solution for a coarse grid (Fig. 15a). This illustration clearly indicates that the volume conservation totally destroys the interface shape for the FTR-LCRM method. In contrast, the volume conservative version FTR-PERM-VBC of our front-tracking method FTR-PERM does not significantly alter the shape of the ligament, except at its extremity where some differences can be exhibited (Fig. 15b).



(a) FTR-LCRM (black line) and FTR-LCRM-VC (red line)

(b) FTR-PERM (black line) and FTR-PERM-VBC (red line)

Figure 15: Circle deformation by a steady vortex flow (Eqs. 3a-3b) at $t = 4\,s$, $h = 1/32$. The red (resp. black) interface is obtained with (resp. without) the volume conservation algorithm.

The last test compares the ability of the different methods presented in Tab. 7 to conserve the volume as a function of time for the steady vortex flow (Fig. 16). Since the solutions computed by the FTR-PERM-VBC and FTR-LCRM-VC methods are, by construction, conservative, they are omitted from this study. The FTR-PERM method is the most accurate with a volume variation less than 0.1%. In the whole, the volume is rather well preserved for the front-tracking methods. By decreasing accuracy, the FTR-LCRM, LSET-PART, VOF-PLIC and LSET-WENO methods create volume variations of about 1%, 4%, 10% to $\approx 90\%$ of the initial value.

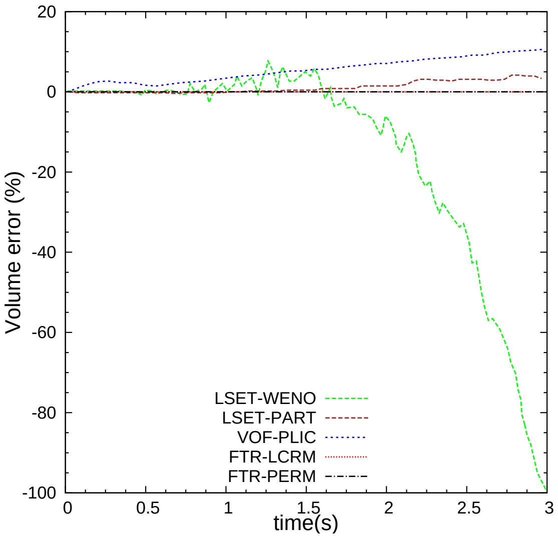


Figure 16: Evolution of the volume error for the advection of a circle in a vortex flow (Eqs. 3a-3b) by different front-tracking methods, $h = 1/64$

To conclude this comparison section, we showed that our front-tracking method FTR-PERM-VBC (i) is very accurate, (ii) converges with a second order slope to the exact solution, (iii) is not degraded by the volume conservation algorithm, (iv) provides always the most accurate results with reasonable CPU times compared to other existing methods of the literature.

3 Interface velocity reconstruction for two-phase flows

The validations performed in the previous sections show the ability of our method to track efficiently interfaces, even for largely stretched and deformed shapes, while keeping the volume constant. We also showed that the most accurate algorithm to calculate the marker velocity from the Eulerian field turns out to be the PERM method. However, the PERM method as well as the bilinear or Peskin methods assume some regularity of the velocity field across the interface, what is clearly questionable for two-phase flows where velocity derivatives are discontinuous due to the jump relations. To overcome this issue, an original velocity reconstruction method based on the physical jump at the interface has been developed and is presented below.

3.1 Reconstruction algorithm

3.1.1 Jump relations

Let us consider an incompressible and immiscible two fluid flow, labeled 1 and 2, separated by an interface Γ, of surface tension σ (Fig. 17).

By assuming a zero interface thickness, the velocity vector $\bar{v}$ must be continuous across interface Γ, whereas the stress force applied by the fluid flow on both sides of the interface, $\bar{\bar{\mathcal{S}}} \cdot \bar{n} \equiv (-p\bar{\bar{\mathcal{I}}} + \mu(\bar{\nabla}\bar{v} + (\bar{\nabla}\bar{v})^t)) \cdot \bar{n}$, is discontinuous [8]. For each point belonging to the interface Σ, we have

$$\bar{v}_1 = \bar{v}_2 \tag{22a}$$

$$\bar{n} \cdot (\bar{\bar{\mathcal{S}}}_1 - \bar{\bar{\mathcal{S}}}_2) \cdot \bar{n} = \sigma(\bar{\nabla} \cdot \bar{n}) \tag{22b}$$

$$\bar{t} \cdot (\bar{\bar{\mathcal{S}}}_1 - \bar{\bar{\mathcal{S}}}_2) \cdot \bar{n} = (\bar{\nabla}\sigma) \cdot \bar{t} \tag{22c}$$

with $\bar{n}$ the unit outward normal vector of the fluid domain 2 and $\bar{t}$ the tangential unit vector to the interface Γ. For three-dimensional problems, Eq. (22c) is a vectorial relation with two components.

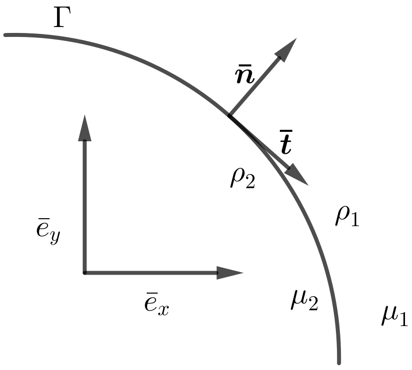


Figure 17: Sketch of a two-fluid interface Γ in a Cartesian coordinate system (x,y). Vectors $\bar{t}$ and $\bar{n}$ are respectively tangent and normal to the interface, the normal pointing toward fluid 1.

Using the Cartesian coordinate system in the standard basis $(\bar{e}_x, \bar{e}_y)$ to express the normal and tangential unit vectors to the interface $\bar{n} = n_x\bar{e}_x + n_y\bar{e}_y$ and $\bar{t} = n_y\bar{e}_x - n_x\bar{e}_y$, and assuming an uniform surface tension $((\bar{\nabla}\sigma)\cdot\bar{t} = 0)$, jump relations (22a–22c) can be reformulated as follows:

$$[v_x]_2^1 = 0, \qquad [v_y]_2^1 = 0 \tag{23a}$$

$$\left[-p + 2\mu\left(\frac{\partial v_x}{\partial x}{n_x}^2 + \left(\frac{\partial v_y}{\partial x} + \frac{\partial v_x}{\partial y}\right)n_x n_y + \frac{\partial v_y}{\partial y}{n_y}^2\right)\right]_2^1 = \sigma\kappa \tag{23b}$$

$$\left[\mu\left(\left(\frac{\partial v_x}{\partial x} - \frac{\partial v_y}{\partial y}\right)n_x n_y + \frac{1}{2}\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right)({n_y}^2 - {n_x}^2)\right)\right]_2^1 = 0 \tag{23c}$$

with $\kappa \equiv \bar{\nabla}\cdot\bar{n}$ the local curvature of the interface and $[X]_2^1 \equiv X_1 - X_2$ denoting the jump of the quantity X across the interface. The principle of our velocity reconstruction method is to evaluate the Lagrangian interface mesh node velocity $\bar{V}_m$, and also the velocity gradients, at node m thanks to the jump conditions (23a)–(23c). Once they are obtained, the interface is advected with these calculated velocities and the different algorithms described in Sec. 2.1 are applied.

The key points of our velocity reconstruction method at the interface node m rely on velocity and pressure approximations based on the staggered grids used for solving the fluid flow equations. These approximations are then introduced into the jump relations (23a)–(23c) to form a set of two algebraic equations whose unknowns are the velocity components of the considered interface node. The different steps to get the marker node velocity are now described.

3.1.2 Linear approximation function for the velocity components

The construction of linear approximation functions for the velocity components in each of the two fluid domains is now detailed thanks to an example. Let us note $V_m^{x,1}$ the unknown horizontal velocity component of the fluid domain 1, expressed at any interface marker $m \in \mathcal{M}$, with coordinates $\bar{x}_m$. To get a local continuous approximation $\tilde{v}^{x,1}(\bar{x})$, a linear interpolation polynomial on a triangular element is constructed using two other Eulerian grid nodes $n_2^{x,1}$ and $n_3^{x,1}$ (Fig. 18). Their coordinates are $\bar{x}_2^{x,1}$ and $\bar{x}_3^{x,1}$ and the related velocities are $v_2^{x,1}$ and $v_3^{x,1}$. They belong to the horizontal velocity components of the fluid domain 1 (Fig. 18). Therefore, the linear approximation of the horizontal component of the velocity simply writes

$$\tilde{v}^{x,1}(\bar{x}) = V_m^{x,1}\psi_m^{x,1}(\bar{x}) + v_2^{x,1}\psi_2^{x,1}(\bar{x}) + v_3^{x,1}\psi_3^{x,1}(\bar{x})$$

The functions $\psi_\alpha^{x,1}(\bar{x})$ are the usual Lagrangian linear polynomials such that $\psi_\alpha^{x,1}(\bar{x}_\beta^{x,1}) = \delta_{\alpha,\beta}$, for $(\alpha, \beta) \in \{m, 2, 3\} \times \{m, 2, 3\}$, with the Kronecker delta $\delta_{\alpha,\beta} = 0$ if $\alpha \neq \beta$ and $\delta_{\alpha,\alpha} = 1$. The variables $v_k^{x,1}$, $k = 2, 3$, are the known horizontal velocities on the horizontal Eulerian grid and $V_m^{x,1}$ is the unknown horizontal velocity of the interface marker m for fluid 1. The choice of the two additional nodes $n_2^{x,1}$ and $n_3^{x,1}$ of the triangle must fulfill some requirements: they must be located (i) in fluid domain 1 and on the horizontal staggered grid, (ii) on either side of the straight line $\Delta = \{\bar{x}_m + \xi\bar{n}_m;\ \xi \in \mathcal{R}\}$ with $\bar{n}_m$ the unit normal vector to the interface at m, (iii) close to m and (iv) so that the triangle of vertices m, $n_2^{x,1}$ and $n_3^{x,1}$ is as equilateral as possible. This last condition is measured by the quality Q of the triangle element which is defined by twice the ratio of the radii between the inscribed and circumscribed circles [43]. For the equilateral triangle, $Q = 1$, and the quality is considered acceptable if $Q \geq 0.1$. The set of triangles $S^{x,1}$ satisfying the conditions (i)-(iv), with $Q \geq 0.1$, gathers the whole set of potential candidates to use for having an accurate linear approximation of the component of the velocity in a given fluid domain. Extra conditions will be later added to select, among all the candidates, the best triangle, for each velocity component in each fluid domain, when the linear system stemming from the jump conditions will be set up.

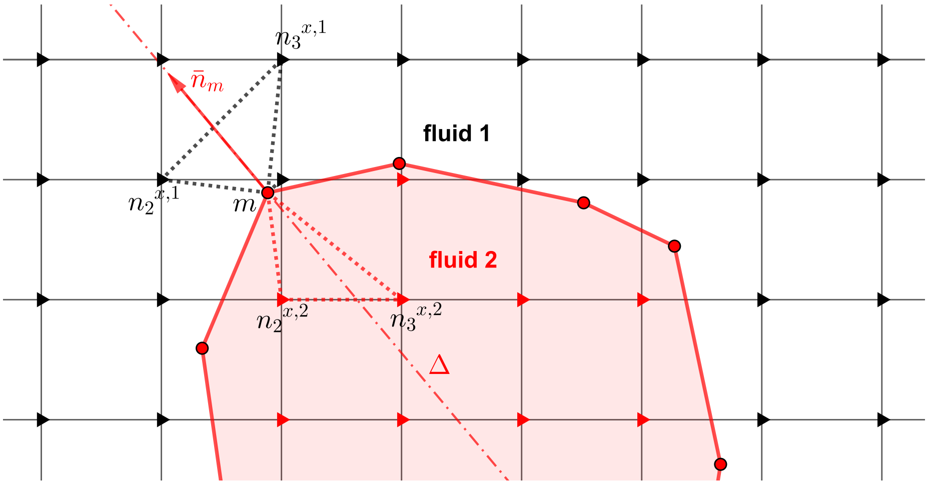


Figure 18: Notations for the calculation of the velocity at any marker $m \in \mathcal{M}$: red (resp. black) triangular symbols are the horizontal velocity nodes of the Eulerian grid belonging to the fluid 2 (resp. 1). The red (resp. black) dotted triangle indicates the support on which the horizontal velocity function is linearly approximated for the fluid 2 (resp. 1).

3.1.3 Pressure approximations at the interface node

Contrary to the velocity which remains still an unknown variable at this stage, the two pressures in the two fluid domains at any interface node $m \in \mathcal{M}$ are directly approximated using the Eulerian nodes of the pressure grid. For each fluid domain, for example fluid 1, this is performed by selecting two grid nodes $n_2^{p,1}$ and $n_3^{p,1}$, at coordinates $\bar{x}_2^{p,1}$ and $\bar{x}_3^{p,1}$ and located (i) in the correct fluid domain on the pressure staggered grid (domain 1 in this example), (ii) on either side of the straight line $\Delta = \{\bar{x}_m + \xi\bar{n}_m;\ \xi \in \mathcal{R}\}$ and (iii) closest to the interface node m. Let us note p_2^1 and p_3^1 the pressures at $n_2^{p,1}$ and $n_3^{p,1}$. Then the pressure p_m^1 at the marker m is simply equal to the interpolated pressure at the intersection point n_I^1 whose coordinate is $\bar{x}_I^1$, located at the intersection between $\Delta = \{\bar{x}_m + \xi\bar{n}_m;\ \xi \in \mathcal{R}\}$ and $(n_2^{p,1} n_3^{p,1})$, namely

$$p_m^1 = \frac{\|\bar{x}_2^{p,1} - \bar{x}_I^1\| p_3^1 + \|\bar{x}_3^{p,1} - \bar{x}_I^1\| p_2^1}{\|\bar{x}_3^{p,1} - \bar{x}_2^{p,1}\|}$$

Fig. 19 illustrates the pressure approximations at m with the representation of all the grid pressure nodes used for the calculation.

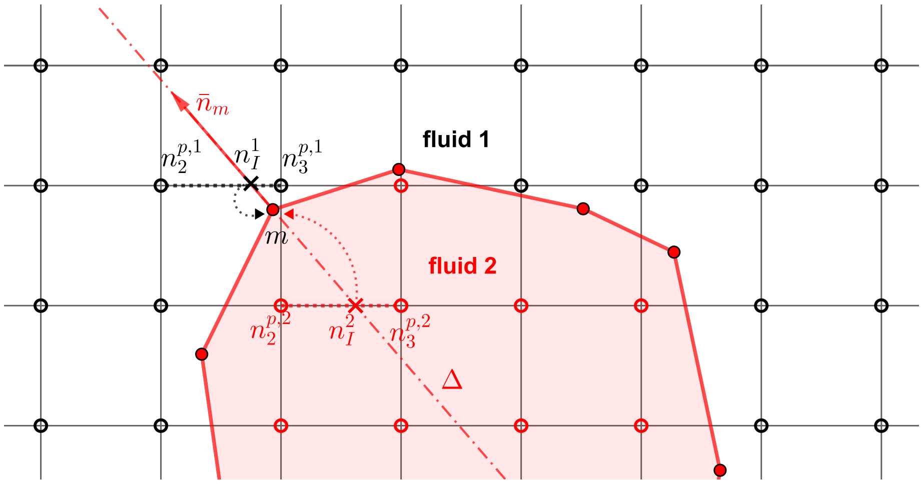


Figure 19: Notations for the calculation of the pressure at any marker $m \in \mathcal{M}$: empty red (resp. black) open circular symbols are the pressure nodes of the Eulerian grid belonging to the fluid 2 (resp. 1). The two symbols connected by the red (resp. black) dotted line are used to interpolate the pressure at the × symbol, and then at the marker position m, for the fluid 2 (resp. 1).

3.1.4 Reconstruction of the velocity at the interface marker m

In this section, the jump relations (23a)–(23c) are enforced by using the linear approximations of the velocity components and the pressure approximations. The first condition (23a) is easily satisfied by imposing the equality $V_m^x \overset{\text{def}}{=} V_m^{x,1} = V_m^{x,2}$ and $V_m^y \overset{\text{def}}{=} V_m^{y,1} = V_m^{y,2}$. Let $\psi_i^{x,k}(\bar{x})$ (resp. $\psi_i^{y,k}(\bar{x})$), $i \in \{m, 2, 3\}$, the i^{th} linear Lagrangian polynomial on the triangle be defined by nodes m, $n_2^{x,k}$ and $n_3^{x,k}$ (resp. m, $n_2^{y,k}$ and $n_3^{y,k}$), in the fluid domain k. With similar notations, $v_i^{x,k}$ (resp. $v_i^{y,k}$) is the horizontal (resp. vertical) velocity component on the Eulerian grid at nodes $n_i^{x,k}$ (resp. $n_i^{y,k}$), $i = 2, 3$, located in the fluid domain k. Substituting the linear interpolations and the pressure approximations into the jump relations (23b)–(23c), we get the following linear system:

$$A(1,1)V_m^x + A(1,2)V_m^y = B(1) \tag{24a}$$

$$A(2,1)V_m^x + A(2,2)V_m^y = B(2) \tag{24b}$$

with

$$A(1,1) = 2\left(\sum_{k=1}^{2}(-1)^k \mu_k \left(n_x^2 \partial_x \psi_m^{x,k} + n_x n_y \partial_y \psi_m^{x,k}\right)\right)$$

$$A(1,2) = 2\left(\sum_{k=1}^{2}(-1)^k \mu_k \left(n_x n_y \partial_x \psi_m^{y,k} + n_y^2 \partial_y \psi_m^{y,k}\right)\right)$$

$$A(2,1) = \left(\sum_{k=1}^{2}(-1)^k \mu_k \left(n_x n_y \partial_x \psi_m^{x,k} + \frac{n_y^2 - n_x^2}{2} \partial_y \psi_m^{x,k}\right)\right)$$

$$A(2,2) = \left(\sum_{k=1}^{2}(-1)^k \mu_k \left(- n_x n_y \partial_y \psi_m^{y,k} + \frac{n_y^2 - n_x^2}{2} \partial_x \psi_m^{y,k}\right)\right)$$

$$B(1) = -\sum_{k=1}^{2}(-1)^k \left(-p_m^k + 2\mu_k \sum_{i=2}^{3} \left(n_x^2 v_i^{x,k} \partial_x \psi_i^{x,k} + n_x n_y \left(v_i^{y,k} \partial_x \psi_i^{y,k} + v_i^{x,k} \partial_y \psi_i^{x,k}\right) + n_y^2 v_i^{y,k} \partial_y \psi_i^{y,k}\right)\right) - \sigma\kappa$$

$$B(2) = -\sum_{k=1}^{2}(-1)^k \mu_k \sum_{i=2}^{3} \left(n_x n_y \left(v_i^{x,k} \partial_x \psi_i^{x,k} - v_i^{y,k} \partial_y \psi_i^{y,k}\right) + \frac{n_y^2 - n_x^2}{2} \left(v_i^{x,k} \partial_y \psi_i^{x,k} + v_i^{y,k} \partial_x \psi_i^{y,k}\right)\right)$$

As described in Sec. 3.1.2, four triangle sets $S^{x,1}$, $S^{y,1}$, $S^{x,2}$ and $S^{y,2}$ have been defined to gather the potential candidates for the linear approximations, for x- and y-velocity components and the two fluid domains. The four finally retained triangles, each one of them belonging to the different sets $S^{x,1}$, $S^{y,1}$, $S^{x,2}$ and $S^{y,2}$, are determined by identifying the combination which provides the smallest condition number for the linear system (24a-24b), $\mathrm{cond}(A) = \lambda_M/\lambda_m$, with λ_M and λ_m the maximal and minimal singular values of A (eigenvalues of A^tA).

3.2 Validation: Hadamard-Rybczynski fluid flow solution

To assess the accuracy of the velocity reconstruction method, we consider the Hadamard-Rybczynski solution which consists in a spherical drop or bubble (fluid 2) moving in another fluid labelled 1 by buoyancy in the gravity field $\bar{g} = -g\bar{e}_y$. We assume that the flow is steady, incompressible and governed by the Stokes equations. The sphere is non-deformable with a fixed radius $r = a$, the surface tension is constant and the jump conditions (23a)–(23c) must be fulfilled. To preserve the sphericity of the interface, the pressure in Eq. (23b) is the hydrostatic pressure. The Stokes equations are solved in the reference frame of the sphere, with spherical coordinates (r, θ, Φ). Since the flow is axi-symmetric with respect to the vertical direction ($\theta = 0$), all the azimuthal derivatives are zero as well as the azimuthal component of the velocity. The length, velocity and pressure are respectively scaled by a, $ga^2(\rho_1 - \rho_2)/\mu_1$ and $\rho_1 ga$. Then, the dimensionless velocity and pressure of fluid k writes:

$$v_{r,k}(r,\theta) = -2\cos\theta\left(\frac{A_k}{r^3} + \frac{B_k}{r} + C_k + D_k r^2\right) \tag{26a}$$

$$v_{\theta,k}(r,\theta) = \sin\theta\left(\frac{-A_k}{r^3} + \frac{B_k}{r} + 2C_k + 4D_k r^2\right) \tag{26b}$$

$$p_k(r,\theta) = p_{0,k} - \frac{\rho_k}{\rho_1} r\cos\theta - \frac{\mu_k}{\mu_1}\left(1 - \frac{\rho_2}{\rho_1}\right)\left(\frac{2B_k}{r^2} + 20D_k r\right)\cos\theta \tag{26c}$$

with $p_{0,2} - p_{0,1} = 2\sigma/(\rho_1 g a^2)$. The expressions for the surrounding fluid 1 write

$$A_1 = \frac{\mu_2}{4(\mu_1+\mu_2)}V, \qquad B_1 = -\frac{2\mu_1 + 3\mu_2}{4(\mu_1+\mu_2)}, \qquad C_1 = \frac{V}{2}, \qquad D_1 = 0$$

whereas those for fluid 2 making up the drop or bubble give

$$A_2 = 0, \qquad B_2 = 0, \qquad C_2 = -\frac{\mu_1}{4(\mu_1+\mu_2)}V, \qquad D_2 = \frac{\mu_1}{4(\mu_1+\mu_2)}V$$

with

$$V = \frac{1}{3}\left(\frac{\mu_1+\mu_2}{\mu_1 + 3\mu_2/2}\right)$$

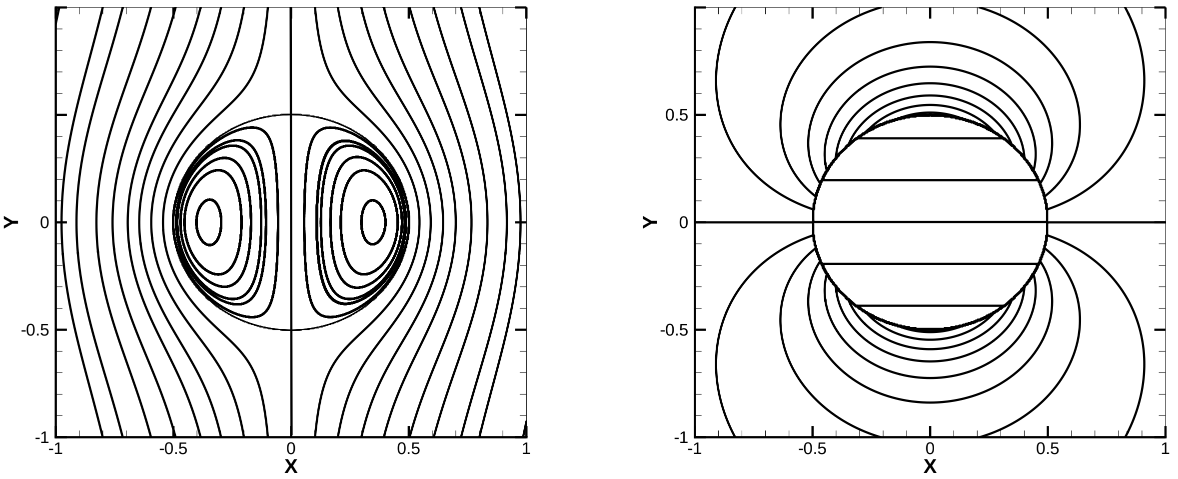
the dimensionless buoyancy velocity.

Calculations were carried out with two fluid couples spanning the viscosity ratio from 0.01 to 100, the density ratio from 0.001 to 1000 and the surface tension from 0.022 to 0.075 N/m (Tab. 9).

Fluids	Air/water	Air/Oil	Oil/air	Water/air
μ_2/μ_1	0.01	0.1	10	100
ρ_2/ρ_1	0.001	0.0015	680	1000
$\sigma(N/m)$	0.075	0.022	0.022	0.075

Table 9: Physical characteristics for the Hadamard-Rybczynski flow

The streamlines drawn in Fig. 20 for the Water/air couple ($\mu_2/\mu_1 = 100$) show two large symmetrical eddies inside the drop as well as the deviation of the outer flow around the drop. The isobars are also



(a) (b)

Figure 20: Streamlines (a) and pressure isobars (b) for the Hadamard-Rybczynski solution - Water/air couple (Tab. 9), **x-y** plane for Stokes flow, viscosity ratio $\mu_2/\mu_1 = 100$

plotted, proving linear variation inside fluid 2 and more complex behaviour outside the drop/bubble with downstream overpressure and underpressure upstream.

Since the analytical velocity is known, it is straightforward to compare the numerically obtained velocities to the exact solution at the interface markers. Table 10 gives the relative error on velocity norm calculation as a function of the grid size h ($h_e = h/2$)

$$\|E\|_1 = \frac{\sum_{m\in\mathcal{M}} |\|\bar{V}_m\| - \|\bar{v}(\bar{x}_m)\||}{\sum_{m\in\mathcal{M}} \|\bar{v}(\bar{x}_m)\|} \tag{27}$$

using either the PERM method, well adapted to track fronts in an homogeneous fluid domain, or the so-called JUMP method designed to take into account the stress discontinuities at the interface between two immiscible fluids. The results indicate that both velocity reconstruction methods provide convergent solutions. However, the convergence order of the velocity jump based method is one order better than that calculated from the PERM algorithm. When the viscosity ratio is less than one, the relative errors are mostly small. Except for the coarsest interface mesh, the JUMP method is more accurate, till one order of magnitude lower for the finest mesh. For viscosity ratios larger than one, the error obtained with the PERM algorithm get dramatically worse, all the more since this ratio increases. This error remains very high, about 20% for the finest grid when $\mu_2/\mu_1 = 100$. On the other hand, those computed with our method turns out to be completely insensitive to μ_2/μ_1, decreasing from ≈ 10% to ≈ 0.02% with the refinement of the grid size.

This test case clearly indicates that the velocity jump based method is much more adapted to the interface velocity calculation than the PERM method when two-phase flows are considered with large fluid property contrasts.

4 Conclusions and perspectives

In two-dimensions, a front-tracking method has been formulated and validated against various kinematic test cases with success. For the first time, a complete accuracy and convergence analysis of front-tracking methods has been investigated. Among the different algorithms considered in the present work, it appears that using the PERM method [33] for markers advection, together with velocity based

	$\mu_2/\mu_1 = 0.1$				$\mu_2/\mu_1 = 0.01$			
$h = 2h_e$	PERM		JUMP		PERM		JUMP	
	$\Vert E\Vert_1$	Order	$\Vert E\Vert_1$	Order	$\Vert E\Vert_1$	Order	$\Vert E\Vert_1$	Order
1/16	5.1E-02		1.2E-01		5.6E-02		1.2E-01	
1/32	3.1E-02	0.7	2.6E-02	2.2	3.4E-02	0.7	2.7E-02	2.2
1/64	1.3E-02	1.2	9.7E-03	1.4	1.5E-02	1.2	1.3E-02	1.1
1/128	7.5E-03	0.9	2.2E-03	2.1	8.2E-03	0.9	2.8E-03	2.2
1/256	3.6E-03	1.0	6.0E-04	1.9	4.0E-03	1.0	7.2E-04	1.9
1/512	1.9E-03	0.9	1.6E-04	1.9	2.1E-03	0.9	1.9E-04	1.9

	$\mu_2/\mu_1 = 10$				$\mu_2/\mu_1 = 100$			
$h = 2h_e$	PERM		JUMP		PERM		JUMP	
	$\Vert E\Vert_1$	Order	$\Vert E\Vert_1$	Order	$\Vert E\Vert_1$	Order	$\Vert E\Vert_1$	Order
1/16	5.6E-01		1.1E-01		6.0E+00		1.3E-01	
1/32	3.1E-01	0.9	4.1E-02	1.5	3.4E+00	0.8	4.5E-02	1.5
1/64	1.4E-01	1.2	1.1E-02	1.9	1.5E+00	1.2	1.2E-02	1.9
1/128	7.5E-02	0.9	2.3E-03	2.2	8.3E-01	0.9	2.8E-03	2.1
1/256	3.6E-02	1.0	6.7E-04	1.8	4.0E-01	1.0	8.3E-04	1.8
1/512	1.9E-02	0.9	1.6E-04	2.0	2.1E-01	0.9	2.0E-04	2.1

Table 10: Relative velocity errors at the interface (Eq. 27) and convergence order (Eq. 6) as a function of the grid size h for the Hadamard-Rybczynski fluid flow solution. Different viscosity ratios (μ_2/μ_1) have been considered (Tab. 9).

volume conservation and circumscribed circle curvature and normal calculation, provides a second order convergence in space of the method whatever the test case considered, as soon as the size of the Lagrangian surface mesh elements is at least twice smaller than the local curvature radius of the interface. Compared to previously published front capturing VOF and Level Set techniques or Level Contour front-tracking method, our FTR-PERM-VBC approach performs always better, with error levels at least 10 times smaller than other methods on interface position for a circle stretched in vortex flow. In addition, our method requires less CPU time than the other interface tracking methods considered in this paper to reach a given accuracy.

In a second part of the paper, real multiphase flows have been considered with viscosity ratios different than one at the interface, together with surface tension effects. In these problems, the velocity gradients are no more continuous across interface. We have proposed a new interface velocity reconstruction based on the use of the jump relations valid at the interface. By using the analytical solution of Hadamard-Rybczynski, we have been able to demonstrate that even the second order PERM approach was wrong or less accurate compared to the jump relation based velocity interpolation of the markers, as soon as the viscosity ratio at the interface was far from 1. These new results are very encouraging for future simulations of real two-phase flows with front-tracking methods.

Ongoing works and perspectives are the extension of the method to three dimensions. The main differences between two- and three-dimensional formulations of the method concern mesh management and curvature/normal estimates. Indeed, in 3D, on one hand, smoothing and swapping procedures are required to ensure a good quality of the interfacial mesh. On the other hand, the curvature and normal calculations are much more subtle and very sensitive to mesh properties. These works have been first presented in [3]. To finish with, in two-dimensions, the coupling between the front tracking method using jump relations for marker velocities and the Navier-Stokes discretization is under investigation. Our main idea is to suppress all density and viscosity averages in the cells cut by the interface (one-fluid model of [26]) and use directly the marker velocities and gradients in the finite volume fluxes coming from the momentum equation approximation.

5 Acknowledgements

This work was granted access to the HPC resources of IDRIS, CINES and CCRT under the allocation A0032b06115 proposed by GENCI (Grand Equipement National de Calcul Intensif). All the numerical methods presented in the present work are part of home made code FUGU, developed in MSME laboratory of Marne-La-Vallée University. FUGU code is patented as digital work by SATT Ile de France and UPEM University under n°IDDN . FR . 001 . 480009 . 000 .S.P. 2017 . 000 . 10000.

References

[1] J. W. Barrett, H. Garcke, and R. Nurnberg. Phase field models versus parametric front tracking methods: Are they accurate and computationally efficient? *Communications in Computational Physics*, 15(2):506–555, 2014.

[2] J. Bell, P. Colella, and H. Glaz. A second-order projection method for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 85(2):257 – 283, 1989.

[3] D.A. Koffi Bi, M. Tavares, E. Chénier, and S. Vincent. A review of geometrical interface properties for 3D Front-Tracking methods. *5th Turbulence and Interactions conference, Les Trois-Ilets, West French Indies, France*, 2018.

[4] F. Bierbrauer and S.P. Zhu. A numerical model for multiphase flow based on the GMPPS formulation. Part I: Kinematics. *Computers and Fluids*, 36:1199–1212, 2007.

[5] V. Le Chenadec and H. Pitsch. A monotonicity preserving conservative sharp interface flow solver for high density ratio two–phase flows. *Journal of Computational Physics*, 249:185–203, 2013.

[6] G. Compère, E. Marchandise, and J.-F. Remacle. Transient adaptivity applied to two-phase incompressible flows. *Journal of Computational Physics*, 227:1923–1942, 2008.

[7] M. De-kang. Towards front-tracking based on conservation in two space dimensions ii, tracking discontinuities in capturing fashion. *Journal of Computational Physics*, 226(2):1550 – 1588, 2007.

[8] J.M. Delhaye. Jump conditions and entropy sources in two-phase systems. Local instant formulation. *International Journal of Multiphase Flow*, 1(3):395 – 409, 1974.

[9] F. Denner and B. G.M. van Wachem. Compressive VOF method with skewness correction to capture sharp interfaces on arbitrary meshes. *Journal of Computational Physics*, 279:127 – 144, 2014.

[10] O. Desjardins, V. Moureau, and H. Pitsch. An accurate conservative level set/ghost fluid method for simulating turbulent atomization. *Journal of Computational Physics*, 227:8395–8416, 2008.

[11] J. Du, B. Fix, J. Glimm, X. Jia, X. Li, Y. Li, and L. Wu. A simple package for front tracking. *Journal of Computational Physics*, 213(2):613 – 628, 2006.

[12] M. Febres and D. Legendre. Enhancement of a 2D front-tracking algorithm with a non-uniform distribution of Lagrangian markers. *Journal of Computational Physics*, 358:173 – 200, 2018.

[13] S. Ganesan and L. Tobiska. An accurate finite element scheme with moving meshes for computing 3D-axisymmetric interface flows. *International Journal for Numerical Methods in Fluids*, 57(2):119–138, 2008.

[14] S. Ganesan and L. Tobiska. Arbitrary Lagrangian-Eulerian finite-element method for computation of two-phase flows with soluble surfactants. *Journal of Computational Physics*, 231(9):3685 – 3702, 2012.

[15] J. Glimm, M.J. Graham, J. Grove, X.L. Li, T.M. Smith, D. Tan, F. Tangerman, and Q. Zhang. Front tracking in two and three dimensions. *Computers & Mathematics with Applications*, 35(7):1 – 11, 1998. Advanced Computing on Intel Architectures.

[16] J. Glimm, J. Grove, B. Lindquist, O. A. McBryan, and G. Tryggvason. The bifurcation of tracked scalar waves. *SIAM Journal on Scientific and Statistical Computing*, 9(1):61–79, 1988.

[17] N. Grenier, M. Antuono, A. Colagrossi, D. Le Touzé, and B. Alessandrini. An Hamiltonian interface SPH formulation for multi-fluid and free surface flows. *Journal of Computational Physics*, 228(22):8380 – 8393, 2009.

[18] F.H. Harlow and J.E. Welch. Numerical calculation of time-dependent viscous incompressible flow with fluid with free surface. *Physics of Fluids*, 8:2182–2189, 1965.

[19] Dalton J.E. Harvie and David F. Fletcher. A new Volume of Fluid advection algorithm: The stream scheme. *Journal of Computational Physics*, 162(1):1 – 32, 2000.

[20] M. Herrmann. On simulating primary atomization using the refined level set grid method. *Atomization and Sprays*, 21:283–301, 2011.

[21] E. Ishii, T. Ishikawa, and Y. Tanabe. Hybrid particle/grid method for predicting motion of micro- and macro–free surfaces. *Journal of Fluid Engineering*, 128:921–930, 2006.

[22] C. B. Ivey and P. Moin. Conservative and bounded volume-of-fluid advection on unstructured grids. *Journal of Computational Physics*, 350:387 – 419, 2017.

[23] D. Jamet, O. Lebaigue, N. Coutris, and J.M. Delhaye. The second gradient method for the Direct Numerical Simulation of liquid - vapor flows with phase change. *Journal of Computational Physics*, 169(2):624 – 651, 2001.

[24] M. Jemison, E. Loch, M. Sussman, M. Shashkov, M. Arienti, M. Ohta, and Y. Wang. A coupled Level Set-Moment of Fluid Method for incompressible two-phase flows. *Journal of Scientific Computing*, 54(2):454–491, Feb 2013.

[25] M. Jemison, M. Sussman, and M. Shashkov. Filament capturing with the Multimaterial Moment-of-Fluid method. *Journal of Computational Physics*, 285:149–172, 2015.

[26] I. Kataoka. Local instant formulation of two-phase flow. *International Journal of Multiphase Flow*, 12(5):745–758, 1986.

[27] F. Kummer and T. Warburton. Patch-recovery filters for curvature in Discontinuous Galerkin-based Level-Set methods. *Communications in Computational Physics*, 19(2):329–353, 2016.

[28] D. Legendre and J. Magnaudet. The lift force on a spherical bubble in a viscous linear shear flow. *Journal of Fluid Mechanics*, 368:81 – 126, 1998.

[29] J. Liu, S. Koshisuka, and Y. Oka. A hybrid particle–mesh method for viscous, incompressible, multiphase flows. *Journal of Computational Physics*, 202:65–93, 2005.

[30] M. S. Longuet-Higgins and E. D. Cokelet. The deformation of steep surface waves on water. I. A numerical method of computation. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 350(1660):1–26, 1976.

[31] J. Lopez, J. Hernandez, P. Gomez, and F. Faura. An improved PLIC–VOF method for tracking thin fluid structures in incompressible two–phase flows. *Journal of Computational Physics*, 208:51–74, 2005.

[32] X. Lv, Q. Zou, D.E. Reeve, and Y. Zhao. A preconditioned implicit free-surface capture scheme for large density ratio on tetrahedral grids. *Communications in Computational Physics*, 11(1):215–248, 2012.

[33] R. McDermott and S.B. Pope. The parabolic edge reconstruction method (PERM) for Lagrangian particle advection. *Journal of Computational Physics*, 227(11):5447 – 5491, 2008.

[34] J. O'Rourke. *Computational Geometry in C:*. Cambridge University Press, Cambridge, 2 edition, 10 1998.

[35] S. Osher and R. Fedkiw. Level Set Methods: An Overview and Some Recent Results. *Journal of Computational Physics*, 169(2):463 – 502, 2001.

[36] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12 – 49, 1988.

[37] M. Owkes and O. Desjardins. A mass and momentum conserving unsplit semi-Lagrangian framework for simulating multiphase flows. *Journal of Computational Physics*, 332:21 – 46, 2017.

[38] C. Peskin. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25(3):220 – 252, 1977.

[39] S. Popinet. An accurate adaptive solver for surface–tension–driven interfacial flows. *Journal of Computational Physics*, 228:5838–5866, 2009.

[40] S. Popinet and S. Zaleski. A front–tracking algorithm for accurate representation of surface tension. *International Journal for Numerical Methods in Fluids*, 30:775–793, 1999.

[41] L. Qian, Y. Wei, and F. Xiao. Coupled THINC and level set method: A conservative interface capturing scheme with high-order surface representations. *Journal of Computational Physics*, 373:284 – 303, 2018.

[42] W. J. Rider and D. B. Kothe. Reconstructing volume tracking. *Journal of Computational Physics*, 141(2):112–152, 1998.

[43] I. Roghair, M. Van Sint Annaland, and J.A.M. Kuipers. An improved Front-Tracking technique for the simulation of mass transfer in dense bubbly flows. *Chemical Engineering Science*, 152:351 – 369, 2016.

[44] L. Scarbolo, D. Molin, P. Perlekar, M. Sbragaglia, A. Soldati, and F. Toschi. Unified framework for a side-by-side comparison of different multicomponent algorithms: Lattice Boltzmann vs. phase field model. *Journal of Computational Physics*, 234:263 – 279, 2013.

[45] R. Scardovelli and S. Zaleski. Direct Numerical Simulation of Free-Surface and Interfacial Flow. *Annual Review of Fluid Mechanics*, 31:567–603, January 1999.

[46] R. Scardovelli and S. Zaleski. Interface reconstruction with least-square fit and split Eulerian-Lagrangian advection. *International Journal for Numerical Methods in Fluids*, 41(3):251–274, 2003.

[47] S. Shin and D. Juric. Modeling Three-Dimensional Multiphase Flow Using a Level Contour Reconstruction Method for Front Tracking without Connectivity. *Journal of Computational Physics*, 180(2):427 – 470, 2002.

[48] S. Shin, I. Yoon, and D. Juric. The Local Front Reconstruction Method for direct simulation of two- and three-dimensional multiphase flows. *Journal of Computational Physics*, 230(17):6605 – 6646, 2011.

[49] M. Sussman and E. G. Puckett. A Coupled Level Set and Volume-of-Fluid method for computing 3D and axisymmetric incompressible two-phase flows. *Journal of Computational Physics*, 162(2):301 – 337, 2000.

[50] M. Sussman, P. Smereka, and S. Osher. A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow. *Journal of Computational Physics*, 114(1):146 – 159, 1994.

[51] G. Tomar, D. Fuster, S. Zaleski, and S. Popinet. Multiscale simulations of primary atomization using Gerris. *Computers and Fluids*, 39:1864–1874, 2010.

[52] P. Trontin, S. Vincent, J.L. Estivalezes, and J.P. Caltagirone. A subgrid computation of the curvature by a particle/level-set method. Application to a front-tracking/ghost-fluid method for incompressible flows. *Journal of Computational Physics*, 231(20):6990 – 7010, 2012.

[53] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W.Tauber, J.Han, S.Nas, and Y.-J. Jan. A Front-Tracking Method for the Computations of Multiphase flow. *Journal of Computational Physics*, 169(2):708 – 759, 2001.

[54] G. Tryggvason, R. Scardovelli, and S. Zaleski. *Direct numerical simulations of gas-liquid multiphase flows*. Cambridge University Press, 2011.

[55] S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 100(1):25 – 37, 1992.

[56] S. Vincent, G. Balmigère, J.-P. Caltagirone, and E. Meillot. Eulerian–Lagrangian multiscale methods for solving scalar equations - Application to incompressible two–phase flows. *Journal of Computational Physics*, 229:73–106, 2010.

[57] S. Vincent and J.-P. Caltagirone. A one–cell local multigrid method for solving unsteady incompressible multiphase flows. *Journal of Computational Physics*, 163:172–215, 2000.

[58] B. Xie and F. Xiao. Toward efficient and accurate interface capturing on arbitrary hybrid unstructured grids: The THINC method with quadratic surface representation and gaussian quadrature. *Journal of Computational Physics*, 349:415 – 440, 2017.

[59] D. Youngs. Time-dependent multi-material flow with large fluid distortion. *Numerical methods for fluid dynamics*, 24(2):273–285, 1982.

[60] D. Zuzio and J.-L. Estivalezes. An efficient block parallel AMR method for two phase interfacial flow simulations. *Computers and Fluids*, 44:339–357, 2011.

[61] D. Zuzio, J.-L. Estivalezes, and B. Di Pierro. An improved multiscale Eulerian-Lagrangian method for simulation of atomization process. *Computers and Fluids*, 176:285–301, 2018.