Journal of Computational Physics 229 (2010) 5788–5802

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Implicit tracking for multi-fluid simulations

A. Villa [a,*], L. Formaggia [b]

[a] Universitá degli Studi di Milano, Department of Mathematics, Via Saldini 50, Milano, Italy
[b] MOX, Politecnico di Milano, Department of Mathematics, Via Bonardi 9, Milano, Italy

ARTICLE INFO

Article history:
Received 21 October 2009
Received in revised form 8 April 2010
Accepted 12 April 2010
Available online 18 April 2010

Keywords:
Level set
Volume of fluid
Multi-fluid simulations

ABSTRACT

In this work, we introduce a new method for tracking the interfaces among several immiscible fluids. To advance the solution in time, we use a MUSCL-type scheme which couples a partial volume representation with a level set one to build the numerical fluxes. In particular we show the positiveness and conservation properties of this method. Some numerical tests are given to demonstrate the conservativeness and the performances of our method.

1. Introduction

In this paper, we describe a multi-fluid tracking method where the fluids are transported by a given velocity field. This method can be used on both structured and unstructured meshes, it has good conservation properties and it provides a quick and smooth reconstruction of the interfaces. Finally, it is robust with respect to topological changes.

Many techniques regarding the two-fluid problem are reported in the literature but they cannot often be extended readily to a multi-fluid problem and, moreover, they do not match our requirements listed above. Tracking methods can be roughly subdivided into two categories: the Lagrangian and Eulerian ones. The former track the interfaces explicitly, while the latter reconstruct them with a post-processing procedure.

Among the many Lagrangian tracking algorithms (see, for instance, [5,16,22,32]), some move all the nodes of the volume mesh. Some others, on the contrary, track just the interface points and reconstruct the mesh in the interior at every time step, or whenever necessary. The Lagrangian approach presents some difficulties, particularly in three-dimensional computations, such as the treatment of possible, physical or numerically induced, topological changes. Furthermore, sophisticated adaption algorithms have to be used to preserve the mesh quality and, if a topological change occurs, complex topology correction algorithms are needed (see, for instance, the one in [18]). Moreover it is almost impossible to prove the algorithm robustness with respect to all topological changes that may happen in complex and realistic 3D situations. Some works, for example [26,31], have been proposed to tackle the topological change problem. These approaches adopt a fixed background grid for the solution of a geometry regularization equation. However, they apply only to two-fluid simulations and are not mass-preserving.

Though the Lagrangian methods have an explicit and immediate representation of the interfaces (see [12,16,22,32]), they are not conservative. There are works, like [17], which present procedures to enforce mass conservation, but fail to be robust for topological changes.

* Corresponding author. Present address: ENEA – Ricerca sul Sistema Elettrico (ERSE), Via Rubattino 54, Milano, Italy.
E-mail addresses: andrea.villa@unimi.it (A. Villa), luca.formaggia@polimi.it (L. Formaggia).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.04.020

The complexity of Lagrangian methods triggered the development of the Eulerian implicit tracking methods; an overview can be found in [6,13]. Let us remind the most effective methods, namely, the volume of fluid (VOF) and the level-set (LS). Some mixed Eulerian–Lagrangian methods exist, such as the ALE methods [8] or the particle methods [19], but often they are not robust with respect to topological changes. The LS [9,19,25,27,28] is a robust method and it is easy to code. But, in many cases, it does not fit the multi-fluid framework and, in general, it does not conserve the mass. A few papers are devoted to the multi-fluid simulation (see, for instance, [11,33,34]) however, the first two works are specifically designed for curvature-driven flows and they cannot be easily adapted to an advective-driven case, where the velocity field is given. In [34] a nested LS structure is used. However, though effective, this technique is not mass conservative.

Many works are devoted to fix the LS non conservativeness, like [21,29], yet all of them consider only the case of two fluids. The mass conservation issue can be partially solved by refining the grid adaptively, as pointed out in [1,2].

VOF methods are mass conservative by construction and relatively robust, although they are usually designed to track only two fluids (see, for instance, [23]) and furthermore they have, in general, an irregular reconstruction of the interfaces. Moreover, VOF methods often require a structured grid. Interface reconstruction, using VOF methods, is a major topic and many works, such as [3,4] are devoted to it.

To overcome the drawbacks of LS and VOF methods many coupled techniques have been developed. For instance in [10] a coupled LS-particle approach is used. Though effective and nearly conservative, this scheme does not seem to be readily applicable to the multi-fluid case. In [21,29,30] some coupled LS–VOF methods have been developed. These schemes are designed to combine the best features of the two methods. However, these coupling schemes usually rely on VOF structured grids and they cannot be easily extended to unstructured grids. One of the most applicable techniques for multi-fluid simulations is the partial volume tracking method (VT) which consists in discretizing, with high order schemes, the volume transport equation. This approach is very similar to the VOF one, however there is no explicit reconstruction of the interfaces. The VOF can be seen as a sub-case of the VT, and in many cases the VOF method itself is called VT method (see [6]). This approach has a moderate success, since the discontinuous initial solutions are quickly diffused even if high resolution methods are used.

In this work, we are interested in exploiting both the partial volume representation and the level set representation. A simple coupling strategy is used: the level set is defined as the first degree interpolation on the dual grid of the partial volumes. This coupling enables, with the solution of only one set of partial differential equations, to access to both the VT and LS representations. Our LS functions are usually steeper than the signed distance functions and a proper finite volume scheme is provided to cope with this kind of steep LS functions.

Let us now briefly review the structure of the paper: in Section 2 we give an outline of our method, in Sections 3 and 4 we analyze the method and we outline some of its properties. In Section 5 we give some details of the numerical techniques adopted and we also analyze theoretically the method performances. In Section 6 we introduce a proper reconstruction algorithm for the level set function and in the last section we describe some numerical results and we show our method performances both in the two-fluid case and in the multi-fluid case.

2. The method

We consider a domain $\Omega \subset \mathbb{R}^d$, where $d = 1, 2, 3$, with a regular boundary $\partial\Omega$; this domain is filled with n_s immiscible fluid species, such that every subdomain $\Omega_i \subset \Omega$, corresponding to a species, does not overlap with the others, i.e., $\Omega_i \cap \Omega_j = 0$ if $i \neq j$ and $\overline{\Omega} = \bigcup_{i=0}^{n_s} \overline{\Omega}_i$, see Fig. 1. The subdomains Ω_i depend on time, i.e., $\Omega_i = \Omega_i(t)$, since they are advected by a time dependent velocity field $\vec{u}(t, \vec{X})$, where $\vec{X} \in \Omega(t)$ and $t \geqslant 0$, whose trace on $\partial\Omega$ has zero normal component, i.e., $\vec{u} \cdot \vec{n} = 0$ on $\partial\Omega$, being $\vec{n}$ the boundary normal unit vector. Other boundary conditions can be considered, however, for the sake of simplicity only the free slip condition is adopted in this work. We also assume that $\vec{u}$ is sufficiently regular, in particular we assume, at each time t, $\vec{u}(t, \vec{X}) \in H^1(\Omega)$. For $i = 0, \ldots, n_s$ we define $\lambda_i^0 \in L^2(\Omega)$ as the characteristic function of subdomain Ω_i at initial time, i.e.

$$\lambda_i^0(\vec{X}) = \begin{cases} 1, & \text{if } \vec{X} \in \Omega_i(t), \\ 0, & \text{if } \vec{X} \notin \Omega_i(t). \end{cases}$$

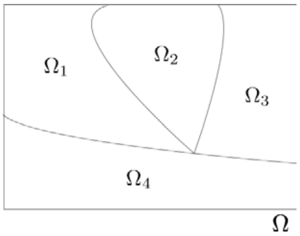

Fig. 1. Domain Ω and its subdomains Ω_i.

Therefore, the following relation holds: $\sum_{i=1}^{n_s}\lambda_i^0 = 1$ almost everywhere in Ω. The VT equation for a given vector field $\vec{u}$ is

$$\begin{cases} \frac{\partial\lambda_i}{\partial t} + \vec{\nabla}\cdot(\lambda_i\vec{u}) - \lambda_i(\vec{\nabla}\cdot\vec{u}) = 0, & t > 0,\ i = 1,\ldots,n_s, \\ \lambda_i = \lambda_i^0, & t = 0, \end{cases} \tag{1}$$

where $\lambda_i(t,\cdot) \in L^2_\Omega$ is a weak solution of (1). This equation is equivalent to the transport equation

$$\frac{\partial\lambda_i}{\partial t} + \vec{u}\cdot\vec{\nabla}\lambda_i = 0. \tag{2}$$

We have considered Eq. (1) instead of (2) because its corresponding discrete form is simpler to analyze.

Problem (1) has some properties we wish to recall.

Proposition 1. *If the initial condition satisfies* $\sum_{i=1}^{n_s}\lambda_i^0 = 1$ *almost everywhere in* Ω, *then* $\sum_{i=1}^{n_s}\lambda_i = 1$ *almost everywhere in* Ω *for* $\forall t > 0$.

Proof. We can proceed formally by summing up the *i*th equations in (1), obtaining

$$\frac{\partial}{\partial t}\left(\sum_{i=1}^{n_s}\lambda_i\right) + \vec{\nabla}\cdot\left(\vec{u}\sum_{i=1}^{n_s}\lambda_i\right) - (\vec{\nabla}\cdot\vec{u})\sum_{i=1}^{n_s}\lambda_i = 0. \tag{3}$$

By inspection it can be verified that $\sum_{i=1}^{n_s}\lambda_i = 1,\ \forall t \geqslant 0$, is a solution of (3) and that it satisfies the initial condition. Since (3) is a linear advection equation it has a unique solution. □

Proposition 2. *If the initial condition satisfies* $0 \leqslant \lambda_i^0 \leqslant 1$ *almost everywhere in* Ω *and the velocity field* $\vec{u}(t,\vec{X})$ *is Lipschitz continuous uniformly on t for* $\vec{X} \in \Omega$, *then* $0 \leqslant \lambda_i \leqslant 1$, $\forall t > 0$, *almost everywhere in* Ω.

Proof. It is a standard characteristic theory argument, see [14]. □

Let us now consider the definition of a level set description of the same subdomains. We define $\phi_i : \mathbb{R}^+ \times \overline{\Omega} \to \mathbb{R}$, with $\phi_i(t,\cdot) \in \mathbb{C}^0(\overline{\Omega})\quad \forall t > 0,\ i = 1,\ldots,n_s$, some level set functions such that $\Omega_i(t) = \{\vec{X} \in \overline{\Omega} : \phi_i(t,\vec{X}) > \frac{1}{2}\}$ and consequently $\partial\Omega_i(t) = \{\vec{X} \in \overline{\Omega} : \phi_i(t,\vec{X}) = \frac{1}{2}\}$. This particular value of the set has been chosen because it will be useful when we find an analogy between the discrete forms of LS and the VT equations. We can write the following evolution equation for each ϕ_i

$$\begin{cases} \frac{\partial\phi_i}{\partial t} + \vec{\nabla}\cdot(\phi_i\vec{u}) - \phi_i(\vec{\nabla}\cdot\vec{u}) = 0, & t > 0, \\ \phi_i = \phi_i^0, & t = 0, \end{cases} \tag{4}$$

by which, at all times, $\lambda_i = H(\phi_i - \frac{1}{2})$, where

$$H(\varrho) = \begin{cases} 1 & \text{if } \varrho > 0, \\ 0 & \text{otherwise} \end{cases}$$

is the Heaviside function and ϕ_i^0 is the initial condition. In other words, at the continuous level, Eqs. (1) and (4) are two equivalent ways to describe the interface motion. However, in the discrete setting, we will use two different spaces for λ_i and ϕ_i.

We now introduce the discrete form of the equations: let $\mathscr{T}_\Delta$ be a conforming (structured or unstructured) grid on Ω made of either simplex or quad elements. The grid $\mathscr{T}_\Delta$ has n_e elements indicated by e_r, $r = 1,\ldots,n_e$ and n_p nodes denoted by $\vec{x}_k,\ k = 1,\ldots,n_p$. Let Δ be the maximum diameter of the elements. Consider the dual mesh made of $n_c = n_p$ cells τ_k, $k = 1,\ldots,n_c$, centered on the nodes $\vec{x}_k$, and built by connecting the barycenters of the elements to the barycenters of the faces, see Fig. 2. Let $\mathbb{I}_k = \{k_j,\ j = 1,\ldots,|\mathbb{I}_k|\}$ be the set of the indexes of the cells surrounding cell τ_k, and let $\{\tau_{k_j}\},\ j = 1,\ldots,|\mathbb{I}_k|$, be the set of cells surrounding τ_k. The common surface between τ_k and τ_{k_j} is indicated by l_k^j. We also indicate by $\imath$ the index such that, given the indexes k and j, $\imath : l_{k_j}^\imath = l_k^j$. In other words, every interface between the cells τ_k and τ_{k_j} can be identified by two indices j and $\imath$, respectively, depending on whether it is a face of τ_k or τ_{k_j}: see Fig. 2.

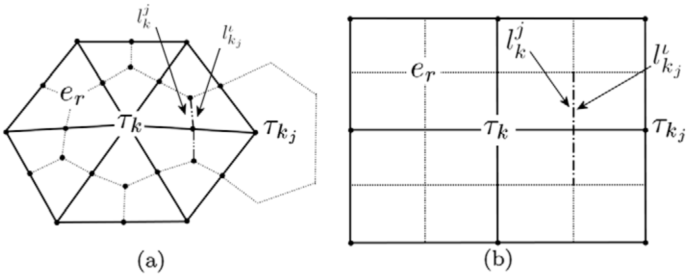

Fig. 2. An example of unstructured (a), and structured (b), two-dimensional meshes with the dual meshes (dotted). The *jth* neighboring cell of τ_k is τ_{k_j}, the common interface between τ_k and τ_{k_j} is called l_k^j. There exists a $\imath$ such that the $\imath$th interface of τ_{k_j} is equal to l_k^j.

For the sake of clarity, in this work we will adopt the following convention: the index i always refers to the fluid species, k to the cell related quantities, j to the interface related values, r to the elements, and n to the time steps. Let us introduce the semi-discrete counterparts of λ_i and ϕ_i denoted by $\lambda_{i,\Delta}(t) \in \mathbb{V}^0$ and $\phi_{i,\Delta}(t) \in \mathbb{V}^1$, respectively, where $\mathbb{V}^0 = \{\lambda \in L^2(\Omega) : \lambda|_{\tau_k} \in \mathbb{P}^0(\tau_k),\ k = 1, \dots, n_c\}$, $\mathbb{V}^1 = \{\phi \in \mathbb{C}^0(\Omega) : \phi|_{e_r} \in \mathbb{Q}^1(e_r),\ r = 0, \dots, n_c\}$ in the case of a rectangular grid and $\mathbb{V}^1 = \{\phi \in \mathbb{C}^0(\Omega) : \phi|_{e_r} \in \mathbb{P}^1(e_r),\ r = 0, \dots, n_c\}$ on a simplicial mesh. Here $\mathbb{P}^s(\Omega)$ denotes the space of polynomials of order at most s on Ω and $\mathbb{Q}^s(\Omega)$ is the space of the tensor product of polynomials of order at most s. We consider the canonical basis $\{\vartheta_k^0\}$ for $\mathbb{V}^0$ and $\{\vartheta_k^1\}$ for $\mathbb{V}^1$, therefore

$$\lambda_{i,\Delta}(t, \vec{X}) = \sum_{k=1}^{n_p} \lambda_{i,k}(t)\vartheta_k^0(\vec{X}), \quad \phi_{i,\Delta}(t, \vec{X}) = \sum_{k=0}^{n_c} \phi_{i,k}(t)\vartheta_k^1(\vec{X}), \tag{5}$$

where $\lambda_{i,k}$ is the mean volume fraction of the species i in the cell τ_k (we will denote, from now on, $\lambda_{i,k}$ as the composition) and $\phi_{i,k}$ are the values of the discrete level set function at node $\vec{x}_k$.

We introduce a rather simple coupling between LS and VT equations, by choosing the level set function as the piecewise linear interpolation from the dual mesh to the original one, i.e., $\phi_{i,\Delta} = \mathbf{I}_\Delta^1 \lambda_{i,\Delta}$ where $\mathbf{I}_\Delta^1 : \mathbb{V}^0 \to \mathbb{V}^1$ is the linear interpolation operator on the $\mathscr{T}_\Delta$ grid. In other terms, we set

$$\phi_{i,k} = \lambda_{i,k} \quad k = 1, \dots, n_p, \quad i = 1, \dots, n_s. \tag{6}$$

The evolution of the interfaces is carried out through the advancement of $\lambda_{i,\Delta}$ by a discrete version of (1), using the information carried by $\phi_{i,\Delta}$ to build the numerical fluxes. Finally, we reconstruct the level set as a post-processing. Therefore, with the solution of only a set of PDE's, we get both the partial volume and the level set representations. This choice implies an error concerning the representation of the initial conditions as, in general, given an initial condition $\lambda_{i,\Delta}(0, \vec{X})$ then $\lambda_{i,\Delta}(0, \vec{X}) \neq H(\phi_{i,\Delta}(0, \vec{X}) - \frac{1}{2})$. This difference can be bounded as we state in the following:

Proposition 3. *Let us assume that* $\lambda \in \mathbb{V}^0$ *and has the image in the set {1, 0}. Moreover, let us consider* $\phi = \mathbf{I}_\Delta^1 \lambda$. *Then*

$$\int_\Omega \left(\lambda - H\left(\phi - \frac{1}{2}\right)\right) = O(\Delta).$$

Proof. Let $S_b = \{k \in [1, n_c] : \int_{\tau_k}(\lambda - H(\phi - \frac{1}{2})) \neq 0\}$. Since this set is also the set of the cells that are crossed by the boundary of Ω_i, its cardinality is $O(\Delta^{1-d})$. Moreover $\int_{\tau_k}(\lambda - H(\phi - \frac{1}{2})) = O(\Delta^d)\ \forall k \in S_b$, therefore $\int_\Omega(\lambda - H(\phi - \frac{1}{2})) = O(\Delta^{1-d})O(\Delta^d)$ and we obtain the thesis. □

We use a finite volume method together with an explicit Euler scheme to advance $\lambda_{i,\Delta}$

$$\lambda_{i,k}^{n+1} = \left(1 + D_{\Delta,k}^n\right)\lambda_{i,k}^n - \sum_{j=1}^{|\Box_k^C|} F_{i,k}^{n,j}, \tag{7}$$

where $\lambda_{i,k}^n = \lambda_{i,k}(t^n)$ and $t^0, t^1, \dots, t^n, t^{n+1}, \dots$ is a sequence of time steps with $t^{n+1} = t^n + \Delta t^n$. The quantity $D_{\Delta,k}^n = \sum_{j=1}^{|\Box_k^C|} v_k^{n,j}$ is the dimensionless discrete divergence factor of element τ_k (i.e., $D_{\Delta,k}^n$ is the discrete approximation of $\frac{\Delta t^n}{|\tau_k|}\oint_{\partial\tau_k} \vec{u} \cdot \vec{n}$) and

$$v_k^{n,j} = \frac{\Delta t^n}{|\tau_k|} \int_{l_k^j} \vec{u} \cdot \vec{n}$$

is a dimensionless quantity which can be considered as the interface Courant number. Finally,

$$F_{i,k}^{n,j} = v_k^{n,j} \Phi(\widehat{\lambda}_{i,k}^{n,j}, \widehat{\lambda}_{i,k_j}^{n,\imath}) \tag{8}$$

are the interface fluxes, where $\Phi(\widehat{\lambda}_{i,k}^{n,j}, \widehat{\lambda}_{i,k_j}^{n,\imath})$ is the upwind function

$$\Phi(\widehat{\lambda}_{i,k}^{n,j}, \widehat{\lambda}_{i,k_j}^{n,\imath}) = \begin{cases} \widehat{\lambda}_{i,k}^{n,j} \text{if} v_k^{n,j} \geqslant 0, \\ \widehat{\lambda}_{i,k_j}^{n,\imath} \text{if} v_k^{n,j} < 0, \end{cases} \tag{9}$$

where $\widehat{\lambda}_{i,k}^{n,j}$, $\widehat{\lambda}_{i,k_j}^{n,\imath}$ are suitable approximations of the composition $\lambda_{i,\Delta}^n$ on the faces l_k^j and $l_{k_j}^\imath$, respectively: see Fig. 3.

The stability of method (7) entails the following time step restriction:

$$\Delta t^n \leqslant \frac{|\tau_k|}{|\Box_k|} \frac{1}{|\int_{l_k^j} \vec{u} \cdot \vec{n}|} \quad k = 1, \dots, n_c,\ j = 1, \dots, |\Box_k|, \tag{10}$$

that is

$$|v_k^{n,j}| \leqslant \frac{1}{|\Box_k|} \quad k = 1, \dots, n_c,\ j = 1, \dots, |\Box_k|. \tag{11}$$

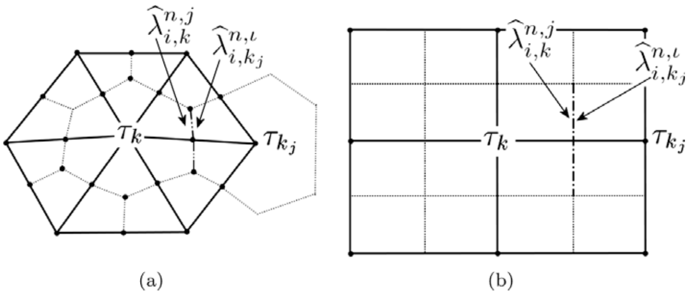
$\widehat{\lambda}^{n,j}_{i,k}$
$\widehat{\lambda}^{n,\iota}_{i,k_j}$
τ_k
τ_{k_j}
$\widehat{\lambda}^{n,j}_{i,k}$
$\widehat{\lambda}^{n,\iota}_{i,k_j}$
τ_k
τ_{k_j}
(a)
(b)

Fig. 3. An example of the boundary compositions $\widehat{\lambda}^{n,j}_{i,k}$ and $\widehat{\lambda}^{n,\iota}_{i,k_j}$ on an unstructured (a), and structured (b) grid. The first one is an approximation of the composition from inside τ_k while the other one is an approximation from the neighboring cell τ_{k_j}.

To define $\widehat{\lambda}^{n,j}_{i,k}$ we have used the following relation

$$\widehat{\lambda}^{n,j}_{i,k} = \lambda^n_{i,k} + \delta\lambda^{n,j}_{i,k}, \tag{12}$$

where $\delta\lambda^{n,j}_{i,k}$ is determined by defining the average of the set on the interfaces $\phi^{n,j}_{i,k} = \frac{1}{|l^j_k|}\int_{l^j_k}\phi_{i,\Delta}(t^n,\vec{X})$ and solving the following constrained minimization problem

$$\begin{cases} \min\limits_{\delta\lambda^{n,j}_{i,k}} \frac{1}{2}\sum\limits_{i=1}^{n_s}(\lambda^n_{i,k} - \phi^{n,j}_{i,k} + \delta\lambda^{n,j}_{i,k})^2, \\ \sum\limits_{i=1}^{n_s}\delta\lambda^{n,j}_{i,k} = 0, \\ \delta\lambda^{n,j}_{i,k,min} \leqslant \delta\lambda^{n,j}_{i,k} \leqslant \delta\lambda^{n,j}_{i,k,max}; \end{cases} \tag{13}$$

where

$$\begin{cases} \delta\lambda^{n,j}_{i,k,min} = -\lambda^n_{i,k}, \\ \delta\lambda^{n,j}_{i,k,max} = \min\left(\frac{(1+D^n_{\Delta,k}) - v^{n,j}_k|\mathbb{J}_k|}{v^{n,j}_k|\mathbb{J}_k|}\lambda^n_{i,k}, 1-\lambda^n_{i,k}\right). \end{cases} \tag{14}$$

In other words, the interface states $\widehat{\lambda}^{n,j}_{i,k}$ are best fit approximations of the local level set functions. Problem (13) has to be solved for all $k = 1,\ldots,n_c$ and for all $j \in \mathbb{J}_k$, where $\mathbb{J}_k = \{j \in [1, |\mathbb{I}^C_j|] : v^{n,j}_k \geqslant 0\}$ is the set of the indices of the outflow faces l^j_k of the kth cell. And therefore the upwind function (8) selects only the upwind state, we can avoid to compute the downwind flux thus reducing the computational burden.

3. Analysis of the method

Proposition 4. *If* $0 \leqslant \lambda^n_{i,\Delta} \leqslant 1$ *and if* $v^{n,j}_k \geqslant 0$ *(i.e. we are considering an outflow face), then problem* (13) *has a unique solution.*

Proof. We point out that the bound $0 \leqslant \lambda^n_{i,\Delta} \leqslant 1$will be proved shortly and we will consider only an outflow face since (13) is solved only for the faces characterized by $v^{n,j}_k \geqslant 0$.

First of all we will show that the feasible region for the $\delta\lambda^{n,j}_{i,k}$, defined by the constraints in problem (13), is a convex, nonempty subset of $\mathbb{R}^{n_s}$. Since the first of (13) is a convex minimization problem, we can conclude (see [20]) that the problem has a unique solution.

As we have stated previously, the feasible region is nonempty, in fact, we can bound all the terms in (14):

$$\delta\lambda^{n,j}_{i,k,min} = -\lambda^n_{i,k} \leqslant 0, \tag{15}$$

and

$$1 - \lambda^n_{i,k} \geqslant 0. \tag{16}$$

Then using (11)

$$D^n_{\Delta,k} = \sum_{j=1}^{|\mathbb{I}_k|} v^{n,j}_k \geqslant \sum_{j\notin\mathbb{J}_k} v^{n,j}_k \geqslant -\sum_{j\notin\mathbb{J}_k}\frac{1}{|\mathbb{I}_k|} = -\frac{|\mathbb{I}_k| - |\mathbb{J}_k|}{|\mathbb{I}_k|}.$$

Finally, we bound the second term in the second equation of (14):

$$\frac{(1+D^n_{\Delta,k})-v^{n,j}_k|\mathbb{J}_k|}{v^{n,j}_k|\mathbb{J}_k|}\lambda^n_{i,k} \geqslant \frac{1-\frac{|\mathbb{I}_k|-|\mathbb{J}_k|}{|\mathbb{I}_k|}-v^{n,j}_k|\mathbb{J}_k|}{v^{n,j}_k|\mathbb{J}_k|}\lambda^n_{i,k} = \frac{|\mathbb{J}_k|-v^{n,j}_k|\mathbb{J}_k||\mathbb{I}_k|}{v^{n,j}_k|\mathbb{J}_k||\mathbb{I}_k|}\lambda^n_{i,k} = \frac{1-v^{n,j}_k|\mathbb{I}_k|}{v^{n,j}_k|\mathbb{I}_k|}\lambda^n_{i,k}.$$

Then, from (11), we obtain that

$$\frac{(1+D^n_{\Delta,k})-v^{n,j}_k|\mathbb{J}_k|}{v^{n,j}_k|\mathbb{J}_k|}\lambda^n_{i,k} \geqslant 0 \tag{17}$$

and consequently using (16) and (17) we have $\delta\lambda^{n,j}_{i,k,max} \geqslant 0$. Therefore, considering (15), we get that the feasible set is not empty.

Let us now prove the convexity of the feasible region. Consider two vectors δ_i, ζ_i with $i = 1,\ldots,n_s$ belonging to the feasible region. Let $\alpha\delta_i + (1-\alpha)\zeta_i$, with $\alpha \in [0,1]$, be a convex combination of the vectors. We have to show that the linear combination belongs to the feasible region. We have

$$\begin{cases} \sum_{i=1}^{n_s}(\alpha\delta_i+(1-\alpha)\zeta_i) = \alpha\sum_{i=1}^{n_s}\delta_i + (1-\alpha)\sum_{i=1}^{n_s}\zeta_i = 0, \\ (\alpha\delta_i+(1-\alpha)\zeta_i) \leqslant \alpha\delta\lambda^{n,j}_{i,k,max} + (1-\alpha)\delta\lambda^{n,j}_{i,k,max} = \delta\lambda^{n,j}_{i,k,max}, \\ (\alpha\delta_i+(1-\alpha)\zeta_i) \geqslant \alpha\delta\lambda^{n,j}_{i,k,min} + (1-\alpha)\delta\lambda^{n,j}_{i,k,min} = \delta\lambda^{n,j}_{i,k,min}. \end{cases}$$

This completes the proof. □

Proposition 5. *The method defined by (12)–(14) is positive, i.e.* $\lambda^n_{i,k} \geqslant 0,\ \forall n,\ i = 1,\ldots,n_s,\ k = 1,\ldots,n_c$.

Proof. Let us proceed by induction. We suppose $\lambda^n_{i,k} \geqslant 0$, then using (7) we get

$$\lambda^{n+1}_{i,k} = \left(1+D^n_{\Delta,k}\right)\lambda^n_{i,k} - \sum_{j=1}^{|\mathbb{I}^C_k|}F^{n,j}_{i,k} \geqslant \left(1+D^n_{\Delta,k}\right)\lambda^n_{i,k} - \sum_{j\in\mathbb{J}_k}F^{n,j}_{i,k},$$

and combining it with Eqs. (8) and (12) we obtain

$$\lambda^{n+1}_{i,k} \geqslant \left(1+D^n_{\Delta,k}\right)\lambda^n_{i,k} - \sum_{j\in\mathbb{J}_k}v^{n,j}_k\left(\lambda^n_{i,k}+\delta\lambda^{n,j}_{i,k}\right) = \left(1+D^n_{\Delta,k}\right)\lambda^n_{i,k} - \lambda^n_{i,k}\sum_{j\in\mathbb{J}_k}v^{n,j}_k - \sum_{j\in\mathbb{J}_k}v^{n,j}_k\delta\lambda^{n,j}_{i,k}.$$

Finally, using (14), we have

$$\begin{aligned}\lambda^{n+1}_{i,k} \geqslant{}& \left(1+D^n_{\Delta,k}\right)\lambda^n_{i,k} - \lambda^n_{i,k}\sum_{j\in\mathbb{J}_k}v^{n,j}_k - \sum_{j\in\mathbb{J}_k}\frac{(1+D^n_{\Delta,k})-v^{n,j}_k|\mathbb{J}_k|}{|\mathbb{J}_k|}\lambda^n_{i,k} = \left(1+D^n_{\Delta,k}\right)\lambda^n_{i,k} - \lambda^n_{i,k}\sum_{j\in\mathbb{J}_k}v^{n,j}_k - (1+D^n_{\Delta,k})\lambda^n_{i,k} \\ &+ \lambda^n_{i,k}\sum_{j\in\mathbb{J}_k}v^{n,j}_k = 0.\end{aligned}$$

And, since $\lambda^0_{i,\Delta} \geqslant 0$, the proof follows. □

Proposition 6. *The sum of partial volumes, on every cell at every time step, equals one*

$$\sum_{i=1}^{n_s}\lambda^n_{i,k} = 1\ \forall n, \quad k = 1,\ldots,n_c. \tag{18}$$

Analogously the sum of the level set functions is everywhere equal to one

$$\sum_{i=1}^{n_s}\phi^n_{i,\Delta} = 1 \quad \forall n.$$

Proof. Let us use the induction principle. At $n = 0$, condition (18) is satisfied and we assume that the condition is satisfied at time t^n. We have

$$\sum_{i=1}^{n_s}\lambda^{n+1}_{i,k} = \sum_{i=1}^{n_s}\left[\left(1+D^n_{\Delta,k}\right)\lambda^n_{i,k} - \sum_{j=1}^{|\mathbb{I}^C_k|}F^{n,j}_{i,k}\right] = \left(1+D^n_{\Delta,k}\right) - \sum_{i=1}^{n_s}\sum_{j=1}^{|\mathbb{I}^C_k|}v^{n,j}_k\Phi(\widehat{\lambda}^{n,j}_{i,k},\widehat{\lambda}^{n,\imath}_{i,k_j}). \tag{19}$$

From the second equation of (7) and from the inductive hypothesis we have

$$\sum_{i=1}^{n_s} \widehat{\lambda}_{i,k}^{n,j} = \sum_{i=1}^{n_s} \lambda_{i,k}^{n} + \delta\lambda_{i,k}^{n} = 1,$$

and therefore

$$\sum_{i=1}^{n_s} \Phi(\widehat{\lambda}_{i,k}^{n,j}, \widehat{\lambda}_{i,k_j}^{n,\imath}) = 1. \tag{20}$$

Then, plugging (20) into (19), we get

$$\sum_{i=1}^{n_s} \lambda_{i,k}^{n+1} = \left(1 + D_{\Delta,k}^{n}\right) - \sum_{j=1}^{|\mathbb{I}_k^C|} v_k^{n,j}.$$

Recalling that $D_{\Delta,k}^{n} = \sum_{j=1}^{|\mathbb{I}_k^C|} v_k^{n,j}$, we obtain the first part of the thesis.

Finally, since $\phi_{i,\Delta}^{n} = \sum_{k=1}^{n_c} \lambda_{i,k}^{n} \vartheta_k^1$, we get

$$\sum_{i=1}^{n_s} \phi_{i,\Delta}^{n} = \sum_{i=1}^{n_s} \sum_{k=1}^{n_c} \lambda_{i,k}^{n} \vartheta_k^1 = \sum_{k=1}^{n_c} \vartheta_k^1 = 1. \qquad \square$$

We can also show a consistency result of the interface fluxes. Let $\psi : \lambda_{i,k}^{n}, \lambda_{i,k_0}^{n}, \ldots, \lambda_{i,k_{|\mathbb{I}_k|}}^{n} \rightarrow \widehat{\lambda}_{i,k}^{n,j}$be the map from the composition of the kth cell $\lambda_{i,k}^{n}$ and from the compositions of its neighboring cells λ_{i,k_j}^{n}, $j = 1, \ldots, |\mathbb{I}_k|$, to the jth interface composition of the kth cell i.e.

$$\widehat{\lambda}_{i,k}^{n,j} = \psi\left(\lambda_{i,k}^{n}, \lambda_{i,k_1}^{n}, \ldots, \lambda_{i,k_{|\mathbb{I}_k|}}^{n}\right).$$

Then we can show that

Proposition 7. *If* $\lambda_{i,k_j}^{n} = \lambda_{i,k}^{n}, \quad j = 1, \ldots, |\mathbb{I}_k|$, *then*

$$\lambda_{i,k}^{n} = \psi(\lambda_{i,k}^{n}, \lambda_{i,k_1}^{n}, \ldots, \lambda_{i,k_{|\mathbb{I}_k|}}^{n}).$$

Proof. In this particular case we have $\phi_{i,k}^{n,j} = \lambda_{i,k}^{n}, \quad j = 1, \ldots, |\mathbb{I}_k|$ and furthermore the optimal solution of (7) is $\delta\lambda_{i,k}^{n,j} = 0$. Actually this solution minimizes the objective function and clearly satisfies the equality constraint: in fact, from Proposition 4, we also know that $\delta\lambda_{i,k}^{n,j} = 0$ is always in the feasible set. □

Then, from (8) and (9), we get that the numerical flux is consistent too.

Finally, we can also prove the following statement:

Proposition 8. *Every discrete subdomain* $\Omega_{i,\Delta}$, *i = 1,...,n*$_s$, *does not overlap with the others, i.e.*

$$\Omega_{i,\Delta}(t) \cap \Omega_{j,\Delta}(t) = \emptyset, \quad i = 1, \ldots, n_s, \quad j = 1, \ldots, n_s, \ j \neq i, \ \forall t > 0$$

and, given a subregion $\widetilde{\Omega}$ *containing only two species identified by the indices* i_1, i_2, *we have*

$$\overline{\widetilde{\Omega}}_{i_1,\Delta} \cup \overline{\widetilde{\Omega}}_{i_2,\Delta} = \overline{\widetilde{\Omega}},$$

where $\overline{\widetilde{\Omega}}_{i_1,\Delta} = \overline{\Omega}_{i_1,\Delta} \cap \widetilde{\Omega}$ *and* $\overline{\widetilde{\Omega}}_{i_2,\Delta} = \overline{\Omega}_{i_2,\Delta} \cap \widetilde{\Omega}$.

Proof. If $\vec{X} \in \Omega_{i,\Delta}(t)$, then $\phi_{i,\Delta}(t,\vec{X}) > \frac{1}{2}$, therefore from proposition (6) we get

$$\sum_{j=1,\ j \neq i}^{n_s} \phi_{i,\Delta}(t,\vec{X}) < \frac{1}{2}$$

and, since the level set functions are the piecewise linear interpolation of a positive function (i.e. the volume fractions $\lambda_{i,\Delta}$), we get $\phi_{j,\Delta}(t,\vec{X}) < \frac{1}{2}\ \forall j \neq i$ namely $\vec{X} \notin \Omega_{j,\Delta}(t)\ \forall j \neq i$. In the special case in which in a subregion $\widetilde{\Omega}$ there are only two species we get, from the general case, that there are no overlaps between $\widetilde{\Omega}_{i_1,\Delta}$ and $\widetilde{\Omega}_{i_2,\Delta}$. We have only to prove that

$$\vec{X} \in \widetilde{\Omega}_{i_1,\Delta} \text{ or } \vec{X} \in \widetilde{\Omega}_{i_2,\Delta}, \text{ or } \vec{X} \in \overline{\widetilde{\Omega}}_{i_1,\Delta} \cap \overline{\widetilde{\Omega}}_{i_2,\Delta} \quad \forall \vec{X} \in \widetilde{\Omega}.$$

We consider the three cases,

1. $\phi_{i_1,\Delta}(t,\vec{X}) > \frac{1}{2}$, then $\vec{X} \in \widetilde{\Omega}_{i_1,\Delta}$;
2. $\phi_{i_1,\Delta}(t,\vec{X}) < \frac{1}{2}$, so $\vec{X} \in \widetilde{\Omega}_{i_2,\Delta}$, in fact: $\phi_{i_2,\Delta}(t,\vec{X}) = 1 - \phi_{i_1,\Delta}(t,\vec{X})$;
3. $\phi_{i_1,\Delta}(t,\vec{X}) = \frac{1}{2}$, consequently the point $\vec{X} \in \overline{\widetilde{\Omega}}_{i_1,\Delta} \cap \overline{\widetilde{\Omega}}_{i_2,\Delta}$.

Therefore, we obtain the thesis. □

If more than two species are present in a subregion $\widetilde{\Omega}$, then some void space could be generated. This can happen only nearby the triple points, i.e., the points where more than two fluids meet. However, as we will see in Section 7, these void spaces are confined to a cell size.

4. Convergence analysis of the one-dimensional case

In one-dimension a more detailed analysis is possible. Let $\mathscr{T}_\Delta$ be a uniformly Δx-spaced 1D mesh (see Fig. 4) with elements $e_0, \dots, e_{r-1}, e_r, e_{r+1}, \dots, e_{n_e}$ and let us consider its dual mesh endowed with an ordered sequence of cells $\tau_0, \dots, \tau_{k-1}, \tau_k, \tau_{k+1}, \dots, \tau_{n_c}$. For the sake of simplicity let u be a constant, positive, velocity field (i.e., we are treating a null divergence case) and let $v^n = \frac{\Delta t^n}{\Delta x} u$ be the Courant number. Notice that, in this case, all the Courant numbers are equivalent to v^n. Besides, every cell is associated to a mean composition $\lambda^n_{i,0}, \dots, \lambda^n_{i,k-1}, \lambda^n_{i,k}, \lambda^n_{i,k+1}, \dots, \lambda^n_{i,n_c}$ and has two boundary sub-cell compositions $\widehat{\lambda}^{n,j}_{i,k}$ with $j = 1, 2$. Using a more explicit notation, for the 1D case, we can define the upwind sub-cell compositions as $\widehat{\lambda}^{n,+}_{i,k}$ and the downwind sub-cell compositions as $\widehat{\lambda}^{n,-}_{i,k}$. Method (7) takes the form

$$\begin{cases} \lambda^{n+1}_{i,k} = \lambda^n_{i,k} - v(\widehat{\lambda}^{n,+}_{i,k} - \widehat{\lambda}^{n,+}_{i,k-1}) \\ \widehat{\lambda}^{n,+}_{i,k} = \lambda^n_{i,k} + \delta\lambda^{n,+}_{i,k}. \end{cases} \tag{21}$$

While the minimization problem (13) becomes

$$\begin{cases} \min\limits_{\delta\lambda^{n,+}_{i,k}} \frac{1}{2} \sum\limits_{i=1}^{n_s} \left(\frac{1}{2} \left(\lambda^n_{i,k} - \lambda^n_{i,k+1} \right) + \delta\lambda^{n,+}_{i,k} \right)^2, \\ \sum\limits_{i=1}^{n_s} \delta\lambda^{n,+}_{i,k} = 0, \\ \delta\lambda^{n,+}_{i,k,min} \leqslant \delta\lambda^{n,+}_{i,k} \leqslant \delta\lambda^{n,+}_{i,k,max}, \end{cases} \tag{22}$$

where

$$\begin{cases} \delta\lambda^{n,+}_{i,k,min} = -\lambda^n_{i,k}, \\ \delta\lambda^{n,+}_{i,k,max} = \min\left(\frac{(1+D^n_{\Delta,k}) - v^n|\mathbb{J}_k|}{v^n|\mathbb{J}_k|} \lambda^n_{i,k}, 1 - \lambda^n_{i,k} \right). \end{cases} \tag{23}$$

In the one-dimensional case it is possible, using the modified equation technique, to carry out a convergence analysis, see [14]. Let $U_i(t, \vec{X})$ be the modified solution such that:

$$\lambda^n_{i,k} = U_i(t^n, \vec{x}_k) \ \forall n, \ i = 1, \dots, n_s, \ k = 1, \dots, n_c.$$

Moreover, we suppose that $U_i \in \mathbb{C}^2(\mathbb{R}^+ \times \Omega)$, then

$$\begin{cases} \lambda^n_{i,k+1} = U_i(t^n, x_k) + \frac{\partial}{\partial x} U_i(t^n, x_k)\Delta x + O(\Delta x^2), \\ \lambda^n_{i,k-1} = U_i(t^n, x_k) - \frac{\partial}{\partial x} U_i(t^n, x_k)\Delta x + O(\Delta x^2), \\ \lambda^{n+1}_{i,k} = U_i(t^n, x_k) + \frac{\partial}{\partial t} U_i(t^n, x_k)\Delta t + O(\Delta t^2). \end{cases} \tag{24}$$

For simplicity we also denote:

$$U^n_{i,k} = U_i(t^n, \vec{x}_k), \quad \frac{\partial U^n_{i,k}}{\partial x} = \frac{\partial}{\partial x} U_i(t^n, \vec{x}_k), \frac{\partial^2 U^n_{i,k}}{\partial x^2} = \frac{\partial^2}{\partial x^2} U_i(t^n, \vec{x}_k), \quad \frac{\partial U^n_{i,k}}{\partial t} = \frac{\partial}{\partial t} U_i(t^n, \vec{x}_k).$$

Let us now attend to the convergence analysis:

Proposition 9. *The method defined by* (22) *is second order accurate in space and first order accurate in time.*

Proof. We first show that

$$\delta\lambda^{n,+}_{i,k} = \frac{1}{2} \frac{\partial U^n_{i,k}}{\partial x} \Delta x \tag{25}$$

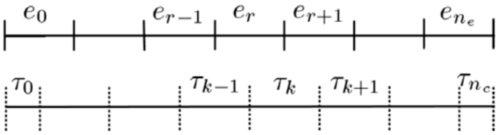

Fig. 4. The one-dimensional mesh. In the first part are depicted the mesh and the elements e_r, while in the second part the dual mesh is shown along with its cells τ_k.

is the optimal solution of problem (22) for $\Delta x \rightarrow 0$. Substituting (24) in the first of (22) we get

$$\frac{1}{2}\sum_{i=1}^{n_s}\left(\frac{1}{2}U^n_{i,k} - \frac{1}{2}U^n_{i,k} - \frac{1}{2}\frac{\partial U^n_{i,k}}{\partial x}\Delta x + O(\Delta x^2) + \frac{1}{2}\frac{\partial U^n_{i,k}}{\partial x}\Delta x\right) = n_s O(\Delta x^2).$$

Therefore, the functional is minimized for $\Delta x \rightarrow 0$. It is also possible to prove that (25) satisfies the constraints of (22), in fact, plugging (25) in the second equation of (22) we get

$$\sum_{i=1}^{n_s}\delta\lambda^{n,+}_{i,k} = \sum_{i=1}^{n_s}\frac{1}{2}\frac{\partial U^n_{i,k}}{\partial x}\Delta x. \tag{26}$$

Then, summing the components of the Taylor expansion $U^n_{i,k+1} = U^n_{i,k} + \frac{\partial}{\partial x}U^n_{i,k}\Delta x + O(\Delta x^2)$ and thanks to (18), we get

$$\sum_{i=1}^{n_s}\frac{\partial U^n_{i,k}}{\partial x}\Delta x = O(\Delta x^2). \tag{27}$$

Plugging (27) into (26) we obtain that (25) satisfies the first constraint of (22). Then, substituting the Taylor expansions (24) into (23), we get

$$\begin{cases}\delta\lambda^{n,+}_{i,k,min} = -U^n_{i,k},\\ \delta\lambda^{n,+}_{i,k,max} = \min\left(\frac{(1+D^n_{\Delta,k})-\nu^n|\mathbb{J}_k|}{\nu^n|\mathbb{J}_k|}U^n_{i,k}, 1 - U^n_{i,k}\right).\end{cases}$$

If Δx is small enough, the conditions, above mentioned for $\delta\lambda^{n,+}_{i,k}$, are equivalent to

$$-\frac{U^n_{i,k}}{\Delta x} \leqslant \frac{1}{2}\frac{\partial U^n_{i,k}}{\partial x} \leqslant \min\left(\frac{(1+D^n_{\Delta,k})-\nu^n|\mathbb{J}_k|}{\nu^n|\mathbb{J}_k|}\frac{U^n_{i,k}}{\Delta x}, \frac{1-U^n_{i,k}}{\Delta x}\right). \tag{28}$$

And for $\Delta x \rightarrow 0$ Eq. (28) becomes $-\infty \leqslant \frac{\partial U^n_{i,k}}{\partial x} \leqslant \infty$ (which is clearly satisfied) except if $U^n_{i,k} = 0$ or $U^n_{i,k} = 1$. In that cases we have

$$\frac{\partial U^n_{i,k}}{\partial x} \geqslant 0, \quad \frac{\partial U^n_{i,k}}{\partial x} \leqslant 1, \tag{29}$$

respectively. Now we prove that (29) are satisfied. In fact, if $U^n_{i,k} = 0$, using the Taylor series, we have

$$U^n_{i,k+1} = \frac{\partial U^n_{i,k}}{\partial x}\Delta x + O(\Delta x^2).$$

Then, since $U^n_{i,k+1} = \lambda^n_{i,k+1} \geqslant 0$ and dividing by Δx, we get $\frac{\partial U^n_{i,k}}{\partial x} + O(\Delta x) \geqslant 0$. Finally, from $\Delta x \rightarrow 0$ we get the first of (28). Using a similar technique we can also prove the second of (28).

We can now estimate the convergence order of our method. Substituting (24) and (25) in the second of (21) we obtain

$$\begin{cases}\widehat{\lambda}^{n,+}_{i,k} = U^n_{i,k} + \frac{1}{2}\frac{\partial U^n_{i,k}}{\partial x}\Delta x,\\ \widehat{\lambda}^{n,+}_{i,k-1} = U^n_{i,k-1} + \frac{1}{2}\frac{\partial}{\partial x}U^n_{i,k-1}\Delta x\end{cases}$$

and combining them with the first of (21) we get

$$\Delta t^n\frac{\partial U^n_{i,k}}{\partial t} + O((\Delta t^n)^2) = -\frac{\Delta t^n}{\Delta x}u\left(U^n_{i,k} - U^n_{i,k-1} + \frac{1}{2}\frac{\partial}{\partial x}U^n_{i,k}\Delta x - \frac{1}{2}\frac{\partial}{\partial x}U^n_{i,k-1}\Delta x\right). \tag{30}$$

Using Taylor expansions we obtain

$$\begin{cases}U^n_{i,k} - U^n_{i,k-1} = +\frac{\partial U^n_{i,k}}{\partial x}\Delta x - \frac{1}{2}\frac{\partial^2 U^n_{i,k}}{\partial x}\Delta x^2 + O(\Delta x^3),\\ \frac{1}{2}\left(\frac{\partial U^n_{i,k}}{\partial x} - \frac{\partial U^n_{i,k-1}}{\partial x}\right)\Delta x = \frac{1}{2}\frac{\partial U^n_{i,k}}{\partial x^2}\Delta x^2 + O(\Delta x^3).\end{cases} \tag{31}$$

Plugging (31) into (30) and dividing by Δt^n, we obtain

$$\frac{\partial U^n_{i,k}}{\partial t} + u\frac{\partial U^n_{i,k}}{\partial x} = O(\Delta t^n) + O(\Delta x^2) \quad \forall n,\ i = 1,\ldots,n_s,\ k = 1,\ldots,n_c$$

and the proof follows. □

5. Numerical solution of the minimization problem

In this section, we study an efficient implementation of a numerical solver for problem (13) and we also give an estimate of its computational cost. In particular, we use a gradient method slightly adapted in order to solve our particular problem.

For every outflow boundary of every cell we pass to the solver the following inputs: λ_i, ϕ_i, D_Δ, v and $|\mathbb{J}|$. Here we drop the indices k and j since we illustrate the algorithm applied to a generical cell and interface. We use the following scheme:

Algorithm 1. We compute the following quantities

$$\begin{cases} \delta\lambda_{i,min} = -\lambda_i, \\ \delta\lambda_{i,max} = \min\left(\frac{(1+D_\Delta)-v|\mathbb{J}|}{v|\mathbb{J}|}\lambda_i, 1-\lambda_i\right) \end{cases} \tag{32}$$

and we set $\delta\lambda_i^{(0)} = 0,\ i = 1,\ldots,n_s$. Then, for m = 0, 1, ..., we compute

$$\begin{cases} N_i^{(m)} = \begin{cases} 0 & \text{if } \delta\lambda_i^{(m)} = \delta\lambda_{i,min} \text{ and } (\lambda_i - \phi_i + \delta\lambda_i^{(m)}) > 0, \\ 0 & \text{if } \delta\lambda_i^{(m)} = \delta\lambda_{i,max} \text{ and } (\lambda_i - \phi_i + \delta\lambda_i^{(m)}) < 0, \\ 1 & \text{otherwise}, \end{cases} \\ \widetilde{G}_i^{(m)} = (\phi_i - \lambda_i - \delta\lambda_i^{(m)})N_i^{(m)}, \\ G_i^{(m)} = \widetilde{G}_i^{(m)} - \frac{\sum_{j=1}^{n_s}\widetilde{G}_j^{(m)}}{\sum_{j=1}^{n_s}N_j^{(m)}}N_i^{(m)}, \\ \alpha = \begin{cases} \min\left(1, \frac{\delta\lambda_{1,max}-\delta\lambda_1^{(m)}}{G_1^{(m)}}, \ldots, \frac{\delta\lambda_{n_s,max}-\delta\lambda_{n_s}^{(m)}}{G_{n_s}^{(m)}}\right) \text{ if } G_i^{(m)} \geqslant 0, \\ \min\left(1, \frac{\delta\lambda_{1,min}-\delta\lambda_1(m)}{G_1^{(m)}}, \ldots, \frac{\delta\lambda_{n_s,min}-\delta\lambda_{n_s}^{(m)}}{G_{n_s}^{(m)}}\right) \text{ if } G_i^{(m)} < 0, \end{cases} \\ \delta\lambda_i^{(m+1)} = \alpha G_i^{(m)} + \delta\lambda_i^{(m)}. \end{cases} \tag{33}$$

If $G_i^{(m)} = 0,\ i = 1,\ldots,n_s$, we stop the iterations.

The algorithm is, in fact, a projected gradient method applied to the given cost function. $\widetilde{G}_i^{(m)}$ is a one-sided projection of the gradient on the boundary of the box [$\lambda_{i,min}$, $\lambda_{i,max}$] while $G_i^{(m)}$ accounts for the constraint $\sum_{i=1}^{n_s}\delta\lambda_i^{(m+1)} = 0$ by imposing that

$$\sum_{i=1}^{n_s} G_i^{(m)} = 0. \tag{34}$$

Finally, the coefficient α ensures that $\lambda_i^{(m+1)}$ belongs to the feasible set, i.e.

$$\delta\lambda_{i,min} \leqslant \alpha G_i^{(m)} + \delta\lambda_i^{(m)} \leqslant \delta\lambda_{i,max}. \tag{35}$$

We now define $F_S^{(m)} = \{i \in \mathbb{R}^{n_s} : N_i^{(m)} = 1\}$ as the set of the active components and, similarly, we define $NF_S^{(m)} = \{i \in \mathbb{R}^{n_s} : N_i^{(m)} = 0\}$ as the set of the constrained components.

We can show that this algorithm has some interesting properties:

Proposition 10. *If $\delta\lambda_{i,min} \leqslant \phi_i - \lambda_i \leqslant \delta\ \lambda_{i,max}$, then Algorithm 1 terminates in two steps.*

Proof. Since $\delta\lambda_i^{(0)} = 0,\ i = 1,\ldots,n_s$, from (35) we have

$$\begin{cases} \text{if } \delta\lambda_{i,min} = 0 \text{ then } \phi_i - \lambda_i \geqslant 0, \\ \text{if } \delta\lambda_{i,max} = 0 \text{ then } \phi_i - \lambda_i \leqslant 0 \end{cases}$$

and this implies that $N_i^{(0)} = 1,\ i = 1,\ldots,n_s$. Consequently

$$\sum_{i=1}^{n_s}\widetilde{G}_i^{(0)} = \sum_{i=1}^{n_s}(\phi_i - \lambda_i) = 0,$$

and

$$G_i^{(0)} = \widetilde{G}_i^{(0)} = \phi_i - \lambda_i.$$

Under this condition α = 1 and then

$$\delta\lambda_i^{(1)} = \alpha G_i^{(0)} + \delta\lambda_i^{(0)} = \phi_i - \lambda_i.$$

Then, at iteration m = 1, we have $\widetilde{G}_i^{(1)} = G_i^{(1)} = 0,\ i = 1,\ldots,n_s$ and, hence, we stop. □

We can also prove a more general stopping estimation:

Proposition 11. *Algorithm 1 terminates in less than $n_s - |NF_S^{(0)}| + 1$ steps.*

Proof. First we show that if$i \in NF_S^{(m)}$, then $i \in NF_S^{(m+1)}$. Indeed, from (33) we have that for all $i \in NF_S^{(m)}$

$$\delta\lambda_i^{(m+1)} = \delta\lambda_i^{(m)}.$$

Therefore, every step of Algorithm 1 can possibly add a further component to $NF_S^{(m)}$. A maximum of $n_s - |NF_S^{(0)}|$ constraints can be added. If all the components are constrained indeed, for an $m \leqslant n_s - |NF_S^{(0)}| + 1$, we have $\widetilde{G}_i^{(m)} = G_i^{(m)} = 0$ with $i = 1,\ldots,n_s$ and the iteration stops.

Conversely, if no constraint is added, the iteration stops. Indeed, if at $m + 1$ no constraint is added, then at step m $\alpha = 1$ and we have

$$\delta\lambda_i^{(m+1)} = G_i^{(m)} + \delta\lambda_i^{(m)} \text{ and } N_i^{(m+1)} = N_i^{(m)}, \quad i = 1,\ldots,n_s.$$

Moreover,

$$\widetilde{G}_i^{(m+1)} = (\phi_i - \lambda_i - G_i^{(m)} - \delta\lambda_i^{(m)})N_i^{(m)},$$

and, since from (33) $(\phi_i - \lambda_i - \delta\lambda_i^{(m)})N_i^{(m)} = \widetilde{G}_i^{(m)}$, we get

$$\widetilde{G}_i^{(m+1)} = -G_i^{(m)}N_i + \widetilde{G}_i^{(m)} = -G_i^{(m)} + \widetilde{G}_i^{(m)}. \tag{36}$$

Finally, using (36) in the definition of $G_i^{(m+1)}$ we get

$$G_i^{(m+1)} = \widetilde{G}_i^{(m)} - \frac{\sum_{j=1}^{n_s}\widetilde{G}_j^{(m)}}{\sum_{j=1}^{n_s}N_j^{(m)}}N_i^{(m)} - G_i^{(m)} + \frac{\sum_{j=1}^{n_s}G_j^{(m)}}{\sum_{j=1}^{n_s}N_j^{(m)}}N_i^{(m)}. \tag{37}$$

Since $G_j^{(m)}$ has a zero mean value, last term in (37) is zero. Moreover, since

$$\widetilde{G}_i^{(m)} - \frac{\sum_{j=1}^{n_s}\widetilde{G}_j^{(m)}}{\sum_{j=1}^{n_s}N_j^{(m)}}N_i^{(m)} = G_i^{(m)},$$

we have that

$$G_i^{(m+1)} = 0, \quad i = 1,\ldots,n_s \tag{38}$$

and the Algorithm terminates. □

Proposition 12. Algorithm 1 *finds the optimal solution for* (13)

Proof. We have already noted that the algorithm produces $\delta\lambda_i^{(m)}$ which are always in the feasible region. Then, if we show that the Algorithm finds a local minimum, from the convexity and differentiability of the objective function and of the constraints, it follows that the minimum solution is also the global one.

Let us now show that, if $G_i^{(m)} = 0$, $i = 1,\ldots,n_s$, then there is no improving direction. The allowed improving direction is $\widetilde{G}_i^{(m)}$ but, since $G_i^{(m)} = 0$, $i = 1,\ldots,n_s$, we have that $\widetilde{G}_i^{(m)}$ is orthogonal to the equality constraint of (13). Therefore there are no improving directions allowed. □

6. Level set function reconstruction

We need also to build an algorithm to reconstruct the LS function. As we have dropped the usual definition of the distance function, we need to define a proper algorithm for the reconstruction of the LS:

Algorithm 2. If there is an index $\bar{i}$ such that:

$$\lambda_{\bar{i},k}^n > \frac{1}{2} \text{ and } \lambda_{\bar{i},k_j}^n > \frac{1}{2} \quad \forall k_j \in \mathbb{I}_k, \tag{39}$$

then we set $\lambda_{\bar{i},k}^n = 1$ and $\lambda_{i,k}^n = 0$ with $i = 1,\ldots,n_s$, $i \neq \bar{i}$. Otherwise, we maintain the nodal value $\lambda_{i,k}^n$. Then the level set function is updated using (6).

This algorithm does not modify the LS function in all the elements where $\phi_{i,\Delta}^n$ equals $\frac{1}{2}$; in other words, the interface position is not modified by this algorithm. Actually, if Eq. (39) is satisfied, from (6) we get $\tau_k \in \Omega_i$ and therefore we can set $\lambda_{\bar{i},k}^n = 1$. Since the evolution of the interfaces is independent of the set function (see [19,25]), this algorithm does not introduce any error from the LS point of view. The Algorithm 2 does not guarantee the mass conservation in the sense that $\sum_{k=1}^{n_c}\lambda_{i,k}^n$ is not conserved. However, as it is shown in [24] and in Section 7, the mass discrepancy introduced by the reconstruction algorithm is small.

Having concluded the definition of our method, we devote the next section to its analysis.

7. Numerical results

7.1. Results

In this section, we introduce some numerical experiments which aim to illustrate the quality of the numerical scheme proposed here. The first one is the convergence result in one-dimension. In this case, in order to accommodate the boundary conditions, we have made a slight modification to the algorithm. In the description of this method we have, so far, neglected the boundary conditions since we wanted to focus on the properties of the method which do not depend on them. We consider a test case with a constant transport speed, i.e., $v = 1$ and the domain Ω is the interval [0,1]. The initial conditions are $\lambda^0_{1,k} = 1,\ \lambda^0_{2,k} = 0\ \forall k$, while $\lambda^b_1(t) = 0,\ \lambda^b_2(t) = 1\ \forall t \in [0, T]$ are the boundary conditions on the left (inflow) side. The problem

$$\frac{\partial \lambda_i}{\partial t} + u\frac{\partial \lambda_i}{\partial x} = 0$$

has the following analytical solution

$$\lambda_1(t,x) = H(x-ut), \quad \lambda_2(t,x) = 1 - H(x-ut).$$

Furthermore, it is possible to compute the L^1 error on (0,T) × (0,1) defined as

$$E_{L^1} = \sum_{i=1}^{n_s} \int_0^T \int_a^b |\lambda_i(t,x) - \lambda_{i,\Delta}(t,x)|.$$

In Fig. 5, we show the L^1 error of the proposed method, compared with a high resolution Discontinuous Galerkin (DG) method with a MinMod Limiter (see [7]) and with the Godunov (G) method. Our method compares favorably with the DG method,

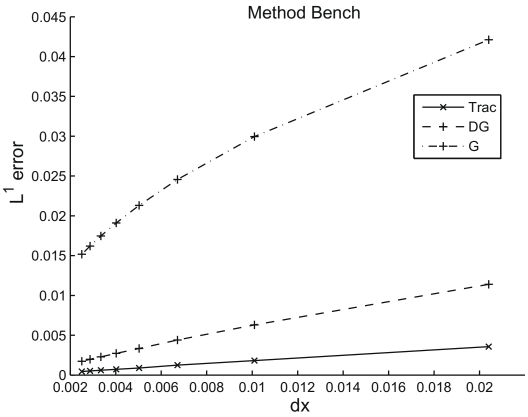

Fig. 5. One-dimensional convergence of the proposed method (Trac) compared with a Discontinuous Galerkin (DG) method and with the Godunov (G) one.

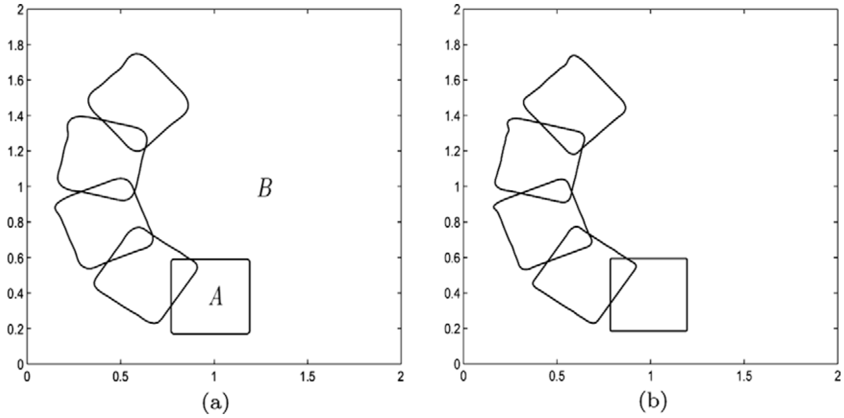

Fig. 6. Tracking of a square with (a) 10,000 degrees of freedom and $cfl = \frac{1}{10}$ and (b) 40,000 degrees of freedom.

though the regularity of the solution limits the convergence rate. In fact, in this case, both our method and the DG one are only first order accurate.

Let us now consider some classical examples in two-dimensions: in Fig. 6(a) we outline some results obtained with a rotational field $\vec{V}(\vec{X}) = [-X_2 - 1,\ X_1 - 1]$, where X_1, X_2 are the cartesian components of $\vec{X}$. A square is filled with a fluid tagged as *A*, the remaining space is filled with a fluid tagged as *B*; the square lower left corner coordinates are [0.8, 0.2] and the upper right corner coordinates are [1.2, 0.6]. If we compare with the comprehensive benchmark analysis performed in [9], we see that our results are intermediate, nevertheless our method has the possibility to track a large number of fluids and has good conservation properties. In Fig. 7(a) the mass ratio between the volume occupied by the subdomains *A* and *B* and the total volume is shown. As it can be seen, the volume is almost perfectly conserved. In Fig. 7(b) the same experiment is performed applying the LS reconstruction Algorithm 2 at time steps 600, 1200 and 1800. The volume changes of subdomain A are very small.

As we have already pointed out, our method has the possibility to track a large number of fluids, according to what we show in Fig. 8. Here we consider the foregoing case with the difference that we deal with three fluids: the first, tagged as *A*, fills the inner square, the outer is filled with fluid *B* and the remaining space in the domain is filled with fluid *C*, see Fig. 8(a). The coordinates of the lower left - upper right corners of the inner square are respectively [0.9, 0.3], [1.1, 0.5] while the outer square corners coordinates are [0.8, 0.2], [1.2, 0.6]. As one can see from Fig. 8, the tracking performances are independent of the number of the species being tracked.

In Fig. 9 we track three non-nested fluids, showing the coherence between the three tracked interfaces; the small rectangle filled with fluid *C* has the following corners coordinates: [0.75, 1], [1.25, 1.45]. The fluid *B* fills a more complex region, that is the complementary part of the rectangle *C* in a rectangle with corner coordinates [0.5, 0.45], [1.5, 1.45]. The remaining part of the computational domain is filled with the fluid *A*. In this test case, we show the performances of our method when some triple points (in this case two) are present. The void spaces are very small and the interfaces between the fluids are almost coherent.

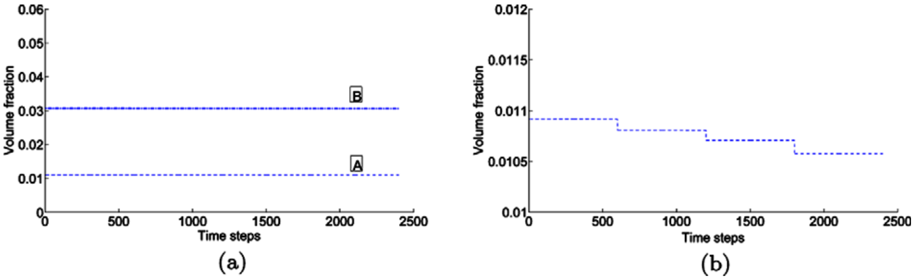

Fig. 7. (a) The conservation of the mass fraction of species *A* (dashed line) and *B* (dash-dot line) plotted against the time steps. (b) The conservation of the species *A* when a reconstruction algorithm is applied.

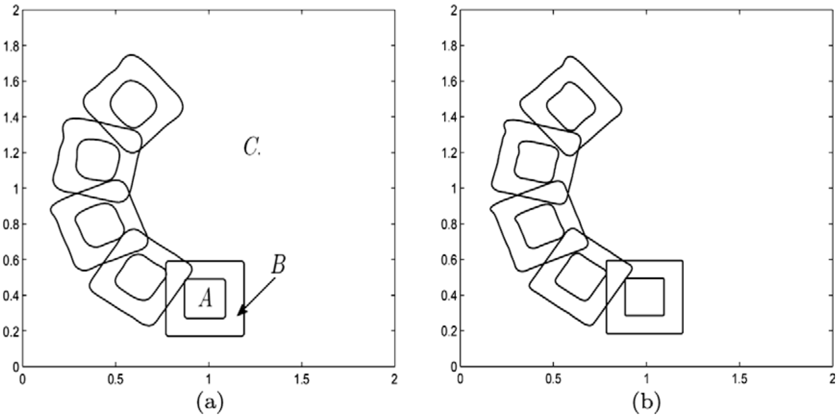

Fig. 8. Tracking of two nested squares with (a) 10,000 degrees of freedom and (b) 40,000 degrees of freedom. In this case three species are involved: the inner square is filled with the species *A*, the outer with species *B* and the rest of the domain with species *C*.

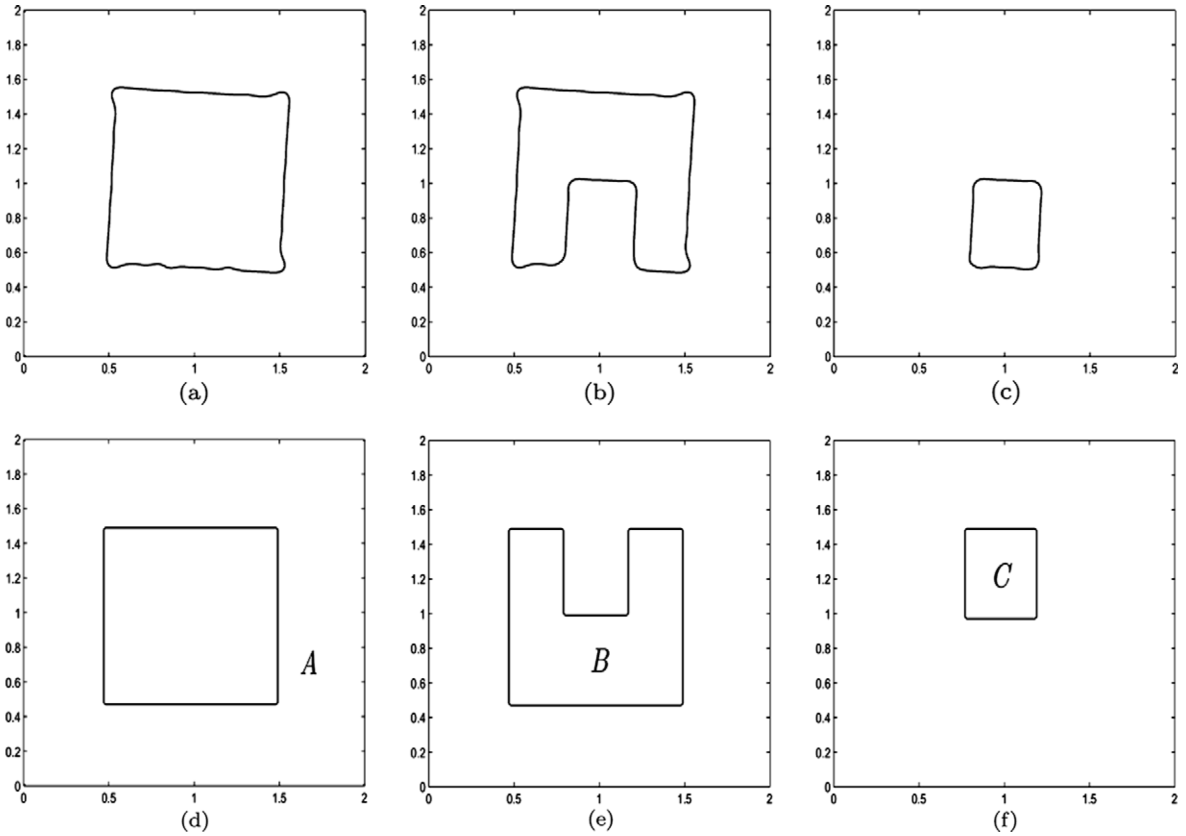

Fig. 9. Multi-fluid tracking using 10,000 degrees of freedom. (a)–(c) are the computed interfaces after half a turn and (d)–(f) are the initial configurations.

8. Conclusions

We have devised a coupled level set – volume tracking method that has been implemented in one, two and three-dimensions, see [24]. It is computationally efficient and able to perform on general unstructured grids. It is currently used in geophysical simulations, where the velocity field is computed by solving a Stokes problem [15] and has demonstrated its flexibility in treating complex simulations. Because of its local structure, it is easily parallelizable. In forthcoming works we will show more details about applications to the multi-fluid geological simulations and we will investigate some higher performance implementations of our code.

Acknowledgments

The authors wish to acknowledge the support of the Italian MIUR (Grant PRIN 2007). They also wish to thank G. Scrofani and P. Ruffo for having supported such interesting research project.

References

[1] A. Anderson, X. Zheng, V. Cristini, Adaptive unstructured volume remeshing – 1: The method, J. Comput. Phys. 208 (2005) 616–625.
[2] A. Anderson, J. Lowengrub, X. Zheng, V. Cristini, Adaptive unstructured volume remeshing – 2: Applications to two-and three-dimensional level-set simulations of multiphase flow, J. Comput. Phys. 208 (2005) 626–650.
[3] E. Aulisa, S. Manservisi, R. Scardovelli, A surface marker algorithm coupled to an area-preserving marker redistribution method for three-dimensional interface tracking, J. Comput. Phys. 197 (2004) 555–584.
[4] E. Aulisa, S. Manservisi, R. Scardovelli, A mixed markers and volume-of-fluid method for the reconstruction and advection of interfaces in two-phases and free-boundary flows, J. Comput. Phys. 188 (2003) 611–639.
[5] T. Brochu, R. Bridson, Fluid animation with explicit surface meshes, in: ACM SIGGRAPH Symposium, 2006.
[6] A. Caboussat, Numerical simulation of two-phase free surface flows, Arch. Comput. Methods Eng. 12 (2) (2005) 165–224.
[7] B. Cockburn, C.W. Shu, TVB Runge Kutta local projection discontinuous Galerkin finite element method for conservation laws 2: General framework, Math. Comput. 52 (1989) 411–435.
[8] R.M. Darlington, T.L. McAbee, G. Rodrigue, A study of ALE simulations of Rayleigh-Taylor instability, Comput. Phys. Commun. 135 (2001) 58–73.
[9] D.A. Di Pietro, S. Loforte, N. Parolini, Mass preserving finite element implementations of the level set method, Appl. Numer. Math. 56 (2006) 1179–1195.
[10] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A hybrid particle level set methods for improved interface capturing, J. Comput. Phys. 183 (2002) 83–116.

[11] S. Esedoglu, P. Smereka, A variational formulation for a level set representation of multiphase flow and area preserving curvature flow, Commun. Math. Sci. 6 (2008) 125–148.

[12] J. Glimm, J.W. Grove, X.L. LI, K. Shyue, Y. Zeng, Q. Zhang, Three-dimensional front tracking, SISC 19 (1998) 703–727.

[13] D.B. Kothe, Perspective on Eulerian finite volume methods for incompressible interfacial flows. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.27.4783&rep=rep1&type=pdf>, 1999.

[14] R.J. LeVeque, Numerical Methods for Conservation Laws, Birkhauser, 1999.

[15] M. Longoni, A.C.I. Malossi, A. Villa, A Robust and Efficient Technique for Simulating Three-dimensional Sedimentary Basins Dynamics, Report MOX 26, <http://mox.polimi.it>, 2009.

[16] P.D. Minev, T. Chen, K. Nandakumar, A finite element technique for multigrid incompressible flow using Eulerian grids, J. Comput. Phys. 187 (2003) 255–273.

[17] K. Ohmori, N. Saito, Flux-free finite element method with lagrange multipliers for two-fluid flows, SISC 32 (2004) 147–173.

[18] E. Onate, S.R. Idelsohn, F. Del Pin, R. Aubry, The particle finite element method: an overview, Int. J. Comput. Methods 1 (2) (2004) 267–307.

[19] S. Osher, R.P. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, 2003.

[20] P. Pedragal, Introduction to Optimization, Springer, 2004.

[21] S.P. van der Pijl, A. Segal, C. Vuik, P. Wesseling, A mass-conserving level-set method for modelling of multi-phase flows, Int. J. Numer. Methods Fluids 47 (2005) 339–361.

[22] S. Popinet, S. Zaleski, A front-tracking algorithm for accurate representation of surface tension, Int. J. Numer. Methods Fluids 30 (1999) 775–793.

[23] S. Popinet, An accurate adaptive solver for surface-tension-driven interfacial flows, J. Comput. Phys. 228 (2009) 5838–5866.

[24] A. Scotti, A. Villa. Predictive numerical models of basin evolution and petroleum generation, in: AIMI III, Proceedings of the 9th Conference SIMAI, 2008.

[25] A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, 1999.

[26] S. Shin, D. Juric, Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity, J. Comput. Phys. 180 (2002) 427–470.

[27] A. Smolianski, Finite-Element/level-Set/Operator-Splitting (FELSOS) approach for computing two-fluid unsteady flows with free moving interfaces, Int. J. Numer. Methods Fluids 24 (2004) 231–269.

[28] M. Sussman, E. Fatemi, P. Smereka, S. Osher, An improved level set method for incompressible two-phase flows, Comput. Fluids 27 (1998) 663–680.

[29] M. Sussman, E.G. Puckett, A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows, J. Comput. Phys. 162 (2000) 301–337.

[30] M. Sussman, K.M. Smith, M.Y. Hussaini, M. Ohta, R. Zhi-Wei, A sharp interface method for incompressible two-phase flow, J. Comput. Phys. 221 (2007) 469–505.

[31] D.J. Torres, J.U. Brackbill, The point-set method: front-tracking without connectivity, J. Comput. Phys. 165 (2000) 620–644.

[32] G. Tryggvason, B. Brunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jant, A front-tracking method for the computations of multiphase flow, J. Comput. Phys. 169 (2001) 708–759.

[33] H.-K. Zhao, T. Chan, B. Merriman, S. Osher, A variational level set approach to multiphase motion, J. Comput. Phys. 127 (1996) 179–195.

[34] S. Zlotnik, P. Diez, Hierarchical X-FEM for n-phase flow ($n > 2$), Comput. Methods Appl. Mech. Eng. 198 (2009) 2329–2338.