Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 188 (2003) 100–122
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Second order accurate volume tracking based on remapping for triangular meshes

Khosro Shahbazi, Marius Paraschivoiu *, Javad Mostaghimi

Department of Mechanical and Industrial Engineering, University of Toronto, 5 King's College Road, Toronto, Ont., Canada M5S 3G8

Received 30 April 2002; received in revised form 7 February 2003; accepted 5 March 2003

Abstract

This paper presents a second order accurate piecewise linear volume tracking based on remapping for triangular meshes. This approach avoids the complexity of extending unsplit second order volume of fluid algorithms, advection methods, on triangular meshes. The method is based on Lagrangian–Eulerian (LE) methods; therefore, it does not deal with edge fluxes and corner fluxes, flux corrections, as is typical in advection algorithms. The method is constructed of three parts: a Lagrangian phase, a reconstruction phase and a remapping phase. In the Lagrangian phase, the original, Eulerian, grid is projected along trajectories to obtain Lagrangian grids. In practice, this projection is handled through the time integration of velocity field for grid vertices at each time step. The reconstruction is based on truncating the volume material polygon for each Lagrangian mixed grid. Since in piecewise linear approximation, the interface is represented by a segment line, the polygon material truncation is mainly finding the segment interface. Finding the segment interface is calculating the line normal and line constant at each multi-fluid cell. Details of applying two normal calculation methods, differential and geometric least squares (GLS) methods, are given. While the GLS method exhibits second order accurate approximation in reproducing circular interfaces, the differential least squares (DLS) method results in a first order accurate representation of the interface. The last part of the algorithm which is remapping of the volume materials from the Lagrangian grid to the original one is performed by a series of polygon intersection procedures. The behavior of the algorithm is investigated for flow fields with constant interface topology and flow fields inducing large interfacial stretching and tearing. Second order accuracy is obtained if the velocity integration as well as the reconstruction steps are at least second order accurate.

1. Introduction

1.1. Motivation

Many diverse areas of industry deal with multi-phase and free surface flows. These flows are mainly characterized by the discontinuous interfaces carried out within the flows. To study and understand such

* Corresponding author.
E-mail addresses: Shahbazi@mie.utoronto.ca (K. Shahbazi), marius@mie.utoronto.ca (M. Paraschivoiu), mostag@mie.utoronto.ca (J. Mostaghimi).

0021-9991/03/$ - see front matter
doi:10.1016/S0021-9991(03)00156-6

flows, numerical simulation have been very successful during the last decades. Recent achievements and different numerical methods have been discussed in the review paper by Scardovelli and Zaleski [1].

One of the most practical techniques for solving the conservation of mass in the presence of moving material interfaces is the volume tracking (volume of fluid). Several codes based on volume tracking idea on Cartesian grid, namely SOLA-VOF [2,3] and its descendants NASA-VOF2D [4], NASA-VOF3D [5], RIPPLE [6,7] and FLOW3D [8,9] are widely used for modeling interfaces and free surfaces in industrial applications. For example, a modified version of SOLA-VOF has been used by researcher to model the flow in thermal ink jet devices [10] and SOLA-VOF and FLOW3D have been used extensively by material scientists to model problems involving melting and solidification (e.g. [11,12]).

However, all of these codes are built around a relatively crude volume tracking interface reconstruction algorithm that relies on a piecewise-constant or "staircase" representation of the interface and the advection algorithms that are at best first order accurate. More modern volume tracking interface reconstruction method use a linear approximation to the interface in each multi-fluid cell (e.g., see [13,14]). For example, Bussmann et al. [15,16] developed a 3D modified version of RIPPLE to model droplet impact. Passandideh-fard et al. [17] added a solidification capability to that code to model the molten droplet impact with solidification and plasma spray coating. All of the above citations piecewise linear interface reconstruction algorithms still produce a first order accurate approximation of the front.

In modeling more delicate physics (e.g., capillarity), first order approximation to the front is inadequate because accurate calculation of the interface curvature and topology (which leads to surface tension calculation) is essential. Moreover, in simulating flows involving complex geometry (e.g., casting) structured domain partitioning, depending on the degree of the geometry complexity, either fails or leads to inefficient computations. Therefore, to overcome these limitations, developing a higher order accurate volume tracking code amenable to any domain partitioning is a necessity. In particular, a second order accurate algorithm on triangular meshes is desirable because it not only enables addressing flows in complex geometry but also can easily be equipped with an adaptation technique to result in efficient and accurate computations.

1.2. Volume tracking

1.2.1. Eulerian methods

Many numerical models of interfacial flows have taken the form of Eulerian methods. Eulerian methods, in which the mesh is fixed, are ideal for flows with large deformation; but the sharp resolution of interfaces is lost (the problem is typically called diffusion of the interfaces). Among many methods for tracking the interfaces in incompressible flows, volume tracking (VT) or volume of fluid (VOF) has benefited from a widespread use due to its simplicity and automatic capturing of interfaces (e.g., see [18] for the advantages of VT). A typical VT method on an Eulerian frame of reference which is called advection or transport method considers the following fluid volume evolution equation (e.g., see [19] or [20]):

$$\frac{\partial f}{\partial t} + \nabla \cdot (\mathbf{u} f) = 0, \tag{1}$$

where f is the volume fraction. If one considers the domain filled by two different dark and light fluids, f is the percentage of volume cell occupied by the dark fluid. Therefore, it can vary between 0 and 1 inclusively. (Note that the volume of dark fluid and volume material are used interchangeably.) A finite difference discretization of Eq. (1) on an uniform Cartesian 2D domain partitioning leads to

$$f_{i,j}^{n+1} = f_{i,j}^{n} + \frac{\delta t}{\delta x}\left[F_{i-1/2,j}^{n} - F_{i+1/2,j}^{n}\right] + \frac{\delta t}{\delta y}\left[G_{i,j-1/2}^{n} - G_{i,j+1/2}^{n}\right], \tag{2}$$

where $F^n_{i-1/2,j} = (fu)^n_{i-1/2,j}$ denotes the flux of f across the left-hand edge of i,jth cell and $G^n_{i,j-1/2} = (fv)^n_{i,j-1/2}$ denotes the flux across the bottom edge of the i,jth cell, etc., and u and v are the components of velocity vector $\mathbf{u}$ in x and y directions, respectively. Eq. (2) simply states that fluxes are required in order to advance f in time. Therefore two actions must be performed: interface reconstruction and time integration. Interface reconstruction involves locating the interface at each cell and then truncating the material polygon in order to define the polygon truncating volume material. Time integration involves performing an appropriate splitting or unsplitting scheme to transport the volume material along the fluxes. Ref. [18,20] presents a complete review of different schemes for these two stages. Since the present work concerns second order accurate methods, among many reconstruction techniques, we only review the piecewise linear interface calculation (PLIC) because it can potentially lead to second order accuracy to the front. The PLIC approximates the interface at each cell by a segment line. Therefore the reconstruction step requires two major calculations; the line normal and the line constant.

We first review three major normal finding methods applicable to irregular grids. These methods are:

(1) Extension of Youngs' second method has been developed by Rider and Kothe [20] based on novel k-exact polynomial interpolation for continuous data distribution on unstructured meshes of Barth [21]. Since this method is based on the differential minimization, we call it differential least squares (DLS).

(2) Least squares method were originally developed by Pucket [22] to model gas interfaces. Since then it has been extensively used to model compressible flows (e.g., see [23–25]) and incompressible flows (e.g., see [26,27]). In particular Pilliod and Pucket [18] have applied this method in a volume of fluid reconstruction algorithm. They named the method ''Least Squares Volume of Fluid Interface Reconstruction Algorithm''. In this method, the assumed line interface of a reference cell is extended to its circumventing cells. The difference between volume fractions in the set of cells created by the given line and the real volume fraction distribution is minimized in a least squares sense and during an iterative process. As opposed to the previous one, the heart of the Least Squares method is a geometric minimization. Therefore to avoid confusion we call it geometric least squares (GLS).

(3) Swartz method [28] has been used and modified in some papers (e.g., see [20,28,29]). This method is an iterative method. A brief description of the method is as follows. Given an initial guess of normal $\mathbf{n}$ (and obviously a volume fraction), the segment line is reconstructed for a mixed (multi-fluid) reference cell. Assuming that a chosen mixed border cell of the reference cell has the same normal $\mathbf{n}$, the segment line is constructed for the border cell as well. The centroid of these two segment lines specifies a unique direction and equivalently a unique normal. This normal is chosen as a new guess $\mathbf{n}$. The same process is repeated until convergence.

The details of implementation of the first two methods on triangular meshes which are not found in the literature are given in this paper. In addition, errors and convergence rates are demonstrated and discussed extensively.

For finding the line constant, Rider and Kothe [20] have recently developed an automatic method amenable to any domain partitioning, unlike the previous case-by-case method suitable for orthogonal grids. Details will be presented in the methodology section.

Although the reconstruction part of the Eulerian VT algorithm is extendible to fully unstructured meshes, the time integration part appears to be complicated and expensive for irregular meshes. In unsplit time integration schemes (e.g. [18,20]), to ensure second order temporal accuracy, the corner fluxes must be considered correcting the edge fluxes. Calculating the corner fluxes for irregular grids means performing the intersection process for as many time as the number of neighbors of a reference cell. This calculation is generally higher than what is required for orthogonal meshes.

1.2.2. Lagrangian–Eulerian methods

Alternative methods that do not deal with the fluxes to advance volume fraction f in time are Lagrangian methods. Lagrangian methods, in which the computational mesh travels with the fluid, are

ideal for many problems which involve interfaces between materials or free surfaces. However, multidimensional Lagrangian calculations can typically be carried out for only a limited time before severe mesh distortion, or even mesh tangling destroys the calculation. To overcome these individual limitations, methods have been developed which combine the features of both the Lagrangian and Eulerian methods. These are called Lagrangian–Eulerian (LE) or remapping methods. The integral form of the fluid volume evolution equation can represent the Lagrangian form of the methods as follows:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V_{\mathrm{L}}(t)} f\,\mathrm{d}V = 0, \tag{3}$$

where $\mathrm{d}V$ is the element of volume, and the integration is over an arbitrary finite Lagrangian volume $V_{\mathrm{L}}(t)$ at time t, a volume whose bounding surface moves with the local fluid velocity, and always contains the same material particles. Eq. (3) indicates that the volume of the dark fluid of a Lagrangian volume is conserved along trajectories. This equation may be used to solve for the time evolution of f. Since the dark fluid volume of the Lagrangian control volume is

$$V(t) = \int_{V_{\mathrm{L}}(t)} f\,\mathrm{d}V, \tag{4}$$

a discretization of these two equations is simply given by

$$V(t^{n+1}) = V(t^n), \tag{5}$$

where the superscript n stands for the current time and $n+1$ stands for some unspecified future time.

Eqs. (3) and (4) may be interpreted and applied in two distinct ways, forward and backward trajectories, as presented by Dukowicz and Baumgardner [30] in modeling of the transport equation.

(a) *Forward trajectories*. The current grid with known volume fraction f is projected to the future time t^{n+1}. We may reconstruct the volume of the dark fluid V_{L}^{n+1}, on each Lagrangian cell. The reconstruction scheme can be the PLIC approximation. We now wish to transfer the new-time dark fluid material to some target grid, which typically will be the same as the current grid if one is interested in an Eulerian method. This is obtained as follows:

$$V_{\mathrm{T}}^{n+1} = \int_{V_{\mathrm{T}}} V_{\mathrm{L}}^{n+1}\,\mathrm{d}V, \tag{6}$$

where the subscript T stands for the target grid. Eq. (6) implies integration on the overlap of grids T and L.

(b) *Backward trajectories*. The target grid, which is the desired grid at time t^{n+1}, is projected back in time to the departure points of the trajectories at time t^n. This gives the Lagrangian grid L at time t^n, i.e., $V_{\mathrm{L}}(t^n)$. However, we have the volume fraction at time t^n on our starting grid, which we assume for simplicity to be the same as the target grid. This volume fraction may be reconstructed to give the dark fluid volume at each target cell V^n. Eq. (4) is now interpreted to give

$$V_{\mathrm{T}}^{n+1} = \int_{V_{\mathrm{L}}(t)} V_{\mathrm{T}}^{n}\,\mathrm{d}V. \tag{7}$$

Again (7) implies integration on the overlap of grids T and L. Note that in this case the reconstruction takes place on the target grid, which may be somewhat simpler because the target grid is usually more regular than the Lagrangian grid.

Evaluating the integral in Eqs. (6) and (7) which is called the remapping phase has been the most difficult part of Lagrangian–Eulerian methods (e.g., see [30]) in solving classic advection or continuity equations.

In the present work, since we are dealing with a geometric solution of 2D incompressible flows, the integral part can be handled in a series of polygon intersections.

In addition to specific advantage of remapping methods over advection ones in addressing irregular grids, the remapping methods does not offer any inherent time step limitation unlike the advection methods. The numerical stability of the Eulerian schemes on a finite grid generally requires that the time step be limited by the CFL condition (Courant–Friedrichs–Lewy condition) of the form $\mathrm{Max}|\mathbf{u}|\delta t/\delta x \leqslant 1$; but in methods employing remapping, there is no inherent time step limit. In principle, one can go forward or backward along trajectories as far as one wishes. However, there is a practical limit of the form $\mathrm{Max}|\nabla\mathbf{u}|\delta t < 1$ to prevent trajectory intersections [24]. There is a price to pay for a longer time step: grid L is in principle arbitrarily located with respect to the grid T, and to go from one to other requires a costly search when the CFL condition is not satisfied.

Note that the LE (remapping) methods belongs to the broad family of arbitrary Lagrangian–Eulerian (ALE) methods. For an introduction to ALE methods see the paper by Hirt et al. [31] and its reprint [32] and for recent application of ALE to two-fluid and front tracking problems see [33,34], respectively. In an ALE method, the Lagrangian equations are solved with the mesh following the fluid (Lagrangian phase). Due to possible mesh tangling over time in complex velocity fields, a rezoning phase is necessary. This phase includes defining a new target grid as close as possible to the Lagrangian grid and transferring the Lagrangian solution to the target grid. On the other hand, in the LE the target grid is always the original Eulerian grid. Thus, we do not deal with the issue of choosing the optimum rezoning strategies as recently discussed by Knupp et al. [35]. Furthermore, our Lagrangian phase is employed as an essential technique to advect volume materials accurately on fully unstructured (Eulerian) meshes. This can be viewed as a remedy to the Eulerian counterpart which has not been extended to unstructured grids due to its difficulties in evaluating edge and corner fluxes.

A few investigations can be found for interface tracking using the remapping method. Mosso et al. [28,29] reported a VT algorithm on unstructured meshes associated with the remapping technique. Their algorithms employ irregular but logically rectangular grids in two dimensions. They used Swartz's method to calculate the interface normals. They verified their model for a simple translation velocity field. The authors also examined the solid body rotation test [28]. They analyzed the same algorithm for different numerical schemes to find the arrival point of trajectories (integration of velocity filed in time to find the trajectories). In the exact integration, mass is conserved exactly, but in forward and backward Euler, the initially circular body undergoes considerable expansion and contraction, respectively. For revolving velocity $w = 2\pi$ and $\delta t = 1/80$, the areas magnify by the factor of 1.002 by using the Euler forward scheme. For the trapezoidal rule, associated with the Crank–Nicholson scheme, they reported 1.0005 amplification factor. Note that Euler forward and backward schemes lead to first order time integration accuracy, while trapezoidal rule is second order accurate. However, they have not investigated the performance of the algorithm for deforming velocity fields.

The focus herein is on developing a second order accurate volume tracking algorithm on triangular grids. The paper has the following objectives:

- Describe in detail the application of DLS and GLS normal finding methods on triangular grids.
- Verify the accuracy of those methods in reproducing the circular interfaces by both qualitative results and error measurements.
- Develop the forward trajectory VT remapping algorithm on triangular meshes.
- Verify the developed VT method on two different evolutionary problems, (a) constant interface tests including simple translation and solid body rotation and (b) a vortical flow field that stretches and potentially tears any interface carried within the flow.

The remainder of this paper is structured as follows. Section 2 details the different parts of the VT algorithm. Section 3 presents some numerical experiments verifying the accuracy and fidelity of the algorithm. And finally, Section 4 concludes this work.

2. Reconstructing the interface

To better describe the algorithm used to reconstruct the interface, the interface segment at each computational cell is defined by

$$\mathbf{n} \cdot \mathbf{x} + \rho = 0, \tag{8}$$

where $\mathbf{n}$ is a unit normal vector to the line representing the interface segment, $\mathbf{x}$ is a point on the line and ρ is the line constant. To locate the interface at each cell with a given volume fraction distribution, one can first calculate the normal and then the corresponding line constant for each line segment. The line constant ρ follows from enforcing the volume conservation and the interface normal $\mathbf{n}$ follows from volume fraction gradients. Extension of Youngs' second method and least squares will be used for calculating normals. Based on [20], ρ is determined.

2.1. Finding the interface normal

We first present the DLS method on triangular meshes. In details, the neighborhood (tessellation) chosen to calculate the normal at each interfecial cell in least squares manner will be illustrated. Since this method is at best first order accurate, we then detail the GLS method on triangular meshes which leads to second order accurate in that it reproduces line interface exactly as Pilliod [18,36] conjectured. Since the GLS method is an iterative method, we will use the DLS solution as an initial guess.

2.1.1. Differential least squares

In the DLS method, volume fraction Taylor series expansions of f_i^{TS} are formed from each reference cell volume fraction f_i at point $\mathbf{x_i}$ to each cell neighbor f_k at point $\mathbf{x_k}$. The sum $(f_i^{\mathrm{TS}} - f_k)^2$ over all n immediate neighbors is then minimized in the least squares sense using the normal equations. For triangular meshes, the n immediate neighbors are triangles having at least one vertex in common with the ith reference triangle. Fig. 1(a) presents the reference cell (the bold cell) and its circumventing neighbors contributing in calculating the normal. All coordinates are evaluated in the mass center of triangles. The L_2 norm minimization yields the volume fraction gradients as solution to the linear system (in 2D)

$$A^{\mathrm{T}} A \mathbf{x} = A^{\mathrm{T}} \mathbf{b}, \tag{9}$$

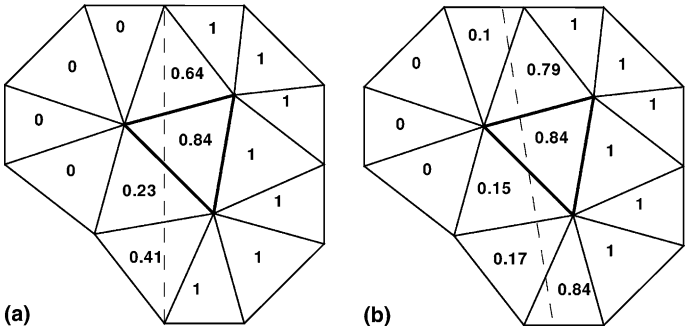

Fig. 1. The tessellation of the bold cell (reference cell). (a) Exact line interface (dashed line) and the corresponding volume fractions. (b) The extension of the reconstructed line interface of the bold cell through the tessellation using DLS and the corresponding volume fractions.

where

$$A = \begin{pmatrix} w_k(x_k - x_i) & w_k(y_k - y_i) \\ \vdots & \vdots \\ w_n(x_n - x_i) & w_n(y_n - y_i) \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} w_k(f_k - f_i) \\ \vdots \\ w_n(f_n - f_i) \end{pmatrix},$$

where $w_k = 1/|\mathbf{x_k} - \mathbf{x_i}|^t$ ($t = 2$ is taken) and

$$\mathbf{x} = \begin{pmatrix} \nabla_x f_i \\ \nabla_y f_i \end{pmatrix}.$$

The normal is then computed as

$$\mathbf{n} = \begin{pmatrix} \frac{\nabla_x f_i}{|\nabla f_i|} \\ \\ \frac{\nabla_y f_i}{|\nabla f_i|} \end{pmatrix}, \tag{10}$$

where

$$|\nabla f_i| = \sqrt{(\nabla_x f_i)^2 + (\nabla_y f_i)^2}.$$

Exact calculation of reconstruction gradients for linear f is guaranteed. However, even for linear interface, the f distribution through the tessellation is not linear (Fig. 1(a)). Therefore, DLS does not yield second order accuracy as can be seen in Fig. 1(b) where the DLS solution results in a different line for the interface. Fig. 1(b) also shows that the approximated interface for the bold triangle produces the volume fractions which are different from those produced by the exact interface (Fig. 1(a)). Minimizing of these difference is the heart of GLS method which has been proved to be second order accurate.

2.1.2. Geometric least squares

We now consider the tessellation of triangles in Fig. 1, circumventing the bold triangle. Let $f(x)$ be a curve that passes through the bold cell and let the f_i for $i = 1, \ldots, n$, where n is the number of triangles, represent the volume fractions due to the function f, in the tessellation. Now let $\tilde{f}$ be a linear approximation of f with slope $\tilde{m}$ and the volume fractions $\widetilde{f_i}$ and assume that $\tilde{f}$ has the same volume fraction in the bold cell as f; i.e., $f_\mathrm{b} = \widetilde{f_\mathrm{b}}$, where b stands for bold cell. Define E_i^2 to be the discrete L_2 error between the volume fractions in the tessellation of triangles circumventing the bold triangle,

$$E_i^2(\tilde{m}) = \left(\sum_{i=1}^{n} (\widetilde{f_i}(\tilde{m}) - f_i)^2 \right)^{1/2}. \tag{11}$$

In the GLS algorithm one minimizes E_i^2 as a function of $\tilde{m}$ by rotating the line $\tilde{f}$ under the constrain that this line exactly reproduces the volume fraction in the bold triangle, $f_\mathrm{b} = \widetilde{f_\mathrm{b}}$.

Note that if $f(x)$ is linear, the GLS algorithm reconstructs the line exactly, assuming the minimization procedure will always find the global minima. This claim is true because the E_i^2 has the minimum value of 0 when $f(x)$ is linear.

In practice $\tilde{f}(x)$ is considered as a function of the angle $\tilde{\theta}$ measured with respect to the horizon instead of $\tilde{m}$. Therefore θ can vary in the interval of 0 and $2 \times \pi$. To obtain the global minima, one may first divide the interval into infinitesimal subdomains. He then evaluates the $E_i^2(\tilde{\theta})$ in all subdomains to get the global minima. However, this method is not efficient due to its large number of $E_i^2(\tilde{\theta})$ evaluations. The practical way starts with an initial guess and uses a 1D minimization method to reach the minima, where Brent's

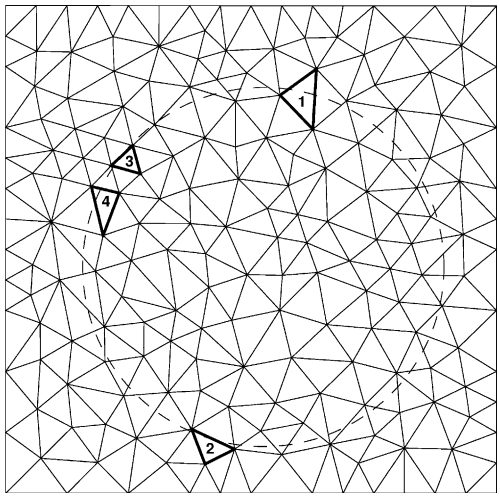

Fig. 2. Circular interface, dashed line, with a radius of 0.368 centered at (0.525, 0.464) on a unit triangulated square. The L_2 norm errors between the exact volume fractions and linear approximations for the mixed bold triangles are plotted in Fig. 3.

algorithm (e.g. [37]) is used to find the minima. The DLS solution offers an ideal initial guess. Usually minimization algorithms need the minima to be banded before iterations start. Brent's algorithm is an iterative method that fits a parabola over the interval, and uses the minimum of the parabola as the next guess for the minimum of the given function. If it cannot fit a parabola, it does a golden section search. The method stops when both the interval and consecutive guesses are within a given tolerance.

In our case, we gradually change the initial guess in two directions until the errors at endpoints of the interval become greater than the one given by the initial guess. How do we guarantee that the initial interval includes the global minima not the local one? Fig. 2 shows a circular volume distribution on the triangular meshes. Fig. 3 illustrates the variation of the $E_i^2(\tilde{\theta})$ with respect to $\tilde{\theta}$ on the bold triangles represented in Fig. 2. Fig. 3 also depicts the DLS solution used as initial guess for those cells. We observe that except for the triangle number 2 on Fig. 2, the other three plots just have one minima. The plot corresponding to the triangle 2 has several minima but the difference between the local minima and the global is more than $\pi/2$ radians. This difference is substantially greater than the difference between the DLS solution and the global minima as depicted in Fig. 3. However, the proof that the initial interval based on the DLS solution whether includes or not the global minima for any interface topology is beyond the scope of this paper. We just claim that it works for our cases because we obtain second order spatial accuracy for all experiments.

Note that in order to evaluate the E_i^2 at each $\tilde{m}$ the interface is reconstructed at all triangles in the tessellation. It means that at each iteration of the minimization procedure n line constant calculations are performed. Since n is multiplied by the number of iterations, the GLS appears to be prohibitively expensive in particular for 3D problems.

2.2. Finding the line constant

The value of ρ is governed by a volume conservation law. In other words, the value of ρ is constrained such that the resulting line passes through the cell with a truncation volume equal to the cell material volume V. This determination requires inverting a relation $V(\rho)$, in which V can vary linearly, quadratically, or cubically with ρ, depending upon the coordinate system and the shape of the n-sided polygon formed by the interface segment truncating the cell. Here $V(\rho)$ is the material volume in the cell bounded by

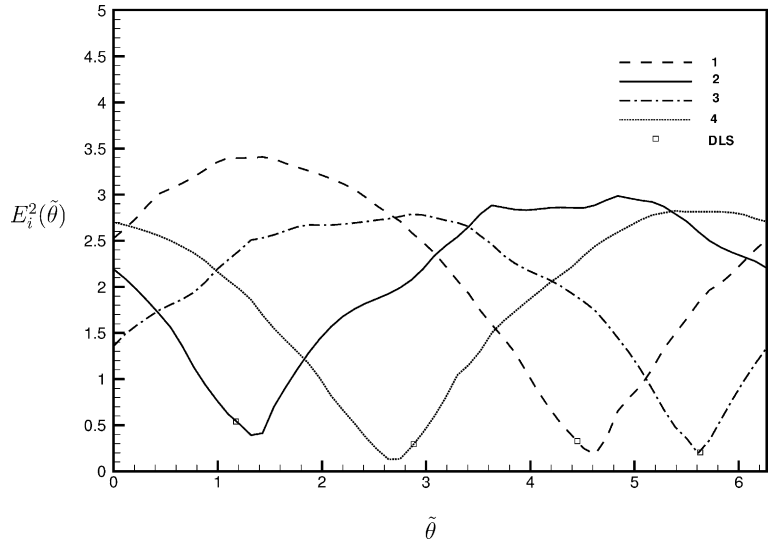

Fig. 3. L_2 error between volume fractions for circular interface at four bold triangles in Fig. 2 versus the angle measured from the horizon. The square marks indicate the DLS solution for those triangles.

the interface segment (with line constant ρ) and the portion of the edges within the material. It is tempting to construct an algorithm for determining ρ based on a direct solution of the relation $V(\rho)$. However, since this relation is often nonlinear and varies in each mixed cell, and since also it depends upon local data, a "case-by-case" implementation showed that it is not efficient (vectorizable or parallel), nor general, nor concise, nor easily maintained and understood, particularly for irregular grids. Instead relation $V(\rho)$ is inverted iteratively in each mixed cell (a mixed cell is a cell with f between 0 and 1). The resulting algorithm is therefore independent of the mixed-cell properties and data.

The line constant ρ is calculated when the general nonlinear function

$$F(\rho) = V(\rho) - V$$

becomes 0. To calculate $V(\rho)$, the truncated volume resulting from the intersection between the assumed line segment and the reference cell is found from an initial guess of ρ. When $V(\rho)$ and V are equal (to within some tolerance), the interface segment is declared "reconstructed" in that cell. To find the zero of this function ($F(\rho)$), Brent's method has been used in this paper. Since Brent's method uses an inverse parabola interpolation, it is suitable for finding the root of $F(\rho)$ as V often varies quadratically with ρ.

A well chosen initial guess for ρ improves convergence. ρ is initialized prior to the iteration in the following manner. Lines possessing the interface normal **n** are passed through each vertex of the polygon cell (triangle), and the resulting truncation volumes are computed. Second, these two lines forming truncation volumes that bound the actual material in the cell, i.e., provide upper ($\rho_{\max}$) and lower ($\rho_{\min}$) bounds for ρ. The initial guess for ρ can then be a linear average of $\rho_{\min}$ and $\rho_{\max}$.

The constant line finding algorithm can be summarized in the following steps as found in [20]:

1. Given an interface line segment, truncate a mixed cell.
2. Find and assemble the n vertices of the polygon formed by those cell vertices inside the fluid and the interface line/cell edge intersection points (using geometric functions).
3. Calculate the volume bounded by this polygon.
4. Determine if the polygon volume differs from the known fluid volume by some prescribed tolerance.

5. If the volumes differ, use Brent's method to find a new estimate for ρ in Eq. (4) and go back to step 1.
6. If the volumes do not differ, the interface line is declared reconstructed.

The intersection in the line constant calculation process can be handled properly by using simple geometric functions as suggested by Rider and Douglas [1]. These functions which are widely used in the field of computational geometry (e.g. [38]) are: (1) line–line intersection, (2) point location, (3) polygon collection and (4) polygon area. They can be freely accessible via http://public.lanl.gov:80/mww/publicas.html.

2.3. Time integration

The time integration part of the VT method employs the forward trajectories remapping. In this algorithm, to evolve volume materials in time, first the original grids (Eulerian grids) are projected along trajectories to obtain the Lagrangian grids. This step is called the Lagrangian phase. The heart of the Lagrangian phase is finding the trajectories. In other words, the velocity field must be integrated from the original grids to Lagrangian grids, and can be written as

$$x_{\mathrm{L}}(t^{n+1}) = x(t^n) + \int_{t^n}^{t^{n+1}} \mathbf{u}\,\mathrm{d}t, \tag{12}$$

where $x_{\mathrm{L}}(t^{n+1})$ is the coordinate of the Lagrangian grid and the $x(t^n)$ is the original coordinate and $\mathbf{u}$ is the velocity vector. Here the integration is carried out in 2D. If the velocity field is analytically integrable in time, exact integration is performed, otherwise, one should use numerical schemes to integrate (1). Note that numerical integration is a practical approach since the real flow often does not even have the exact expression for velocity due to nonlinearity of the Navier–Stokes equations. Therefore, using numerical integration even for the exactly integrable cases reveals more insight into the performance of the method under real situations. For some of our problem tests we use numerical schemes like trapezoidal role, associated with Crank–Nicholson schemes, or second order Range–Kutta techniques for system of differential equations.

For the trapezoidal rule, the new coordinates of Lagrangian meshes are obtained from averaged velocity over the time step multiplied with the time step. The descritized form of Eq. (12) employing the basic Runge–Kutta (midpoint method) for the given velocity field

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{u}(t, \mathbf{x}) \tag{13}$$

is

$$\mathbf{k}_1 = \delta t\, \mathbf{u}(t^n, \mathbf{x}(t^n)),$$

$$\mathbf{k}_2 = \delta t\, \mathbf{u}\left(t^n + \delta t, \mathbf{x}\left(t^n + \frac{1}{2}\mathbf{k}_1\right)\right),$$

$$\mathbf{x}(t^{n+1}) = \mathbf{x}(t^n) + \mathbf{k}_2, \tag{14}$$

where $\mathbf{x}, \mathbf{u}, \mathbf{k_1}, \mathbf{k_2}$ are 2D vectors. $\mathbf{x}, \mathbf{u}$ are coordinate and velocity vectors, respectively. In the next step, the volume materials are reconstructed on the new Lagrangian grids (reconstruction step), assuming that volume fractions remain constant during the Lagrangian phase. The reconstruction step includes calculation of normals, interface segment location and finally volume material truncation at each Lagrangian grid. In Fig. 4 that visualize the VT algorithm, quadrilateral $a_{\mathrm{L}}b_{\mathrm{L}}ed$ (shaded region) is the truncation polygon on the $a_{\mathrm{L}}b_{\mathrm{L}}c_{\mathrm{L}}$.

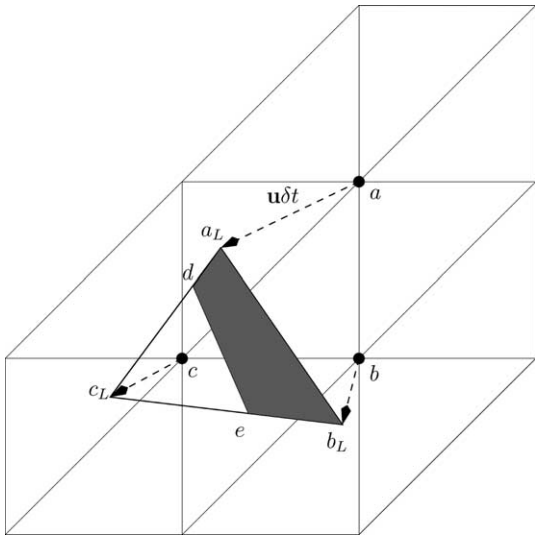

Fig. 4. Forward trajectories remapping. Triangle abc as an original grid, triangle $a_Lb_Lc_L$ the corresponding Lagrangian grid and quadrilateral a_Lb_Led the truncated material polygon on the $a_Lb_Lc_L$.

The assumption of constant f leads to exact results of volume material conservation provided no changes of the original volume occur during the Lagrangian phase (velocity integration). Since all velocity fields studied in this work are divergent free due to incompressibility, the exact integration does not lead to any volume conservation violation, but numerical schemes, like trapezoidal and Runge–Kutta schemes, do not consider the divergent free constrain, so the volume material is not conserved. However, in most cases this error is at least two order of magnitude smaller than the interface error. Therefore it does not defect the solution.

The final step of the VT algorithm is to deposit the volume materials truncated on the Lagrangian grids to the target grids which are the original grids in our algorithm (remapping phase). Remapping phase is performed via a series of polygon intersections if volume materials and target grids are treated as polygons. In practice, depending on the CFL number, some levels of circumventing triangles must be searched for intersection. (Circumventing triangles are triangles circumventing the reference triangle, triangle abc in Fig. 4.) For instance, if the CFL number is substantially smaller than 1, searching one level of circumventing triangles is enough. This searching procedure makes the remapping algorithm to be more expensive than the corresponding advection one.

For intersections, the algorithm developed by Schutte [39] is used. This algorithm is capable of finding an intersection of two possibly concave polygons with quadratic time complexity. Since polygons formed during the reconstruction steps have small number of vertices (four-sided or three-sided polygons for triangular meshes), quadratic time complexity does not considerably defect the complexity of the whole algorithm.

In summary, consider Fig. 4. It illustrates the three steps of the remapping method for the reference element abc. This element moves along velocity trajectories, dashed arrows, to $a_Lb_Lc_L$. This is simply the Lagrangian phase. Forming the quadrilateral a_Lb_Led on the element $a_Lb_Lc_L$ is the reconstruction part. Finally, part of the remapping phase for the reference is finding the intersection of the shaded polygon a_Lb_Led with the reference cell abc. To complete the remapping phase and finding the new volume fraction f for abc, the overlap of abc with other material polygons formed on the neighboring Lagrangian elements must be found.

3. Verifying experiments

In this section, we first examine the accuracy of both the DLS and the GLS interface normal calculation methods. Since no approximation occurs during the line constant process, normal finding tests actually clarify the behavior of the reconstruction portion. We then report the behavior of the VT algorithm on evolutionary problems: simple translation and solid body rotations as primary tests and a vortical flow as a more realist test.

3.1. Interface reconstructing accuracy

The DLS and GLS methods are examined on reproducing a circular body (radius 0.368) centered at (0.525, 0.468). The domain is a unit square partitioned by either unstructured or structured triangles. Structured grids are just used for convergence study. Figs. 5 and 6 show the qualitative comparison between the DLS and GLS methods on reproducing the circular interface.

Qualitative results for the reconstruction algorithms are also presented in the form of L_1 errors and convergence rates. The L_1 error is the absolute value of the area difference between the material corre-

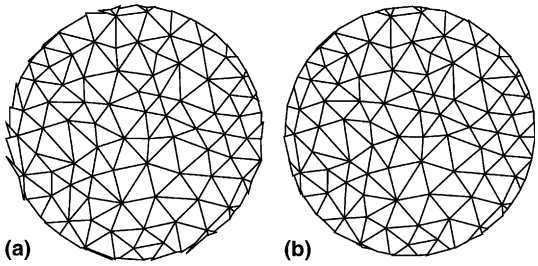

Fig. 5. Piecewise linear reconstructed interfaces for the interface normal **n** on a triangular grid, offset circle. Reconstructed interface using the (a) DLS method and (b) GLS method.

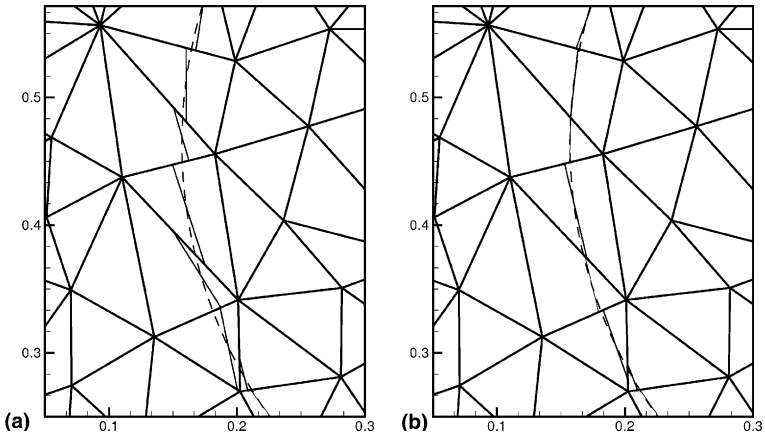

Fig. 6. Piecewise linear reconstructed interfaces on a triangular grid, offset circle. (a) DLS method; zoom of a specific region, dashed line represents the initial circle. (b) GLS method; zoom of a specific region of the interface, dashed line represents the initial circle.

Table 1
L_1 circle reconstruction error

Triangles	Error	Order
(a) Using DLS method		
2×10^2	4.44×10^{-3}	
		0.91
2×20^2	2.37×10^{-3}	
		1.08
2×40^2	1.12×10^{-3}	
		0.93
2×80^2	5.87×10^{-4}	
		1.08
2×160^2	2.77×10^{-4}	
(b) Using GLS method		
2×10^2	2.34×10^{-3}	
		2.03
2×20^2	5.50×10^{-4}	
		2.07
2×40^2	1.31×10^{-4}	
		2.04
2×80^2	3.18×10^{-5}	
		1.89
2×160^2	8.59×10^{-6}	

sponding the exact interface and the approximate solution at each interfacial(mixed) grid. The summation of these portions over all mixed cells gives the L_1 error. Table 1 represents the error and the rate of convergence for 2×10^2, 2×20^2, 2×40^2, 2×80^2 and 2×160^2 structured triangles for the two reconstruction methods.

Table 1 clearly shows the rate of convergence which is first order accurate for the DLS method and second order accurate for the GLS method. First order accuracy of the DLS is due to the discontinuity of the volume fraction f distribution caused by treating the integrated quantity f as a continuous quantity at the center of the grid. This behavior of the DLS has also been reported on uniform orthogonal grids by Rider [20] and Pilliod [18,36]. Since the GLS method is a linear approximation of the interface, the greater the ratio of the grid size over the curvature radius of the real interface, the better the approximation. As seen in Table 1, the finer the grid (with the constant curvature radius) the smaller the GLS error with respect to the DLS error. For example, the error ratio of these two methods is around 2 for the coarsest mesh (2×10^2) and is around 20 for the finest mesh (2×160^2). Fig. 6(b) also shows this concept but in a slightly different way on a rather coarse mesh. In other words, for the GLS method, different local accuracy is recognizable in different grids. For instance, if the cell contains a longer interface, the real interface and the approximated one are not fully matched due to the fact that the cell involves higher curvature.

3.2. Evolutionary problems

To examine the VT algorithm, we set up two inherently different problems. First, translation and solid body rotation are investigated so no deformation is taken place during the evolution. Therefore, the initial shape must be conserved. Second, a vortical flow is investigated; therefore, bodies undergo large deformations enable us to asses the performance of the algorithm on physical problems.

(a) *Translation test*. An initially circular body (radius of 0.25) centered at (0.5, 0.5) of a unit square, is translated the length of two domain diagonals (1 unit) with a unit constant solenoidal (divergent free)

velocity field. The translation time is divided to four smaller time periods in which velocity sign is constant. Therefore, the circular body should return to its initial position after one complete period.

Fig. 7 shows the qualitative result of the translation test on a fully unstructured triangular grid. The CFL number is 0.3 and is based on the characteristic mesh size resulting from the minimum area constraint in mesh generating. The triangulation of 3430 triangles, is charaterized by all areas greater than 0.00048 and all angles greater than 31.5°. (Note that all results, except for convergence studies, occur on this mesh.) (The Triangulation is performed by the freely accessible C code "Triangle.c" via http://gams.nist.gov.) Fig. 7(a) and (c) show the circular body after one complete translation period for the two normal finding methods, DLS and GLS, respectively. Fig. 7(b) and (c) report the corresponding error contours. Error norms can be defined based on some positive-definite measure of the differences observed between the computed and exact values of f. The L_1 norm is defined as

$$E^{L_1} = \sum_{grid} V_{i,j} |f_{i,j}^{\text{computed}} - f_{i,j}^{\text{exact}}|. \tag{15}$$

It is chosen to estimate the computed error.

Quantitative study exploits 2×32^2, 2×64^2, 2×128^2 and 2×256^2 structured triangles. The L_1 error norm and convergence rate are presented in Table 2, (a) on the left corresponding errors of the DLS and (b) on the right the ones of the GLS. A CFL number of 0.5 is used.

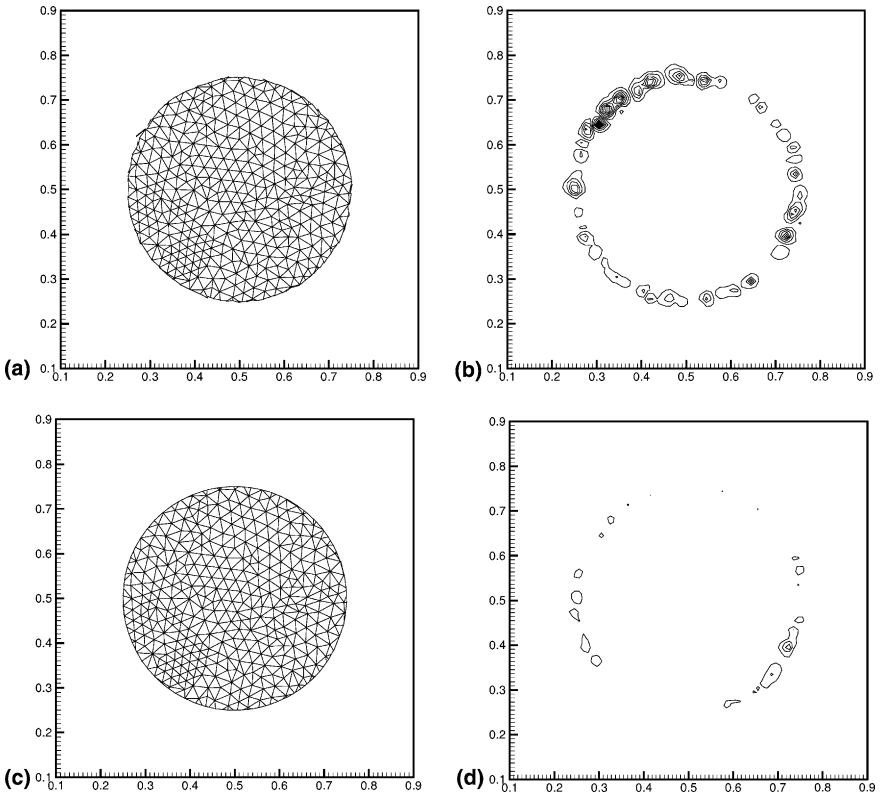

Fig. 7. Performance of the VT method in translating an initially circular fluid body at 45° for a distance of two domain diagonals on 3034 unstructured triangles. (a) Reconstructed interfaces based on the DLS normal calculation. (b) Corresponding error contours (20 contour levels on the interval of -5×10^{-5} and 5×10^{-5}). (c) Reconstructed interfaces based on the GLS normal calculation. (d) Corresponding error contours (20 contour levels on the interval of -5×10^{-5} and 5×10^{-5}).

Table 2
L_1 error norms and convergence rates for a circular fluid body translated two domain diagonals at 45° angle to the grid

Triangles	Error	Order
(a) The DLS method		
2×32^2	9.87×10^{-4}	
		0.88
2×64^2	5.38×10^{-4}	
		0.81
2×128^2	3.06×10^{-4}	
		1.26
2×256^2	1.28×10^{-4}	
(b) The GLS method		
2×32^2	1.60×10^{-4}	
		1.09
2×64^2	7.53×10^{-5}	
		1.61
2×128^2	2.46×10^{-5}	
		0.83
2×256^2	1.39×10^{-5}	

The VT algorithm associated with the GLS method performs better than the DLS method as it is clear from Table 2 and Fig. 7. The GLS represents results with more accuracy. This is also obvious from the error contours in Fig. 7(d) which are fewer than the ones in Fig. 7(b).

The rate of convergence is around 1 for the DLS method. However, the GLS method does not appear second order accurate. What is the problem? Since throughout this test the number of time steps or equivalently the number of reconstruction increases with respect to mesh refinement (under the condition of a constant CFL number and constant evolution distance of 1 unit), the accumulation of error due to reconstruction also increases. Note that the DLS method is insensitive to this kind of error due to its one order of magnitude larger error values. Therefore, we should set up a test unbiased for different resolutions. This is achieved by translating the circle with constant number of time steps. Table 3 supports this argument showing the errors and rate of convergences. In Table 3(a) the number of time steps is 4 and in Table 3(b) it is 16. A CFL number of 0.5 is used similar to the previous test. It is clear from this table that the second order accuracy is indeed obtained under mesh refinement.

(b) *Rotation test*. An initially circular body (radius of 0.15) centered at (0.5, 0.75) revolves around the center of the unit squared domain with the unit constant angular velocity for one complete revolution. The similar qualitative results are presented with a CFL of $\pi/12$, Fig. 8. The convergence study are also reported in the same manner but with a CFL of $\pi/6$. All of these results are associated with the exact velocity integration. However, in order to study the performance of numerical integration, we present the quantitative results of VT algorithm associated with the trapezoidal time integration for the GLS reconstruction algorithm. In Table 4, the volume error due to the numerical integration is reported in addition to L_1 error norm and the convergence rate.

Fig. 8 and Table 4 clearly show that the GLS method is more accurate than the DLS method for the solid body rotation. The GLS method is second order accurate while the DLS appears to be first order accurate. Table 5 shows that the behavior of the reconstruction error and the rate of convergence in the case of trapezoidal numerical integration is almost the same as the exact integration reported in Table 4 for the GLS normal calculation. Since the spatial accuracy governed by the reconstruction portion of the VT is second order accurate, the exact velocity integration with the second order spatial accuracy just yields the VT of the second order accurate which is also achieved by using the Trapezoidal rule integration. However, in contrast to the previous cases, the mass is not conserved exactly due to violating divergent free constraint

Table 3
L_1 error norms and convergence rates for a circular fluid body translated at 45° angle to the grid; using the GLS method and a CFL of 0.5 and constant number of time step for different resolutions

Triangles	Error	Order
(a) Four time steps		
2×32^2	1.23×10^{-4}	
		1.76
2×64^2	3.62×10^{-5}	
		2.11
2×128^2	8.41×10^{-6}	
		2.09
2×256^2	1.98×10^{-6}	
(b) Sixteen time steps		
2×32^2	1.70×10^{-4}	
		1.59
2×64^2	5.56×10^{-5}	
		2.07
2×128^2	1.35×10^{-6}	
		2.00
2×256^2	3.38×10^{-6}	

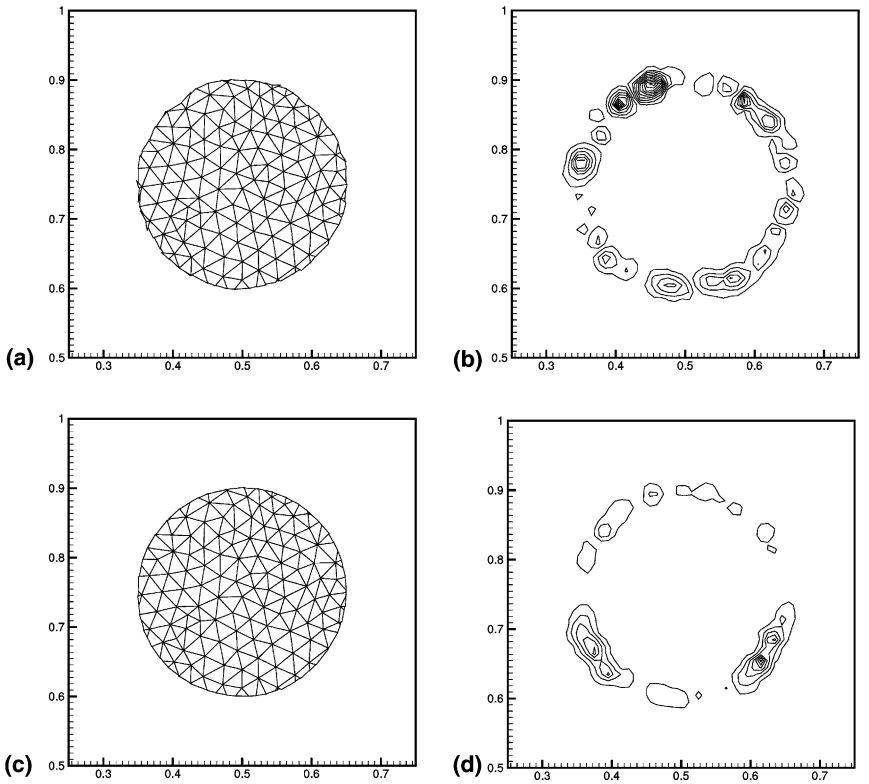

Fig. 8. Performance of the VT method in rotating an initially circular fluid body for one revolution on a 3034 unstructured mesh. (a) Reconstructed interfaces based on the DLS normal calculation. (b) Corresponding error contours (20 contour levels on the interval of -5×10^{-5} and 5×10^{-5}). (c) Reconstructed interfaces based on the GLS normal calculation. (d) Corresponding error contours. (20 contour levels on the interval of -5×10^{-5} and 5×10^{-5}.)

Table 4
L_1 error norms and convergence rates for a circular fluid body rotated one complete circle associated with exact velocity integration

Triangles	Error	Order
The DLS method		
2×32^2	1.10×10^{-3}	
		0.81
2×64^2	6.27×10^{-4}	
		0.91
2×128^2	3.33×10^{-4}	
The GLS method		
2×32^2	6.50×10^{-4}	
		1.95
2×64^2	1.68×10^{-4}	
		1.89
2×128^2	4.54×10^{-5}	

Table 5
L_1 error norms, convergence rates and volume errors for a circular fluid body rotated one complete circle associated with trapezoidal velocity integration and the GLS method

Triangles	Error	Order	Volume error
2×32^2	6.40×10^{-4}		1.30×10^{-6}
		1.97	
2×64^2	1.63×10^{-4}		1.62×10^{-7}
		1.90	
2×128^2	4.37×10^{-5}		2.03×10^{-8}

during the numerical integration; but it does not affect the solution because the volume errors are at least 2 order of magnitude smaller than the reconstruction errors. Conservation of mass and other quantities are important issues associated with the Lagrangian phase of any ALE method [31,32]. In our case, instead of the forward trajectory based method we recommend backward trajectory remapping that guarantees mass conservation exactly.

(c) *Vortical flow*. Our last test problem imposes a vortical flow that stretches and potentially tears any interfaces carried within the flow. It contains a single vortex that will spin fluid elements, stretching them into a filament that spirals toward the vortex center. The initial condition of the test is the same as the rotation test. Pictures on the right sides of Fig. 9 resembles the exact behavior of the flow. These pictures are obtained by considering the initial circular body as fluid particles on a fined grid (2×250^2 triangles) and simply evolving those particles in time with a simple Euler forward scheme albeit with a very small time step (0.001).

For the VT model of the flow, the vertices of the grid are integrated in time using the basic Range–Kutta method (the midpoint method) to obtain the Lagrangian grid. The GLS normal calculation method is used in the reconstruction procedure. The VT model of the flow is presented on the left of Fig. 9 which are performed with a CFL of $\sqrt{8}/3$ on the same unstructured triangulation of the unit square. By integrating to late times, it is possible to observe the behavior of our VT method under rather extreme circumstances, whereby filaments become thinner than it is supportable by the computational mesh. In Fig. 9, the solution becomes poor when interface topology is not resolved, i.e., as exhibited by single filament breaking into series of fluid clumps that are each supportable by the reconstruction method. By $t = 4.5$ (Fig. 9(j)) the body has fragmented into numerous pieces.

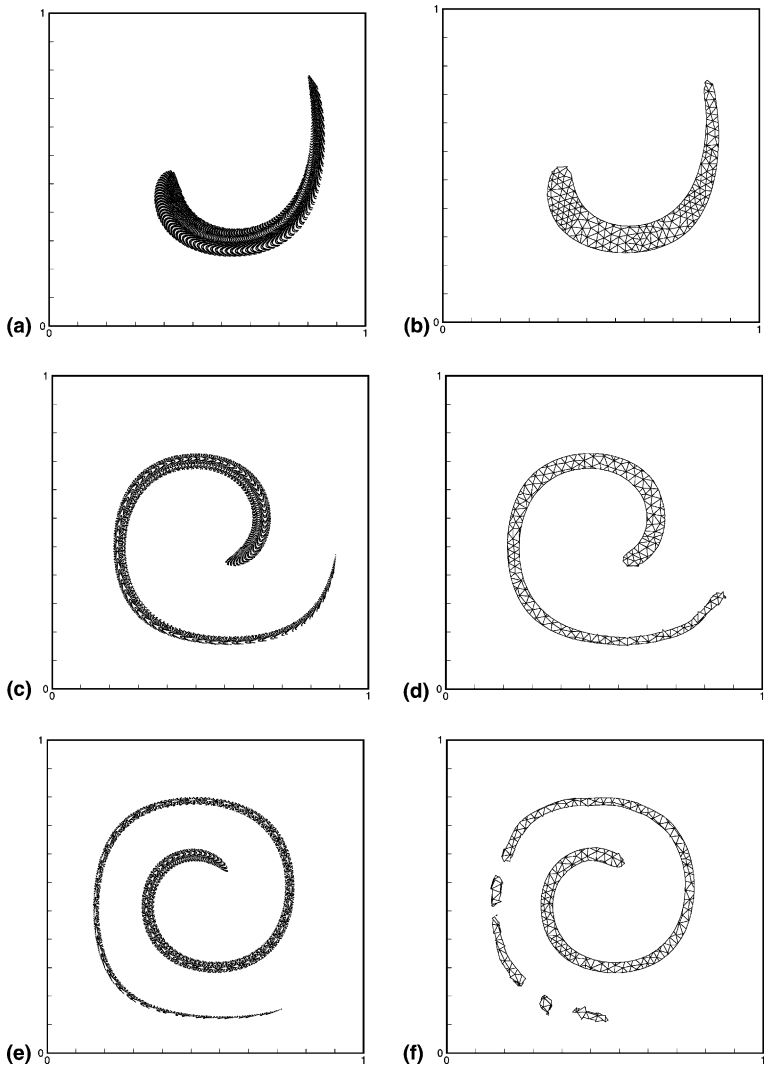

Fig. 9. Long time integration of a circular fluid body (Fig. 9) placed in the single-vortex flow field for both exact resembled particle solution on a 2×250^2 triangles and the VT model using the GLS normal finding on 3430 triangles. (a) "Exact" solution at $t = 0.75$. (b) VT simulation at $t = 0.75$. (c) "Exact" solution at $t = 1.5$. (d) VT simulation at $t = 1.5$. (e) "Exact" solution at $t = 2.25$. (f) VT simulation at 2.25. (g) "Exact" solution at $t = 3.0$. (h) VT simulation at 3.0. (i) "Exact" solution at $t = 4.5$. (j) VT simulation at 4.5. (k) "Exact" solution at $t = 6$. (l) VT simulation at 6.

Convergence results are obtained by time-reversing the flow using Leveque's cosine term. As the reversal period T becomes longer, the fluid body evolves further away from its initial circular configuration; hence, it must undergo increasingly complicated topological change to reassemble properly at $t = T$. Convergence results for the single vortex velocity field indicate that our method is remarkably resilient. The method exhibits second order convergence, even for long periods ($T = 8$), where appreciable interface tearing and topological change have occurred. This is indicated by the L_1 error norms and convergence results shown in Tables 6–8. Fig. 10 illustrate that solution errors are more evident for a longer reversal period.

The mass error is also presented in Table 6. Again, it is at least one order of magnitude smaller than the interface error; therefore, it does not nullify our results.

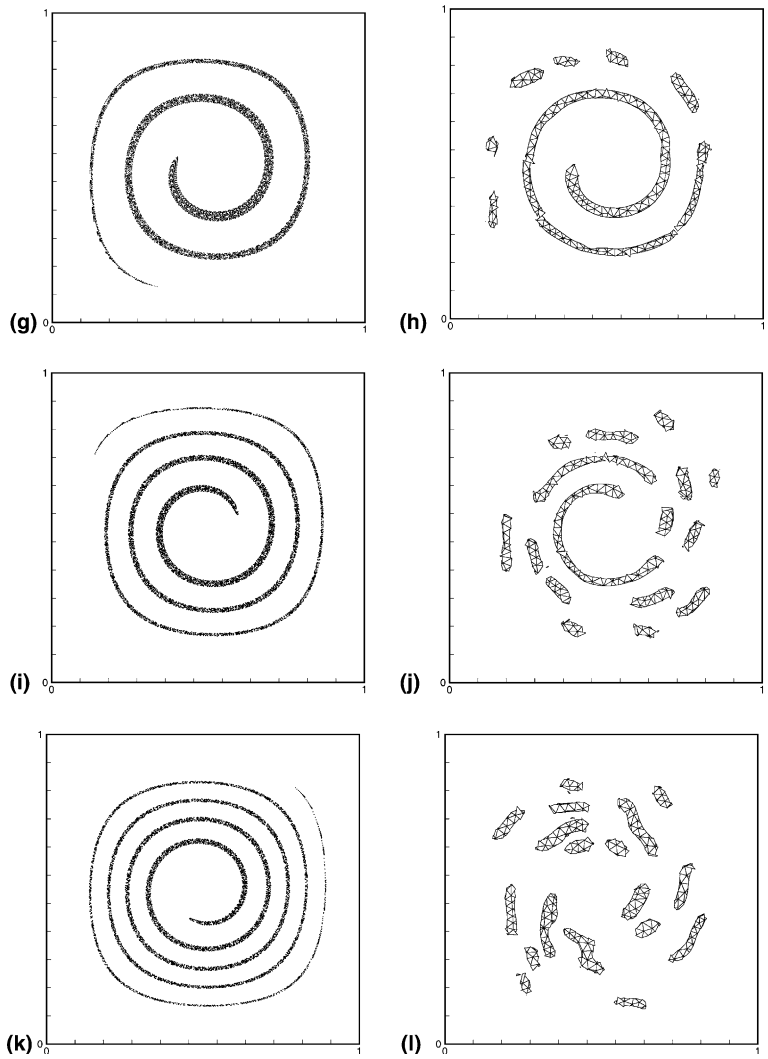

Fig. 9. (*continued*)

Table 6
L_1 error norms, convergence rates and volume error of the single vortex flow field for $T = 0.5$ and using the GLS method as the normal finder

Triangles	Error	Order	Volume error
2×32^2	3.28×10^{-4}		1.26×10^{-5}
		1.88	
2×64^2	8.94×10^{-5}		1.59×10^{-6}
		2.43	
2×128^2	1.66×10^{-5}		1.86×10^{-7}

Table 7
L_1 error norms, convergence rates and volume error of the single vortex flow field for $T = 2.0$ and using the GLS method as the normal finder

Triangles	Error	Order	Volume error
2×32^2	1.85×10^{-3}		4.43×10^{-5}
		1.82	
2×64^2	5.25×10^{-4}		5.44×10^{-6}
		1.89	
2×128^2	1.42×10^{-4}		7.44×10^{-7}

Table 8
L_1 error norms, convergence rates and volume error of the single vortex flow field for $T = 8.0$ and using the GLS method as the normal finder

Triangles	Error	Order	Volume error
2×32^2	3.55×10^{-2}		1.57×10^{-4}
		2.31	
2×64^2	7.17×10^{-3}		2.65×10^{-5}
		2.32	
2×128^2	1.44×10^{-3}		2.80×10^{-6}

This behavior is reasonable and expected, given the assumptions inherent in the reconstruction, namely a piecewise linear interface approximation. The break up exhibited in Fig. 9 can be interpreted as an application of "numerical surface tension" along interfaces that are resolved inadequately. High curvature regions are those unresolvable regions having interfaces with a radii of curvature less than roughly a mesh spacing. The piecewise linear approximation immediately flattens these regions, effectively applying numerical surface tension. Thin filament regions can also be the recipients of numerical surface tension, because poor linear reconstructions occur in these regions from inaccurate interface normal estimations. Numerical surface tension in these high curvature and thin filament regions can be easily reduced with increased refinement.

4. Conclusion

A piecewise linear remapping (LE) volume tracking algorithm on triangular meshes is presented in this paper. The details of the normal line finding is given for two methods; differential least squares (DLS) and geometric least squares (GLS). Applying these two methods for reproducing circular interfaces leads to first order accuracy for the DLS method and second order accuracy for the GLS method. Moreover, the error associated with the GLS method is substantially smaller than the DLS method. The volume tracking algorithm is also examined under two inherently different evolutionary problems: (a) a translation test and a solid body rotation test and (b) a vortical flow field test.

The first test series that induce no topology changes not only demonstrate the behavior of the volume tracking algorithm in translation and rotation, but also, compare the performance of the associated DLS and GLS normal calculation methods. Again, the volume tracking with the GLS method appears to be remarkably superior to the DLS method in both error values and convergence rates. Since the Lagrangian phase of the remapping method is handled through exact time integration of the velocity field, second order accuracy in the rotation test is achieved. In order to examine the volume tracking method with inexact velocity integration, the rotation test is repeated with numerical Trapezoidal rule integration. Since the

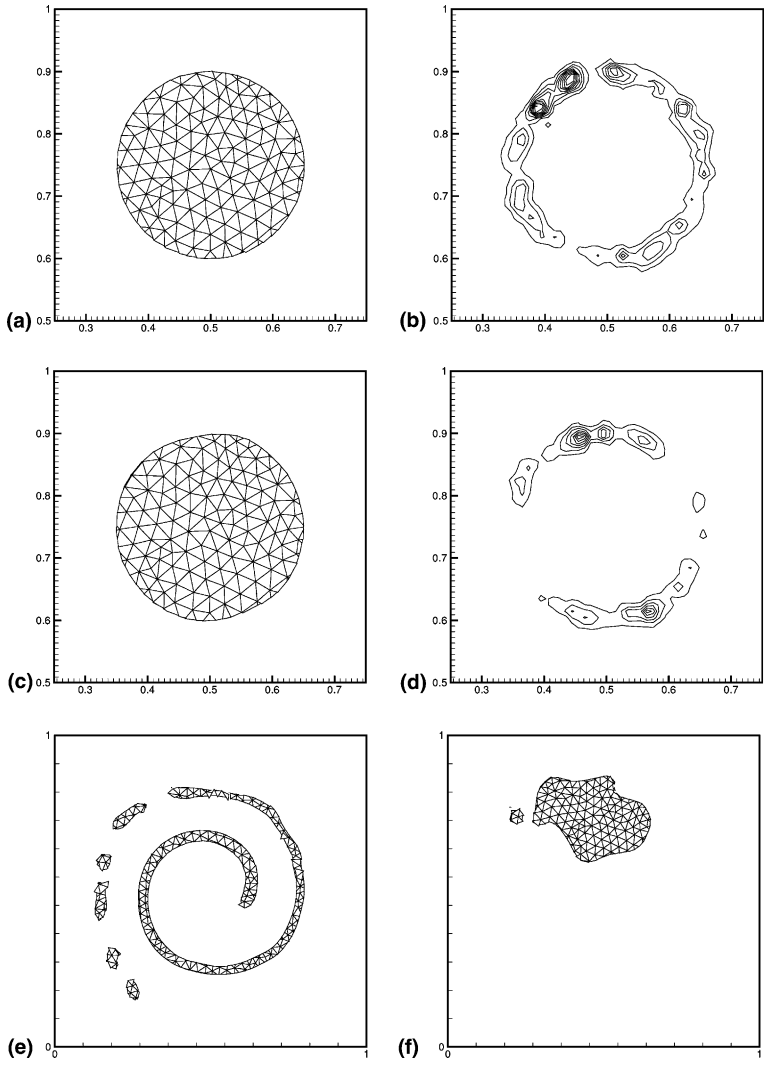

Fig. 10. Results for a circular body placed in the time-reversed, single-vortex flow field on 3430 triangles using GLS as normal finder. (a) Reconstructed interfaces at $t = T$ for $T = 0.5$. (b) Error contours at $t = T$ for $T = 0.5$ (20 contour levels in the interval of -10^{-5} and 10^{-5}). (c) Reconstructed interfaces at $t = T$ for $T = 2.0$. (d) Error contours at $t = T$ for $T = 2.0$ (20 contour levels in the interval of -9×10^{-5} and 9×10^{-5}). (e) Reconstructed interfaces at $t = T/2$ for $T = 8.0$. (f) Reconstructed interfaces at $t = T$ for $T = 8.0$.

trapezoidal rule is second order accurate, the accuracy and convergence rate is almost the same as the previous exact integration. However, volume is not conserved; although, it does not defect our solutions. Mass conservation violation is due to the forward trajectory remapping in which the reconstruction, with constant volume fraction during the Lagrangian phase, occurs on the Lagrangian grid that may have different volume compared to the original grid. (It may expand or contract depending on the approximated velocity integration).

Our last test imposes a vortical flow field that contains a single vortex. It stretches and potentially tears any interface carried within the flow. To find the trajectories, the basic second order Runge–Kutta integration is carried out on the corresponding velocity field. The qualitative results on a rather coarse grid are

remarkably good for short periods of times; but, become poor for long periods, as the filaments break up into numerous pieces. The break up is due to existence of high curvature and thin filament regions that are not resolvable by comparably coarse computational cells. Despite the break up, quantitative convergence (with the quadratic rate) does occur under grid refinement.

Several aspects of the algorithm deserve further attentions. First, the algorithm can be equipped with adaptation techniques to capture accurately and efficiently high curvature and thin filament regions of flows. Adaption can be easily implemented, being an important feature of the triangular grid. Second, for more difficult applications in which mass conservation is essential, backward trajectory remapping can be replaced with forward trajectories method. Since in backward trajectories, reconstruction occur on the original grid, no mass violation happens. Finally, the algorithm can be coupled with an appropriate incompressible Navier–Stokes solver to model more physical flows in which a principle feature is the presence of an interface between two fluids with different fluid properties.

Acknowledgements

The authors would like to acknowledge Natural Sciences and Engineering Research Council of Canada for its financial support.

References

[1] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annu. Rev. Fluid Mech. 31 (1999) 567–603.
[2] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201.
[3] B.D. Nichols, C.W. Hirt, R.S. Hotchkiss, SOLA-VOF: A solution algorithm for transient fluid flow with multiple free boundaries, Tech. Rep. LA-8355, Los Alamos National Laboratory, Augest 1980, unpublished.
[4] M.D. Torrey, L.D. Cloutman, R.C. Mjolsness, C.W. Hirt, NASA-VOF2D: A computer program for incompressible flows with free surfaces, Tech. Rep. LA-10612MS, Los Alamos National Laboratory, December 1985, unpublished.
[5] M.D. Torrey, R.C. Mjolsness, R.L. Stein, NASA-VOF3D: A three-dimensional computer program for incompressible flows with free surfaces, Tech. Rep. LA-11009MS, Los Alamos National Laboratory, July 1987, unpublished.
[6] D.B. Kothe, R.C. Mjolsness, Ripple: A new model for incompressible flows with free surfaces, AIAA J. 30 (11) (1992) 2694.
[7] D.B. Kothe, R.C. Mjolsness, M.D. Torrey, Ripple: A computer program for incompressible flows with free surfaces, Tech. Rep. LA-12007-MS, Los Alamos National Laboratory, 1991.
[8] C.W. Hirt, Flow3D Users Manual, Flow Sciences, Inc., 1988.
[9] N.V. Deshpande, Fluid mechanics of bubble growth and collapse in a thermal ink-jet printhead, in: in SPSE/SPIES Electronic Imaging Devices ans System Symposium, January 1989.
[10] P.A. Torpey, Prevention of air ingestion in a thermal ink-jet device, in: Proceedings of the 4th International Congress on Advances in Non-Impact Print Technology, March 1988.
[11] H. Liu, E.J. Lavernia, R.H. Rangel, Numerical investigation of micropore formation during substrate impact of molten droplets in plasma spray process, Atomization Sprays 4 (1994) 2694.
[12] G. Trapaga, E.F. Matthys, J.J. Valencia, J. Szekely, Fluid flow, heat transfer, and solidification of molten deoplets impiging on substrates-comparison of numerical and experimental results, Metall. Trans. 23 (6) (1992) 701.
[13] D.L. Youngs, Time-dependent multi-material flow with large fluid distortion, in: K.W. Morton, M.J. Baines (Eds.), Numerical Methods for Fluid Dynamics, Academic Press, New York, 1982, p. 273.
[14] D.L. Youngs, An interface tracking method for a 3d Eulerian hydrodynamics code, Tech. Rep. 44/92/35, Los Alamos National Laboratory, AWRE, 1984.
[15] M. Bussmann, S. Chandra, J. Mostaghimi, On a three-dimensional volume tracking model of droplet impact, Phys. Fluids 11 (1999) 1406–1417.
[16] M. Bussmann, S. Chandra, J. Mostaghimi, Modeling the splashing of a droplet impacting a solid surface, Phys. Fluids 12 (2000) 3121–3132.
[17] M. Pasandideh-Fard, S. Chandra, J. Mostaghimi, A three-dimensional model of droplet impact and solidification, Int. J. Heat Mass Transfer 45 (2002) 2229–2242.

[18] J.E. Pilliod, E.G. Pucket, Second-order accurate volume-of fluid algorithm for tracking material interfaces, Tech. Rep. LBNL-40744, Lawrence Berkeley National Laboratory.
[19] D.B. Kothe, Perspective on Eulerian finite volume methods for incompressible interfacial flows, Tech. Rep. LA-UR-97-3559, Los Alamos National Laboratory.
[20] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, J. Comput. Phys. 141 (1998) 112–152.
[21] T.J. Barth, Aspects of unstructured grids and finite volume solvers for euler and navier-stokes equations, in: VKI/NASA/AGARD Special Courses on Unstrucured Grid methods for Advection Dominated Flows, AGARD Publications R-787, Los Alamos, NM, 1995.
[22] E.G. Pucket, A volume-of-fluid interface tracking algorithm with application to computing shock wave refraction, in: 4th International Synposium on Computational Fluid Dynamics, Davis, CA, 1991.
[23] G.H. Miller, E.G. Pucket, Edge effects on molybdenum-encapsulated molten silicate shock wave targets, J. Comput. Phys. 73 (3) (1994) 1426–1434.
[24] G.H. Miller, E.G. Pucket, A high-order godonov method for multiple condensed phases, J. Comput. Phys. 128 (1996) 134–164.
[25] E.G. Pucket, L.F. Henderson, P. Colella, A general theory of anomalous refraction, in: R. Brun, L.Z. Dumitrescu (Eds.), Shock Waves at Marsielles, Springer, Berlin, 1995, pp. 139–144.
[26] E.G. Pucket, L.F. Henderson, P. Colella, Computing surface tension with high-order kernels, in: K. Oshima (Ed.), Proceedings of the 6th International Symposium on Computational Fluid Dynamics, Lake Tahoe, CA, 1995, pp. 6–13.
[27] E.G. Pucket, A.S. Almgren, J.B. Bell, D.L. Marcus, W.J. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, J. Comput. Phys. 130 (1997) 269–282.
[28] D.K.S. Mosso, B. Swartz, C. Clancy, Recent enhancement of volume tracking algorithm for irregular grids, Tech. Rep. LA-cp-96-226, Los Alamos National Laboratory, Los Alamos, NM, 1996.
[29] D.K.S. Mosso, B. Swartz, R. Ferrel, A parallel volume-tracking algorithm for unstructured meshes, Tech. Rep. LA-UR-96-2420, Los Alamos National Laboratory, Los Alamos, NM, 1996.
[30] J.K. Dukowicz, J.R. Baumgardner, Incremental remapping as a transport/advection algorithm, J. Comput. Phys. 160 (2000) 318–335.
[31] C.W. Hirt, A.A. Amsden, J.L. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speed, J. Comput. Phys. 14 (1974) 227.
[32] L.G. Margolin, Introduction to an arbitrary Lagrangian–Eulerian computing method for all flow speed, J. Comput. Phys. 135 (1997) 198–202.
[33] R.M. Darlington, T.L. McAbee, G. Rodrigue, A study of ALE simulations of Rayleigh-Taylor instability, Comput. Phys. Commun. 135 (2001) 58–73.
[34] M. Jaeger, M. Carin, The Front-Tracking ALE method: application to a model of the freezing of cell suspensions, J. Comput. Phys. 179 (2002) 704–735.
[35] P. Knupp, L.G. Margolin, M. Shashkov, Reference jacobian optimization-based rezoning strategies for arbitrary Lagrangian Eulerian methods, J. Comput. Phys. 176 (2002) 93–128.
[36] J.E. Pilliod, An analysis of piecewise linear interface reconstruction algorithms for volume-of-fluid methods, Master's thesis, U.C Davis, September 1992.
[37] B.F.W. Press, S. Teukolsky, W. Vetterling, Numerical Recipes in C, Cambridge University Press, Cambridge, 1988.
[38] J. O'Rourke, Computational Geometry in C, Cambridge University Press, Cambridge, 1993.
[39] K. Schutte, An edge labeling approach to concave polygon clipping, in: ACM Transactions on Graphics (ftp://ftp.ph.tn.tudelft.nl/pub/klamer/clippoly.tar.gz), 1995.