Available online at www.sciencedirect.com

ScienceDirect

Computers & Fluids 36 (2007) 1199–1212

computers & fluids

www.elsevier.com/locate/compfluid

A numerical model for multiphase flow based on the GMPPS formulation. Part I: Kinematics

Frank Bierbrauer, Song-Ping Zhu *

School of Mathematics and Applied Statistics, University of Wollongong, NSW 2522, Australia

Received 30 September 2005; received in revised form 26 September 2006; accepted 3 December 2006
Available online 9 March 2007

Abstract

The CFD modeling of two-dimensional multiphase flows is a useful tool in industry, although accurate modeling itself remains a difficult task. One of the difficulties is to track the complicated topological deformations of the interfaces between different phases. This paper describes a marker-particle method designed to track fluid interfaces for fluid flows of at least three phases. The interface-tracking scheme presented in this paper is the first part of a series of papers presenting our complete model based on a one-field Godunov marker-particle projection scheme (GMPPS). In this part, we shall focus on the presentation of the interface-tracking scheme and the kinematic tests we conducted to examine the scheme's ability to accurately track interfacial movements typified by vorticity-induced stretching and tearing of the interface. Our test results show that for a set of carefully designed and commonly used error measures, relative percentage errors never exceed 2% for all of the tests and grid sizes considered, provided a sufficient number of marker particles are used. We shall also demonstrate that the method is of second-order accuracy and the interface transition width remains constant never exceeding three cell widths.

1. Introduction

Typically, physical interactions in the natural world as well as in industry demonstrate a need to study complex multiphase flow problems. For example: water-jet impingement cooling [1], the production of non-skid galvanised steel surfaces through water spray impact [2] or cavity filling in compression molding of polymeric resins [3] are typical problems in industry; the splash of water droplets [4] and the breaking of waves [5] are two in the natural world. A common feature of all of these seemingly quite-different problems is the presence of a free surface in the flow and the dynamical interaction of two or more immiscible fluids at their mutual interfaces.

While experimental studies provide valuable data and physical understanding of such multifluid systems they also present some difficulties. For example: probes can interfere with the flow and multifluid systems can be optically opaque [6], which make such flows notorious for their problems in setting up fully controlled physical experiments [7]. Thanks to major developments in both algorithm construction and computer power the simulation of, at least, two-phase flows, an aspect of computational fluid dynamics (CFD), is now almost routine.

Currently, it is the examination of very complex systems, where it is necessary to follow the evolution of a large range of scales for a long time, that presents the greatest challenge [7]. The goal of this and the following paper, Part II, is to construct an accurate, high fidelity (physically meaningful) and robust numerical model which is capable of solving multiphase flows. While the presented approach has great potential to be extended to three-dimensional cases, we shall only present, in this series of papers, the fundamental concept of the approach and focus on the discussion of some two-dimensional numerical test results. Extension to three-dimensions is currently being worked on and the results shall be presented in forthcoming papers upon their completion.

* Corresponding author. Tel.: +61 2 4221 3807; fax: +61 2 4221 4845.
E-mail addresses: frankb@uow.edu.au (F. Bierbrauer), spz@uow.edu.au (S.-P. Zhu).

0045-7930/$ - see front matter Crown Copyright © 2007 Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2006.12.001

In the case of multiple immiscible fluids, interfaces are present as material discontinuities in the physical properties of the fluids in contact. This is the classical sharp interface model of fluid mechanics where the interface is a surface of zero thickness. In contrast to sharp interface models, models which inherently describe both the bulk phases as well as the interface structure, are often called single domain or one-field models and do not require the direct application of boundary conditions at the interface. Rather, the interfacial conditions are built into the governing equations [8]. The one-field formulation provides a uniform description of multifluid flow so that the total fluid dynamical system is represented by a single mass and momentum conservation equation as well as a set of advection equations for each (scalar) component fluid phase present in the domain [8]. Material properties are recovered from a phase indicator function, usually the volume fraction. The one-field formulation is now commonly used for solving multiphase flow problems as it possesses certain advantages over the older two-fluid approaches [9] and is used in this paper.

Incompressible fluids are dominated by the solenoidality constraint, which must be obeyed both within the domain as well as at the boundaries for all time [10]. The greatest difficulty in numerically solving the unsteady incompressible Navier–Stokes (NS) equations, in primitive variable form, arises from the intimate coupling of the velocity and pressure [11]. One way to decouple the pressure from the velocity field is to use the momentum equations to construct a non-solenoidal intermediate velocity field which is then orthogonally projected onto the nearest div-free subspace. In this paper we make use of such an approximate projection method in combination with a high order Godunov discretisation of the non-linear term in the NS equations. A discussion of projection methods and a detailed description of the approximate projection method used in this series of papers is given in [12].

The ability to accurately 'track' an interface of arbitrary topology which is advected with the flow is crucial to any numerical model involving multiphase flows. Any such numerical method must be able to correctly predict the position of the interface while maintaining, as much as possible, its sharpness. This represents the geometry of the motion, or the interfacial *kinematics* [8] and will be the focus of this paper.

In general, the interface between two immiscible fluid phases is either explicitly tracked or implicitly captured. So-called *interface tracking* methods explicitly track the interface by retaining a position history of discrete points lying on the interface and solving a Lagrangian equation of motion [13]. On the other hand *interface capturing* methods 'track' the interface by assigning a scalar step or indicator function, often the volume fraction. As the fluid is immiscible and incompressible, the interface is passively advected with the flow [14] and a scalar transport equation is used to update the indicator function in an Eulerian manner.

Exact interface information is not contained in capturing methods; the interface location is identified through the indicator function, which is a field variable within the whole of the domain rather than just at the interface [13]. Although purely Lagrangian techniques allow the accurate tracking of the interface without smearing (numerical diffusion that causes the interface to become progressively wider), the method becomes too expensive in calculation time and requires constant remeshing when the interface undergoes large deformation [8].

On the other hand Eulerian methods allow large interfacial deformations to take place while maintaining accuracy and are easily extended to deal with multiple interfaces [15]. Their main difficulty lies in locating the interface accurately. Examples include the volume-of-fluid (VOF) method [16] and its successors [17–19]. Although VOF methods are topologically robust, possess excellent mass conservation and are easy to implement, they are prone to 'numerical surface tension', which, like physical surface tension, smoothes high curvature regions. However, unlike real physical surface tension, the smoothing of high curvature regions due to 'numerical surface tension' is artificial and is a consequence of the numerical approximation of the interface geometry coupled with volume conservation [13]. Therefore, it numerically disperses or merges fluid chunks [20].

An early Eulerian method is the marker-and-cell (MAC) method where massless marker particles are used to identify or 'mark' an individual fluid, usually by the volume fraction [21]. Although the method can deal with any number of interacting fluids and deforming interfaces the interface may be smeared and particles may accumulate in portions of the grid hindering resolution in other regions [15]. The use of particles in conjunction with a grid such as the MAC method has been extended to so-called particle-in-cell (PIC) methods [22]. An underlying Eulerian grid is used to compute field variables, such as pressure and velocity, while material information, such as mass and volume, is transported from cell to cell by the particles. The PIC method is able to deal with colliding interfaces which undergo large distortions although a large amount of numerical diffusion may arise. As well, numerical noise arises because a finite number of particles must be used [23]. In addition, multistreaming can result when two particles at the same point possess two different velocities [24] and the ringing instability is caused when particles possess more degrees of freedom than is present in the grid [25]. The ability of particle-mesh methods to separate the treatment of flow variables, such as pressure and velocity, within an Eulerian grid while advecting fluid identity in a Lagrangian manner, prompted Rider and Kothe [26] to improve on the older MAC method and make use of the advantages of PIC. In the marker-particle (MP) method [27] massless particles act as markers of fluid identity or the 'colour' as in the MAC method.

As in PIC the particles carry a position within the domain and are assigned an interpolated particle velocity from the grid values thereby avoiding multistreaming. The domain is filled with particles, each carrying its own colour which, after having been transported to its new position, is used in conjunction with surrounding particles in a weighted average to reconstruct the *grid* colour at each time step [23]. Colour information is transferred one way only from the particles to the grid so that numerical diffusion is almost eliminated other than interpolation errors caused by the transfer. A relatively high number of particles must be used so that numerical noise is kept to a minimum. Higher order interpolation functions are used to lessen problems due to the ringing instability. Unlike earlier VOF methods there is no diffusion or dispersion to destroy interface information so that the method is not subject to numerical surface tension [27].

While many natural processes, for example the splash of a raindrop, can be represented as two-phase flows, the computational modeling of industrial problems may require the accurate numerical solution of interfacial flow problems of more than two interacting fluid phases. One of the aims of this series of papers is to construct a multiphase flow algorithm which is more than capable of dealing with typical industrial problems which have to accurately track the interfaces of at least three immiscible fluid phases. The marker-particle method is, at least potentially, ideally suited to track the motion of more than two phases. While the MP method has been used to model two-phase flows, its ability to accurately track at least three fluid phases has not been adequately studied. In addition, since the MP method has not been fully documented, a second goal of this paper is to clarify the construction of the method in some detail. Lastly, a kinematic study of the MP method, involving two-phase flows with imposed velocity fields which translate, rotate, stretch and eventually tear fluid bodies, forms the basis of an analysis of the tracking ability of the method and represents the third aim of this paper.

The combination of a sophisticated higher order Godunov (approximate) projection method with the marker-particle method gives this the name: Godunov marker-particle projection scheme (GMPPS). Part I, the current paper, describes the kinematics of the GMPPS whereas Part II [12] studies the dynamical aspects. The remaining paper is organized as follows. In Section 2, some relevant details of the marker-particle method are provided. In Section 3, some numerical test results are presented to demonstrate how the present approach handles large kinematic deformation of interfaces. Our conclusions follow in Section 4.

2. The marker-particle method

The marker-particle method is a Lagrangian scheme to track fluid colour. An individual fluid colour indicates a particular fluid, that is, one 'phase' of the multiphase flow problem. This is the phase indicator function, designated C. Although marker particles act as markers of the presence of a particular fluid they do not explicitly track the interface. They are not assigned to lie along a particular interface and so only track it implicitly. However, since the particles carry fluid identity at scales smaller than the computational grid they strongly maintain individual fluid identity for all time.

For multiphase flow involving m fluids, m sets of marker particles are assigned within the domain, one set for each individual fluid. Each particle, in each set, is given an initial colour value of either one or zero depending on whether the fluid is present there or not. The velocity of each particle is assigned by interpolating surrounding grid velocity data at each time step. The particles are advected with these velocities by solving the equation of motion $\mathbf{u} = \mathrm{d}\mathbf{x}/\mathrm{d}t$ for one full time step. After advection is complete particle colour information is interpolated back to the grid to construct updated grid densities and viscosities which are used in the next time step of the NS solution algorithm.

2.1. The computational domain

The solution of the NS equations takes place in a domain defined by $\Omega = \{(x, y) : X_{\min} < x < X_{\max}, Y_{\min} < y < Y_{\max}\}$ with boundaries along the lines $x = X_{\min}$, $x = X_{\max}$, $y = Y_{\min}$ and $y = Y_{\max}$. The domain is discretised using a two-dimensional Eulerian grid made up of computational cells with centres (x_i, y_j), where

$$x_i = x_{\frac{1}{2}} + \left(i - \frac{1}{2}\right)\Delta x, \quad y_j = y_{\frac{1}{2}} + \left(j - \frac{1}{2}\right)\Delta y \tag{1}$$

as well as vertex nodes $(x_{i\pm 1/2}, y_{j\pm 1/2})$ and cell edges $(x_{i\pm 1/2}, y_j)$, $(x_i, y_{j\pm 1/2})$, where

$$x_{i-\frac{1}{2}} = x_{\frac{1}{2}} + (i-1)\Delta x, \quad y_{j-\frac{1}{2}} = y_{\frac{1}{2}} + (j-1)\Delta y \tag{2}$$

in a computational domain with integer counters $i = 1, 2, 3, \ldots, I$; $j = 1, 2, 3, \ldots, J$ such that the boundary lines $x = X_{\min}$, $x = X_{\max}$, $y = Y_{\min}$ and $y = Y_{\max}$ are defined along cell edges:

$$x_{\frac{1}{2}} = X_{\min}, \quad x_{I+\frac{1}{2}} = X_{\max}, \quad y_{\frac{1}{2}} = Y_{\min}, \quad y_{J+\frac{1}{2}} = Y_{\max} \tag{3}$$

and the space steps are given by

$$\Delta x = \frac{X_{\max} - X_{\min}}{I}, \quad \Delta y = \frac{Y_{\max} - Y_{\min}}{J}$$

2.2. Initialisation of particle position and colour

For each fluid phase a set of marker particles is required. Each marker particle (p) is initially located at position (x_p, y_p). Every marker particle of the mth set is assigned a colour $C_p^m = C^m(x_p, y_p)$, being an integer value of either one or zero, depending on whether that particle lies within the mth fluid or not, respectively. That is

$$C_p^m = \begin{cases} 1 & \text{if particle } p \text{ is located in fluid } m \\ 0 & \text{if particle } p \text{ is not located in fluid } m \end{cases} \tag{4}$$

This is shown diagrammatically for the case of three-phase flow in Fig. 1.

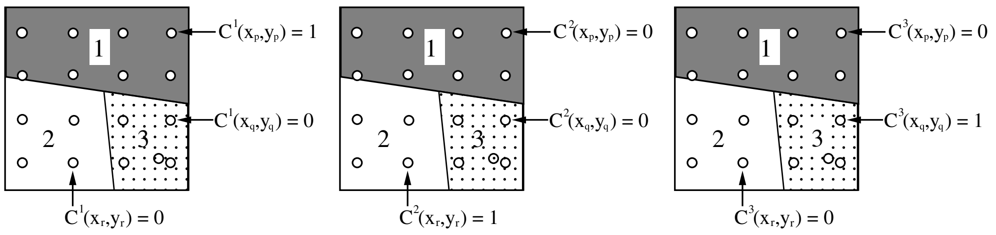

Fig. 1. The allocation of fluid colour $C^m(x_p, y_p)$ in a computational cell, containing three fluids, $m = 1, 2, 3$, requiring three sets of marker particles, one for each fluid involved.

Each particle now possesses a particular colour value which is permanently assigned for the entire duration of the calculation.

2.2.1. Initial particle positions

Particles are positioned in the entire domain by assigning a particular number within each computational cell. This can easily be done through assigning particle positions in a similar way as was done for cell centres. This ensures particles are regularly spaced within each cell. It is useful to have at least some particles aligned at cell edges so boundary data may be accurately tracked.

2.2.2. Initial particle colour

Initially, the particles are assigned a colour depending on whether they lie inside or outside a particular fluid, see Fig. 1. Each fluid is defined through some function of position, i.e. $f = f_m(x, y)$, for the mth fluid. For example, consider the initial condition of a droplet of fluid *one* traveling through another fluid, *two*, and eventually impacting a third fluid, *three*, shown in Fig. 2.

For the case of Fig. 2 a typical set of functions might be, for fluid one

$$f_1(x, y) = \{(x, y) : (x - x_0)^2 + (y - y_0)^2 \leqslant R^2\}$$

for a circle with centre located at (x_0, y_0) and of radius R. Similarly, fluid three could be defined by

$$f_3(x, y) = \{(x, y) : 0 \leqslant x \leqslant X, 0 \leqslant y \leqslant H\}$$

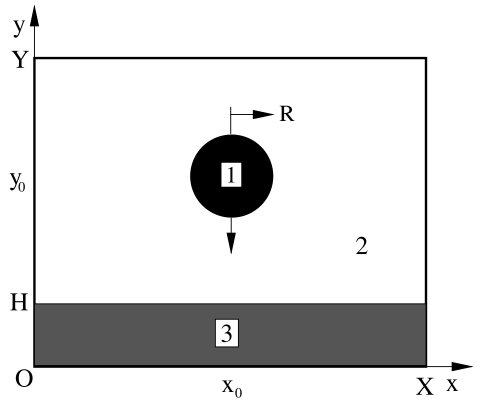

Fig. 2. The initial condition of a droplet (fluid 1)-layer (fluid 3) impact problem involving three-phase flow with ambient fluid 2.

for a layer of height H measured from the line $y = 0$. Particle colours may then be assigned as

$$C_p^1 = \begin{cases} 1 & \text{if } (x_p - x_0)^2 + (y_p - y_0)^2 \leqslant R^2 \\ 0 & \text{otherwise} \end{cases}$$

$$C_p^3 = \begin{cases} 1 & \text{if } y_p \leqslant H \\ 0 & \text{otherwise} \end{cases}$$

NB: Since at any position and time the particle colour is one for one of the fluids and zero for the other two, then $C_p^1 + C_p^2 + C_p^3 = 1$. So that in general for m fluids:

$$\sum_{k=1}^{m} C_p^k = 1 \Rightarrow C_p^m = 1 - \sum_{k=1}^{m-1} C_p^k$$

so that given m fluids only $m - 1$ are needed to fully define the initial particle colours over the whole domain.

2.3. Particle kinematics

As part of the solution algorithm of the NS equations grid velocity data is given both at cell centres at time t^n i.e. $u_{i,j}^n$ and $v_{i,j}^n$ and, through the Godunov algorithm, time centred at cell edges, i.e. $u_{i\pm1/2,j}^{n+1/2}$ and $v_{i,j\pm1/2}^{n+1/2}$. These values are used to interpolate velocity values from the grid to the particles.

2.3.1. Cell to particle velocity interpolation

Cell to particle interpolation is carried out with the use of a bilinear weighting function $S(x - x_i, y - y_j)$ [27] from the cell coordinates (x_i, y_j) to the point (x, y) by Eq. (14) of Appendix A.1:

$$S(x - x_i, y - y_j) = \begin{cases} \left(1 - \left|\frac{x - x_i}{\Delta x}\right|\right)\left(1 - \left|\frac{y - y_i}{\Delta y}\right|\right) \\ \quad \text{if } 0 \leqslant \left|\frac{x - x_i}{\Delta x}\right|, \left|\frac{y - y_i}{\Delta y}\right| \leqslant 1 \\ 0 \quad \text{otherwise} \end{cases} \tag{5}$$

this is equivalent to selecting those cell contributions lying within one cell width of the coordinate point (x, y). A particle will always be located within a region shared by four other grid cells, see Fig. 3, where the cell centres (x_i, y_j), (x_{i+1}, y_j), (x_i, y_{j+1}) and (x_{i+1}, y_{j+1}) all lie within the square with area $4\Delta x\Delta y$ surrounding the particle.

The cell to particle velocity interpolation $\mathbf{u}\left(\mathbf{x}_p^n\right)$ for a particle at $\left(x_p^n, y_p^n\right)$ at time t^n is then given by the expressions, using Eq. (13)

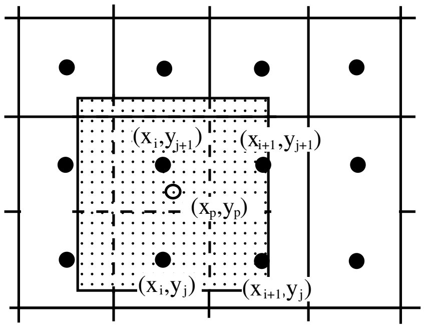

Fig. 3. Shaded area (square) surrounding the particle at position (x_p, y_p) and covering the grid cells with centres (x_i, y_j), (x_{i+1}, y_j), (x_i, y_{j+1}) and (x_{i+1}, y_{j+1}) used to interpolate velocity grid data to the particle.

$$
\begin{aligned}
u(x_p^n, y_p^n) &= \sum_{j=1}^{J}\sum_{i=1}^{I} S\left(x_p^n - x_i, y_p^n - y_j\right) u_{i,j}^n \\
v\left(x_p^n, y_p^n\right) &= \sum_{j=1}^{J}\sum_{i=1}^{I} S\left(x_p^n - x_i, y_p^n - y_j\right) v_{i,j}^n
\end{aligned} \tag{6}
$$

Similarly, the cell to particle velocity interpolations $\mathbf{u}\left(\mathbf{x}_p^{n+1/2}\right)$ for the time centred velocities are

$$
\begin{aligned}
u\left(x_p^{n+1/2}, y_p^{n+1/2}\right) &= \sum_{j=1}^{J}\sum_{i=1}^{I} S\left(x_p^{n+1/2} - x_i, y_p^{n+1/2} - y_j\right) u_{i,j}^{n+1/2} \\
v\left(x_p^{n+1/2}, y_p^{n+1/2}\right) &= \sum_{j=1}^{J}\sum_{i=1}^{I} S\left(x_p^{n+1/2} - x_i, y_p^{n+1/2} - y_j\right) v_{i,j}^{n+1/2}
\end{aligned} \tag{7}
$$

here the time centred velocities at cell centres were calculated from time centred cell edge velocities as $u_{i,j}^{n+1/2} = \left(u_{i-1/2,j}^{n+1/2} + u_{i+1/2,j}^{n+1/2}\right)\Big/2$ and $v_{i,j}^{n+1/2} = \left(v_{i,j-1/2}^{n+1/2} + v_{i,j+1/2}^{n+1/2}\right)\Big/2$.

2.3.2. Particle advection

Particles are advected over a full time step although, in order to make use of the grid velocities at the current and time centred values, a predictor corrector procedure [27] is used to solve $\mathbf{u} = \mathrm{d}\mathbf{x}/\mathrm{d}t$. That is

(1) *Predict*: to the new time centred particle positions $\left(x_p^{n+1/2}, y_p^{n+1/2}\right)$ by computing:

$$
x_p^{n+1/2} = x_p^n + \frac{\Delta t}{2} u\left(x_p^n, y_p^n\right), \quad y_p^{n+1/2} = y_p^n + \frac{\Delta t}{2} v\left(x_p^n, y_p^n\right) \tag{8}
$$

(2) *Correct*: to the new updated particle position $\left(x_p^{n+1}, y_p^{n+1}\right)$ by computing:

$$
\begin{aligned}
x_p^{n+1} &= x_p^n + \Delta t\, u\left(x_p^{n+1/2}, y_p^{n+1/2}\right), \\
y_p^{n+1} &= y_p^n + \Delta t\, v\left(x_p^{n+1/2}, y_p^{n+1/2}\right)
\end{aligned} \tag{9}
$$

2.3.3. Particle boundary conditions

It is possible that particles near the boundary at one time step may overshoot it in the next one. In that case particle position $\mathbf{x}_p$ could be one of four possible cases: $x_p > X_{\max}$, $x_p < X_{\min}$, $y_p > Y_{\max}$ and $y_p < Y_{\min}$ for each of the x and y particle coordinates.

(1) *No-slip conditions*: On approaching the boundary the fluid velocities there approach zero. The simplest way to impose this boundary condition is to reflect the particle back into the domain by the amount it has exceeded it.
(2) *Periodic conditions*: For periodic conditions the particle must exit the domain and appear out of the opposite face by the amount it exceeded the first boundary.

Table 1 shows the amount of particle overshoot and the remedy for both no-slip and periodic conditions.

These situations are shown in Fig. 4 for the case of a particle overshooting the x boundary at $x = X_{\max}$ and the y boundary at $y = Y_{\max}$ for (a) no-slip conditions and (b) periodic conditions.

2.4. Updating the colour function

As part of the solution of the NS equations in a multiphase flow problem the grid density and viscosity must

Table 1
Table of possible particle overshoot and reflection back into the domain

Overshoot	Reflection into domain	Periodic addition
$x_p > X_{\max}$	$x_p \to X_{\max} - (x_p - X_{\max})$	$x_p \to X_{\min} + (x_p - X_{\max})$
$x_p < X_{\min}$	$x_p \to X_{\min} + (X_{\min} - x_p)$	$x_p \to X_{\max} - (X_{\min} - x_p)$
$y_p > Y_{\max}$	$y_p \to Y_{\max} - (y_p - Y_{\max})$	$y_p \to Y_{\min} + (y_p - Y_{\max})$
$y_p < Y_{\min}$	$y_p \to Y_{\min} + (Y_{\min} - y_p)$	$y_p \to Y_{\max} - (Y_{\min} - y_p)$

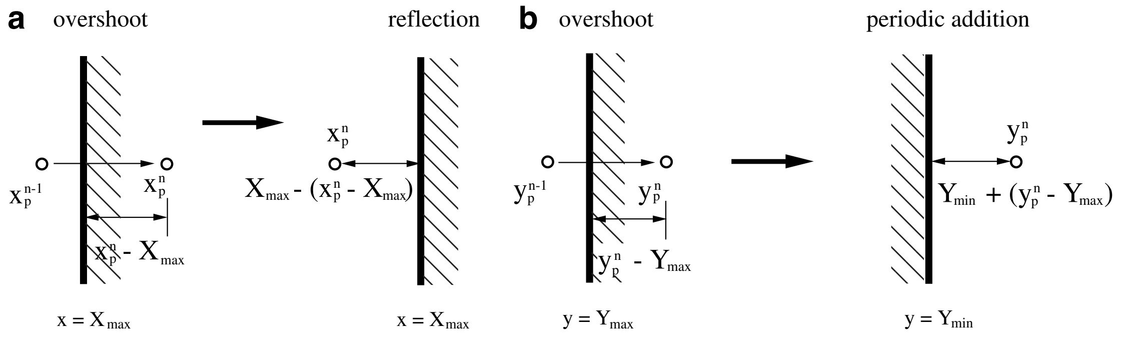

Fig. 4. Particle overshoot for the boundaries at (a) $x = X_{\max}$, no-slip boundaries and (b) $y = Y_{\max}$, periodic boundaries.

be calculated at each new time step. Updated grid densities and viscosities are calculated via Eq. (11) which require the updated grid volume fraction $C_{i,j}^{m,n+1}$ of the mth fluid. After each particle has been advected to its new position the fluid colour information C_p^m, which had been assigned to it initially, has to be interpolated back to the grid cell as the grid volume fraction. This may be done using the previously defined bilinear weighting function S so that the updated, cell centred, colour function for the mth fluid colour $C_{i,j}^{m,n+1}$ is given by

$$C_{i,j}^{m,n+1} = \frac{\sum_{p=1}^{\mathrm{TNP}} S\left(x_p^{n+1} - x_i, y_p^{n+1} - y_j\right) C_p^m}{\sum_{p=1}^{\mathrm{TNP}} S\left(x_p^{n+1} - x_i, y_p^{n+1} - y_j\right)} \tag{10}$$

where the sum is over the total number of particles (TNP) in the domain. This is an averaging of particle information to the cell centre with each particle having a weight in a grid cell of $S(x_p - x_i, y_p - y_j)$ [27]. This process is shown in Fig. 5.

Although initial particle configurations are very simply ordered (equi-spaced) the dynamics of the flow imply that, over time, particle configurations will become disordered. This makes the sums in (10) essential for keeping up with the continuous deformation of the interface, especially when it has suffered large deformations due to the drastic movement of all tracking particles. For this reason, a simple summation over known nearby particles will not work. Either the summation is taken over the whole set of particles or a search algorithm needs to be designed and implemented to track down only those particles within range of the interpolation function so that the summation is conducted only over these particles. However, the additional computational effort involved in the search algorithm is not compensated for by the time saved using a partial sum and thus we have decided to adopt the former approach.

Note that each individual fluid phase m, represented as a particle p with colour C_p^m, is distinctly separate from other fluid colours and as such do not explicitly interact. It is possible that some particles of different colour may overlap each other at a fluid interface through numerical error.

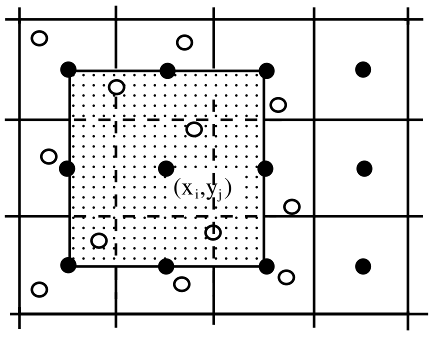

Fig. 5. Only those particles (open circles) within the shaded area (square), surrounding the cell centred at (x_i, y_j), are used to interpolate particle colour data to the cell centre.

As the summation in Eq. (10) is carried out over individual particles this does not present a problem because only those particles that belong to the same fluid will be summed over.

One should notice the difference between Eqs. (6) and (10) as the interpolation function in the latter has been normalised. The reason for this normalisation process is that the interpolation functions derived for the grid-to-particle interpolation cannot be directly applied to the interpolation process associated with particles-to-grid as these functions were derived based on a fixed number of grid points, see Appendix A.1. For the particle-to-grid interpolation process, the number of particles involved may now be other than four and thus these weight functions, developed for the process of interpolating the particle velocity field, must be weighted properly again. Strictly speaking, a new interpolation function, which depends on the number of particles present, needs to be used for the particle-to-grid interpolation. However, this would be computationally too expensive because the number of weight functions would vary from one grid point to another, and the calculation updating these weight functions at each grid point implies a considerable increase of computational effort.

For three-phase flow:

$$C_{i,j}^{1,n+1} = \frac{\sum_{p=1}^{\mathrm{TNP}} S\left(x_p^{n+1} - x_i, y_p^{n+1} - y_j\right) C_p^1}{\sum_{p=1}^{\mathrm{TNP}} S\left(x_p^{n+1} - x_i, y_p^{n+1} - y_j\right)},$$

$$C_{i,j}^{3,n+1} = \frac{\sum_{p=1}^{\mathrm{TNP}} S\left(x_p^{n+1} - x_i, y_p^{n+1} - y_j\right) C_p^3}{\sum_{p=1}^{\mathrm{TNP}} S\left(x_p^{n+1} - x_i, y_p^{n+1} - y_j\right)}$$

and $C_{i,j}^{2,n+1}$ may be updated as $C_{i,j}^{2,n+1} = 1 - C_{i,j}^{1,n+1} - C_{i,j}^{3,n+1}$. Using these values, which represent the cell volume fractions occupied by each fluid, the cell density and viscosity may be updated as

$$\begin{aligned} \rho_{i,j}^{n+1} &= C_{i,j}^{1,n+1}\rho_1 + \left(1 - C_{i,j}^{1,n+1} - C_{i,j}^{3,n+1}\right)\rho_2 + C_{i,j}^{3,n+1}\rho_3 \\ \mu_{i,j}^{n+1} &= \left(\frac{C_{i,j}^{1,n+1}}{\mu_1} + \frac{\left(1 - C_{i,j}^{1,n+1} - C_{i,j}^{3,n+1}\right)}{\mu_2} + \frac{C_{i,j}^{3,n+1}}{\mu_3}\right)^{-1} \end{aligned} \tag{11}$$

where the ρ_i's and μ_i's are the constant densities and viscosities in each fluid away from the interfaces.

2.5. The MP algorithm

The marker-particle algorithm reads as follows:

(1) *Initialisation at $t = 0$*:
  (a) Assign a set number of particles per cell, with a total number, TNP, in the whole domain.
  (b) Assign an initial particle colour C_p^m (Eq. (4)) for each fluid m using analytical functions $f_m(x, y)$ defining the mth initial fluid region.

(c) Use the particle colour data C_p^m just obtained to construct the grid cell initial colour data $C_{i,j}^{m,0}$ from Eq. (10) for each fluid m. Then initialise the density and viscosity as

$$\rho_{i,j}^0 = C_{i,j}^{1,0}\rho_1 + \left(1 - C_{i,j}^{1,0} - C_{i,j}^{3,0}\right)\rho_2 + C_{i,j}^{3,0}\rho_3$$

$$\mu_{i,j}^0 = \left(\frac{C_{i,j}^{1,0}}{\mu_1} + \frac{(1 - C_{i,j}^{1,0} - C_{i,j}^{3,0})}{\mu_2} + \frac{C_{i,j}^{3,0}}{\mu_3}\right)^{-1}$$

(2) *For time step* t^n, $n \geqslant 0$:

(a) Given the initial cell centred and time centred edge velocities: $u_{i,j}^n$, $v_{i,j}^n$, $u_{i\pm 1/2,j}^{n+1/2}$ and $v_{i,j\pm 1/2}^{n+1/2}$ interpolate velocities to all particles, i.e. the values $u\left(x_p^n, y_p^n\right)$, $v\left(x_p^n, y_p^n\right)$, $u\left(x_p^{n+1/2}, y_p^{n+1/2}\right)$ and $v\left(x_p^{n+1/2}, y_p^{n+1/2}\right)$, using Eqs. (6) and (7).

(b) Solve the equation of motion, using Eqs. (8) and (9):

$$x_p^{n+1/2} = x_p^n + \frac{\Delta t}{2} u\left(x_p^n, y_p^n\right),$$
$$y_p^{n+1/2} = y_p^n + \frac{\Delta t}{2} v\left(x_p^n, y_p^n\right)$$
$$x_p^{n+1} = x_p^n + \Delta t\, u\left(x_p^{n+1/2}, y_p^{n+1/2}\right),$$
$$y_p^{n+1} = y_p^n + \Delta t\, v\left(x_p^{n+1/2}, y_p^{n+1/2}\right)$$

(c) Interpolate the new grid centred colour function $C_{i,j}^{m,n+1}$ from advected particle colours C_p^m at new particle positions $\left(x_p^{n+1}, y_p^{n+1}\right)$ using Eq. (10).

(d) Update density and viscosity using the new cell centred colour functions with Eq. (11):

$$\rho_{i,j}^{n+1} = C_{i,j}^{1,n+1}\rho_1 + \left(1 - C_{i,j}^{1,n+1} - C_{i,j}^{3,n+1}\right)\rho_2 + C_{i,j}^{3,n+1}\rho_3$$

$$\mu_{i,j}^{n+1} = \left(\frac{C_{i,j}^{1,n+1}}{\mu_1} + \frac{\left(1 - C_{i,j}^{1,n+1} - C_{i,j}^{3,n+1}\right)}{\mu_2} + \frac{C_{i,j}^{3,n+1}}{\mu_3}\right)^{-1}$$

(e) Store old time particle positions $\left(x_p^{n+1}, y_p^{n+1}\right)$ along with the associated particle colour C_p^m at each position and for each fluid. Increment the time step, $n \to n+1$, and go to step 2(a).

3. Benchmark testing and discussion

Rigorous test problems exist for methods such as continuum advection schemes, however, no such benchmark tests are as yet available to track the advection of discontinuities present, for example, at interfaces between two immiscible fluids [28].

Typically, the solenoidal velocity field arising from the solution of multiphase flow problems will advect the flow in various ways. The simplest of these is a velocity field which merely translates fluid elements. The second slightly more complicated velocity field is one which rotates fluid elements. These two kinds of velocity field, translation and solid body rotation, represent simple advection which does not significantly disrupt initial fluid integrity. It is at least a necessary condition that a robust interface-tracking algorithm is able to admit such simple advection tests without significant distortion or degradation of fluid interfaces. Even with such restrictive constraints on the velocity field it is important to note that the achievement of good translation and rotation of a step function on an Eulerian mesh is still a difficult task [29].

Once an interface-tracking scheme has passed these two fundamental tests more serious tests with flows near interfaces possessing strong vortical content are required. Stretching, shearing, fluid merging and break-up are all possible in realistic multiphase flows [29]. A second series of imposed velocity fields was used to study the ability of the method to withstand changes in interface topology. The first imposed a shearing flow which stretches fluid elements into a thin continuous filament that spirals towards the centre of a vortex. It was taken from the well-known vortex-in-a-box problem. The second severely deforms initial fluid integrity by trapping fluid within multiple separate vortices [26]. Both are representatives of interfacial flow in real physical systems such as the Rayleigh–Taylor and Kelvin–Helmholtz instabilities [28].

3.1. Advection tests

When an exact solution is not known one way in which quantitative error measures are possible is to time-reverse the flow after half of the calculation period so that it returns to its initial state at the end of the full calculation period. A comparison of the initial and final flows can then be made. For all of the tests used in this paper the flow field is time-reversed through a cosinusoidal time-dependence with period T.

The four test cases are summarised in Table 2 under the headings of simple advection: translation and rotation velocity fields, and topology change: shearing flow and vortex velocity fields, and were first used by Rider and Kothe [26].

The current problem is the one stated in [26]: a fluid cylinder of radius 0.15 is centred at the point $(0.5, 0.75)$ in a unit computational domain. All boundaries are periodic. Fluid one occupies the cylinder and fluid two the surroundings. Note that since this is a two-phase problem only one grid colour function is needed so that $C = C^1$ (the drop) and $1 - C = C^2$ (the surrounding fluid). Therefore we use the symbol C to indicate the volume fraction of the drop. Although these tests assess only two-phase flow they are sufficient to study the method's ability to track interfaces in multiphase flows since particle advection and particle-to-cell interpolation only ever deal with a single set of particles for each fluid.

In all of the test cases a time-reverse strategy is implemented where each velocity field is multiplied by $\cos(\pi t/T)$ producing a flow that reverses in time after a time $T/2$ and

Table 2
Velocity field specified for the four different test cases

Test type	Velocity field	The specified velocity field
Simple	Translation	$\mathbf{u}(x,y) = (1,0)$
Advection	Rotation	$\mathbf{u}(x,y) = (y - 1/2, -(x - 1/2))$
Topology	Shearing flow	$\mathbf{u}(x,y) = (-\sin^2 \pi x \sin 2\pi y, \sin^2 \pi y \sin 2\pi x)$
Change	Vortex	$\mathbf{u}(x,y) = (\sin 4\pi(x+1/2)\sin 4\pi(y+1/2), \cos 4\pi(x+1/2)\cos 4\pi(y+1/2))$

returns to the initial state at $t = T$. The initial, $t = 0$, and final, $t = T$, flow structure, e.g. the volume fraction, can then be compared. The difference between the two produces an absolute error whereas the ratio of absolute error to initial state gives a relative error measure- ment.

An inferior interface-tracking algorithm will not be able to pass these two final tests while a superior method should maintain certain physical constants such as total domain mass and drop volume. In addition, as a measure of numerical diffusion, the interface thickness should remain approximately constant. Finally, the method itself should maintain second-order accuracy in space, this is tested by studying error norms of the method using two progressively finer spatial grids.

3.2. Error measures

The final to initial relative errors are measured in the L^1 norm and are given by the ratio

$$\frac{\|C^N - C^0\|_1}{\|C^N\|_1} \times 100$$

where C^N is the final grid volume fraction after time-reversal (at $t^N = T = N\Delta t$) and C^0 the initial volume fraction.

Mass errors may involve a loss of mass from the total mass of the original system, that is, the mass of the drop and the mass of the surrounding fluid as compared to the time dependent total domain mass calculated as an integral of the drop volume fraction $C^n(x,y)$.

Drop fluid volume must of course also be conserved, that is, for an initial drop volume of πR^2 the time dependent drop volume is calculated by integrating the drop volume fraction over the whole domain.

The region over which the grid volume fraction changes from a value of one, in the drop fluid, to a value of zero, in the surrounding fluid is here called the transition width. This width should remain approximately constant no matter what flow field is applied. It is expected that this value should be approximately one to two grid spacings wide, i.e. $h - 2h$. This parameter is measured by comparing individual grid cells one of which (x,y) possess $C = 1$ and the second of which (X,Y) has $C = 0$. The distance between these two cells is then minimised provided it is greater than zero. That is, the width is given by

$$\min_{0\leqslant x,X;\,y,Y\leqslant 1} \sqrt{(X-x)^2 + (Y-y)^2} > 0$$

Note that since this calculation necessarily obtains the smallest minimum at any one time this can change significantly at each time step possibly undergoing discontinuous jumps.

The order of accuracy of the method is required to be of second order in space. By comparing the grid cell colour function at the end of the time-reversed calculation to the original fluid body for two different size grids an order of accuracy may be calculated. Firstly, calculate the numerically (n) obtained grid colour function, $C_h^{(n)}$, on a grid with $h = \Delta x = \Delta y$. Secondly, use the initial, 'exact' (e) solution evaluated on the same grid, $C_h^{(e)}$, to calculate the L^1 error norm, i.e. $E(h) = \|C_h^{(n)} - C_h^{(e)}\|_1$ on that grid. Then, repeat the procedure for a second, finer grid, i.e. $h/2$, obtaining $E(h/2)$. Finally, take $\log_2$ of the ratio of errors to obtain the order of accuracy [30], which is approximately

$$\log_2\left(\frac{E(h)}{E(h/2)}\right) \tag{12}$$

Note that these measures are defined on a given computational grid so that they make use of cell centred grid data not particle data.

3.3. Results

The performance of the MP method is analysed quantitatively for all four tests. Fig. 6 shows particle colour function plots for the shearing flow and vortex tests at $t = 3$ and $t = 1$, respectively showing how either field can induce a most severe disruption of the original fluid body. This represents the 'exact' solution since fluid elements, represented by particles, have been advected with the given analytic velocity field so that no numerical errors are introduced as far as the particle colour function is concerned. A quantitative assessment is made by comparing relative percentage errors, for all of the advection fields, for total domain mass and drop volume over the time of the calculation. The variation in transition width is also plotted as a function of time. In all cases a CFL number of one is used (based on the maximum velocity in the domain). Finally, an order of accuracy is determined from L^1 error norms.

3.3.1. Translation

Fig. 7a shows the variation in relative percentage error with time for the translation of the original drop, up to

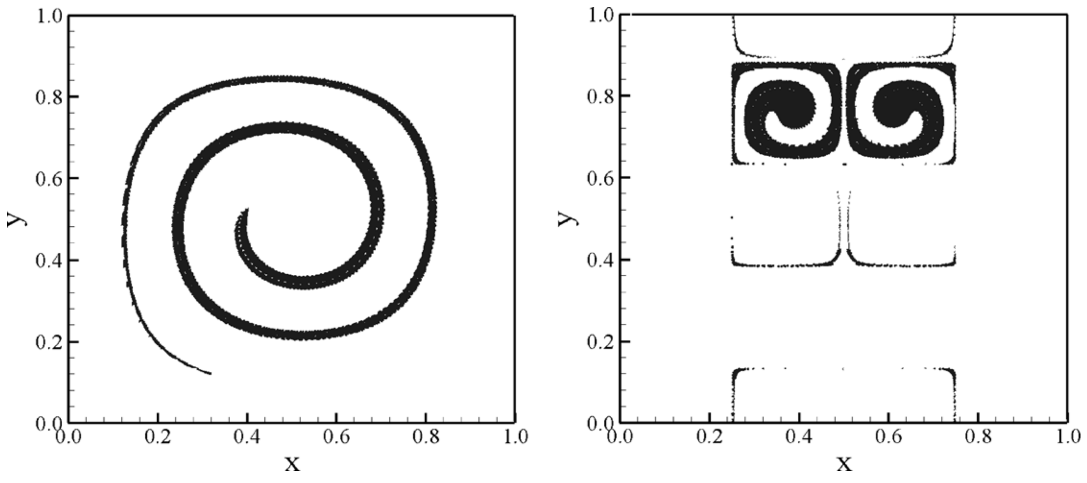

Fig. 6. Particle plots of the shearing flow and vortex fields at $t = 3$ and $t = 1$, respectively on a 128^2 grid using 16 particles per cell.

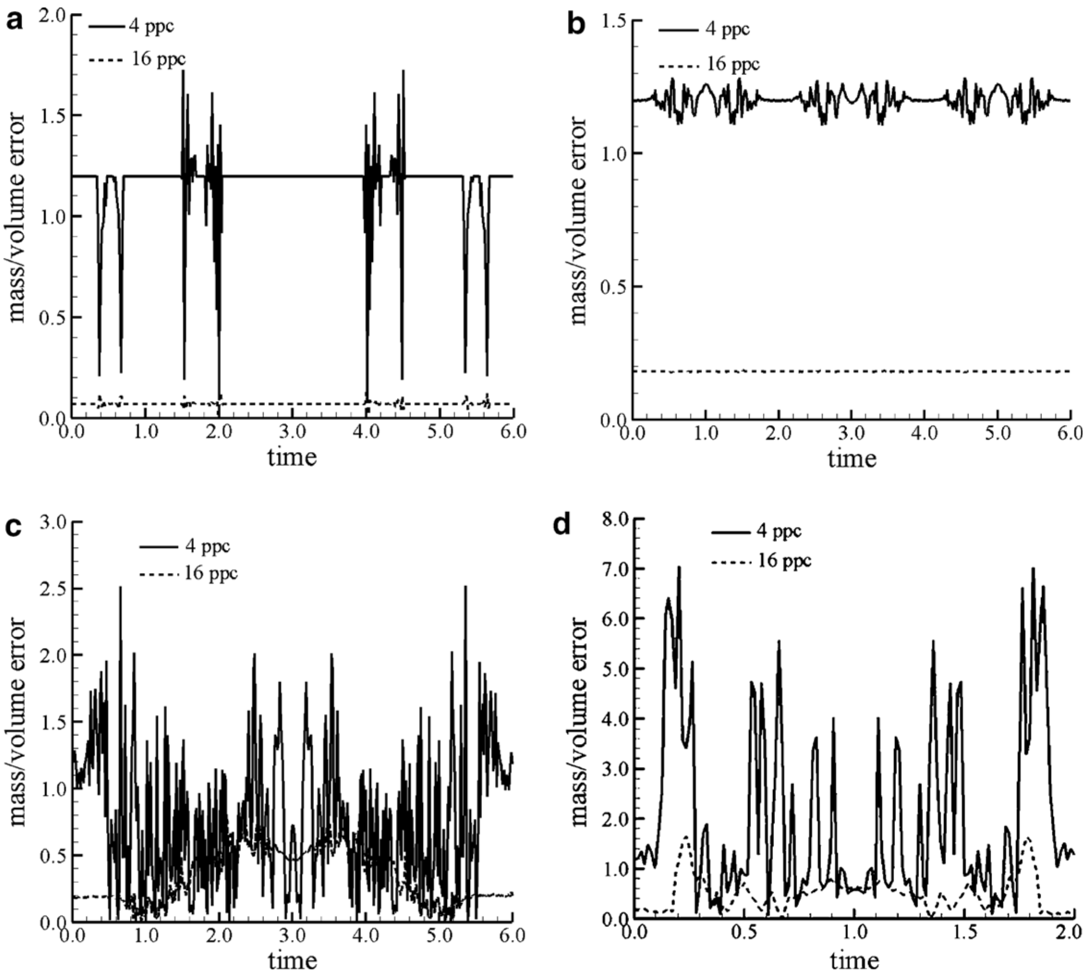

Fig. 7. Relative percentage errors for the domain mass and drop volume variation in time using a 64^2 grid with 4 (solid line) and 16 (dotted line) particles per cell for the (a) translation, (b) rotation, (c) shearing flow and (d) vortex velocity fields.

$t = 3$, and a reverse of the velocity field (i.e. with $T = 6$) to return the drop to its original position when $t = 6$. Note that the right hand edge of the drop leaves the domain at the right-hand-side boundary at about $t = 0.37$ and the left hand edge of the drop leaves the domain at about $t = 0.7$. Similarly, the drop, after having entered on the opposing face, leaves the domain again at about $t = 1.5$ with the left hand edge of the drop leaving and re-entering at about $t = 2.0$. The drop is stationary at $t = 3$ after which it reverses direction. Two sets of plots are shown for the cases of using 4 particles per cell (ppc) and 16 particles per cell, respectively. For the 4 ppc case the initial mass and volume error remains about 1.2%. On the other hand the results for the 16 ppc case show a drop in mass and volume errors down to 0.2%, which is very reasonable for the model-predicted results being used in engineering applications. The drop in mass error, by six times, has clearly indicated that the particle-to-cell interpolation is much more accurate when more particles are used. One should also notice in Fig. 7a that the mass errors are constant for most of the time, except some occasional fluctuations. These fluctuations are a result of the grid integral being less accurately evaluated when the drop is leaving and re-entering the computational domain. Clearly, when the number of particles per cell is increased to 16, the magnitude of the fluctuation has been substantially reduced.

Fig. 8a shows the transition width over time for the translating drop. The expected width is approximately $2h = 0.03125$ in the 64^2 grid. The 4 and 16 ppc cases both show a constant value of 0.031. Given that the translation flow field transports fluid elements exactly the transition width remains constant since all elements are translated equally.

3.3.2. Rotation

Fig. 7b shows the same error measures used for the translation case except that the original drop has now been rotated about the domain centre for a time of $t = 3$ after which it reverses its direction and returns to its starting position when $t = 6$ (using $T = 2$). Note that the drop never leaves the domain and the fluctuations in mass error are due to the number of particles used in the calculation. The 4 ppc case shows approximately constant mass and volume errors of 1.2% with the 16 ppc case dropping to 0.2%. This again shows a decrease in mass errors of six times. Note that whereas the 4 ppc case shows some oscillation about the mean value of the mass error the 16 ppc case possesses virtually no such variation. This is a direct product of an increase in the number of particles used in the particle-to-cell interpolation.

Fig. 8b shows the variation in transition width over time for the rotation test. Unlike the translation test there is some variation in transition width which jumps discontinuously. For the 4 ppc case (average width 0.026) these jumps are larger than for the 16 ppc case (average width 0.037). As the number of particles is increased the jumps are smaller in size.

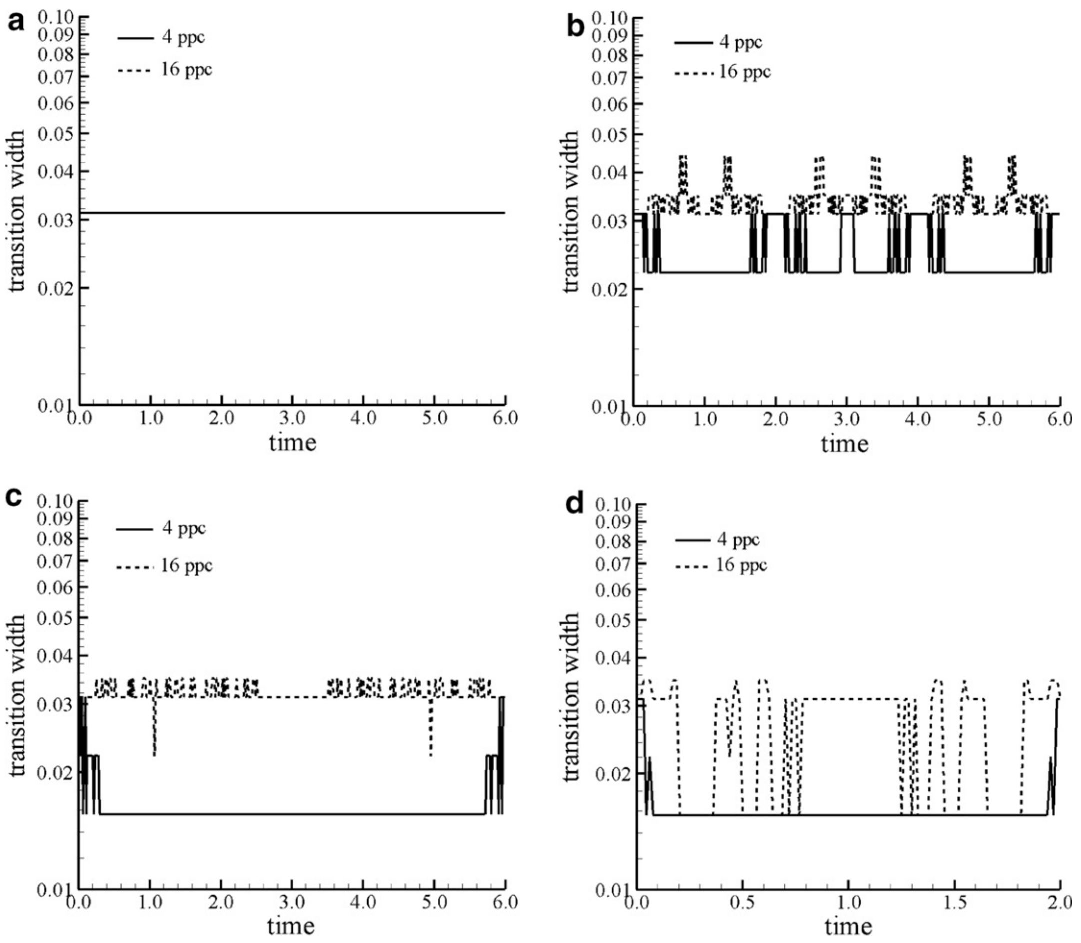

Fig. 8. Transition width variation in time using a 64^2 grid with 4 (solid line) and 16 (dotted line) particles per cell for the (a) translation, (b) rotation, (c) shearing flow and (d) vortex velocity fields.

3.3.3. Shearing flow

Fig. 7c shows quantitative errors of the time-reversed shearing flow solution over a period of $T = 6$. The 4 ppc case shows a large increase in mass and volume error as the initial drop is stretched out reaching a value as high as 2.5% but dropping to about 0.5% at the half way point. The graph is symmetric about this point retracing its course for the remaining time. On the other hand the 16 ppc case shows a much more even variation remaining approximately constant at about 0.5%. This happens when the severely stretched fluid elements, whose individual particles become widely separated when few are used, are no longer able to correctly represent the grid volume fractions. As the number of particles is increased this error is greatly decreased, by a factor of 5, to a satisfactory level of less than 1%.

Fig. 8c shows the transition width for the shearing flow test with some variation in width at the start and finish of the calculation time giving an average width of about 0.022 whereas the 16 ppc case produces 0.033. The variations are not as severe as the rotation test possibly due to the smoothness with which fluid elements are stretched.

3.3.4. Vortex test

For the time-reversed (with $T = 2$) vortex solution, Fig. 7d, the previous situation for the shearing flow test is exacerbated as the fluid elements are now not only stretched but torn. As a result the 4 ppc case shows errors as large as 7% with a reduction to about 1% for the 16 ppc case. The relatively small differences seen at $t = T/2$, in both the shearing flow ($t = 3$) and vortex ($t = 1$) mass errors is due to the zero velocity reached at this point as the flow is reversed.

The vortex test result of Fig. 8d shows significantly more variation in transition width than any of the previous cases. Remarkably the 4 and 16 ppc cases both show an almost constant width of about 0.025 (on average) although the 16 ppc case undergoes more variation.

3.3.5. Dependence of width on number of particles

All of the advection tests except the translation test (see Fig. 8) show that the average transition width for the 4 ppc case is smaller than the 16 ppc case. This could be accounted for by the likelihood that for the 4 ppc case only a small number of particles possessing $C_p = 1$ contribute to the grid volume fraction having $0 < C_{i,j} < 1$ whereas for the 16 ppc case even a single particle with $C_p = 1$ will induce a grid volume fraction greater than 0 so that there will exist more cells having an intermediate volume fraction (i.e. not 1 or 0). This will produce a wider transition width than for the 4 ppc case. The translation test does not show this difference because all particles are exactly translated by the same amount.

3.3.6. Time-reversed flow

Fig. 9 shows the relative percentage error, in the L^1 norm, of the grid volume fraction for the time-reversed

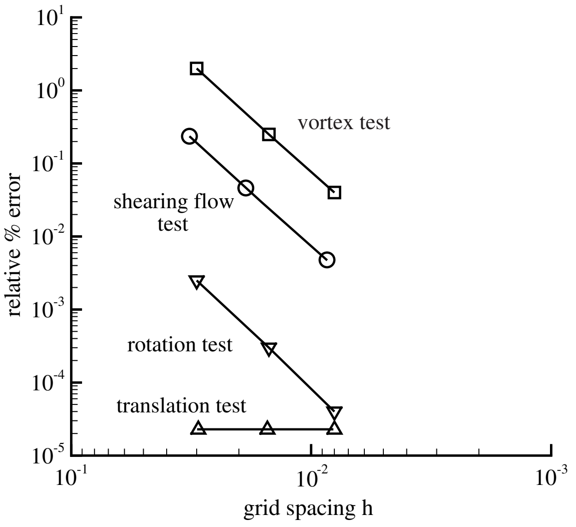

Fig. 9. Relative percentage errors in the L^1 norm comparing the time-reversed and initial grid volume fraction for the shearing flow and vortex tests as a function of grid size.

translation, rotation, shearing flow and vortex tests compared to the initial condition as a function of grid size ($h = 1/32, 1/64, 1/128$). Note that the graph also shows the convergence of the algorithm.

Note that for the translation test there is no improvement as the grid is refined or as the number of particles per cell is increased. This occurs since fluid elements are translated exactly so that errors at the limit of double precision arithmetic, used in the code, are seen. This cannot improve as the grid is refined. This is not the case for the remaining flow fields. As expected the rotation test has the lowest relative error of the remaining three tests. All remaining tests show the same log–log slope as the grid is refined. It is clear that for both the shearing flow and vortex tests the time-reversed relative percentage errors are small even for the coarsest grid shown, no larger than 2%. In addition, there is a logarithmic decrease in error as the grid spacing is decreased. It is also evident that the number of particles per cell used in the calculation does not significantly effect this outcome. This is true since the particles having been accurately advected resume their initial positions at the end of the cycle.

3.3.7. Order of accuracy

The order of accuracy for each test was calculated for the time-reversed flow fields up to $t = T$ for both the 4 and 16 particles per cell cases on $h = 1/32$, 1/64 and 1/128 grids. The time-reversal period T is chosen to coincide with that used in [28] to allow a comparison, especially their Tables 2 and 3. Note that as the time-reversal period increases the fluid body is allowed to evolve further away from its initial configuration, at $t = 0$, so that it needs to undergo more complicated topological change in order to reconstruct correctly at $t = T$.

For the case of simple translation with a constant velocity field the MP method time-reverses the flow so exactly that the error obtained is already at the double precision limit of 10^{-8} of the code for the coarsest grid used so that a comparison between grids is not possible with the method used in Eq. (12).

However, Tables 3–5 show the L^1 error norms comparing the time-reversed and initial flow for each grid and number of particles used as well as the order of accuracy in each case.

As expected the results of [28] for the rotation test, their Table 8, show second-order accuracy as do those of Table 3 although the present results show an accuracy approaching the limit of double precision arithmetic used in the code.

For the shearing flow and vortex tests a comparison is possible to the results of Rider and Kothe [28]. For the vortex test the periods $T = 0.5$, 2.0 and 8.0 were used identical to that in [28], their Table 2. Their paper compared the order of accuracy demonstrated in 32^2, 64^2 and 128^2 grids for the PLIC method. Whereas their results show mostly second-order accuracy. Table 4 shows consistently high second-order accuracy of the current model for both short

Table 3
The L^1 error norms and order of accuracy for the rotation test using the time-reversed flow field for both 4 and 16 particles per cell

Grid (4 ppc)	$E(h)$ ($T = 1.0$)	Order ($T = 1.0$)	$E(h)$ ($T = 2.0$)	Order ($T = 2.0$)	$E(h)$ ($T = 4.0$)	Order ($T = 4.0$)
32^2	2.31×10^{-7}		4.82×10^{-7}		9.85×10^{-7}	
		2.87		2.89		2.91
64^2	3.16×10^{-8}		6.46×10^{-8}		1.31×10^{-7}	
		2.75		2.76		2.78
128^2	4.71×10^{-9}		9.51×10^{-9}		1.91×10^{-8}	
Grid (16 ppc)						
32^2	2.33×10^{-7}		4.89×10^{-7}		9.98×10^{-7}	
		2.93		2.97		2.98
64^2	3.06×10^{-8}		6.25×10^{-8}		1.26×10^{-7}	
		2.78		2.79		2.79
128^2	4.46×10^{-9}		8.99×10^{-9}		1.81×10^{-8}	

Table 4
The L^1 error norms and order of accuracy for shearing flow using the time-reversed flow field for both 4 and 16 particles per cell

Grid (4 ppc)	$E(h)$ ($T = 0.5$)	Order ($T = 0.5$)	$E(h)$ ($T = 2.0$)	Order ($T = 2.0$)	$E(h)$ ($T = 8.0$)	Order ($T = 8.0$)
32^2	9.48×10^{-6}		4.19×10^{-5}		4.29×10^{-4}	
		2.77		2.86		2.93
64^2	1.38×10^{-6}		5.77×10^{-6}		5.63×10^{-5}	
		2.90		2.91		2.98
128^2	1.83×10^{-7}		7.69×10^{-7}		7.13×10^{-6}	
Grid (16 ppc)						
32^2	9.50×10^{-6}		4.02×10^{-5}		4.15×10^{-4}	
		2.83		2.86		2.94
64^2	1.33×10^{-6}		5.48×10^{-6}		5.42×10^{-5}	
		2.92		2.94		2.97
128^2	1.75×10^{-7}		7.09×10^{-7}		6.91×10^{-6}	

Table 5
The L^1 error norms and order of accuracy for the vortex test using the time-reversed flow field for both 4 and 16 particles per cell

Grid (4 ppc)	$E(h)$ ($T = 1.0$)	Order ($T = 1.0$)	$E(h)$ ($T = 2.0$)	Order ($T = 2.0$)	$E(h)$ ($T = 4.0$)	Order ($T = 4.0$)
32^2	4.51×10^{-4}		1.45×10^{-3}		5.22×10^{-3}	
		2.63		2.69		2.57
64^2	7.27×10^{-5}		2.25×10^{-4}		8.80×10^{-4}	
		2.89		2.92		2.67
128^2	9.79×10^{-6}		2.98×10^{-5}		1.38×10^{-4}	
Grid (16 ppc)						
32^2	4.59×10^{-4}		1.53×10^{-3}		5.43×10^{-3}	
		2.73		2.81		2.77
64^2	6.90×10^{-5}		2.18×10^{-4}		7.94×10^{-4}	
		2.88		2.94		2.93
128^2	9.33×10^{-6}		2.85×10^{-5}		1.04×10^{-4}	

and long periods. In addition, the individual error norms are better than theirs by almost two orders of magnitude.

Table 5 shows the error norms and order of accuracy data for the vortex test using time-reversal periods of $T = 1.0$, 2.0 and 4.0. This test is more severe than the shearing flow case and whereas the results of [28], their Table 3, show that their $T = 1.0$ case is second-order accurate the remaining results for $T = 2.0$ and 4.0 are only first-order accurate. However, Table 5 again demonstrates consistent second-order accuracy no matter the time-reversal period or the number of particles used in the calculation.

It is interesting to note that the error norms obtained from the rotation test are approximately of size 10^{-8} whereas the shearing flow test possesses an average error norm around 10^{-6} and the vortex test average errors of about 10^{-4}. It can be seen that as the test itself becomes more severe the errors increase by two orders of magnitude although strictly still maintaining second-order accuracy throughout.

As mentioned earlier the MP method does not strictly adhere to solenoidality. That is, the interpolation of an exactly divergence-free velocity field, discretised on the grid as $\mathbf{u}_{i,j}$, does not carry over the div-free property from the grid points to the field point (x, y). That is

$$\nabla \cdot \mathbf{u}(x,y) = \nabla \cdot \sum_{j=1}^{J} \sum_{i=1}^{I} S(x - x_i, y - y_j)\mathbf{u}_{i,j} \neq 0$$

As particle velocities are required to advect fluid colour information and these are interpolated from grid velocities this implies that particles are given slightly incorrect velocity data. This is not a problem for the simple advection tests such as translation and rotation but in the presence of vorticity, i.e. for the shearing flow and vortex tests, the particles may have velocities imposed on them which are poor representations of the actual field variables [27]. Although this may seem to be a major shortcoming of the method, in practice it has not been found to be a problem especially when a fine grid is used. The results of Figs. 7 and 9 show that mass is conserved up to 1% relative error even for severe deformation and that these errors do not appear to accumulate as fluid bodies are sheared due to strong vorticity.

4. Conclusions

In this paper a marker-particle method is outlined with the potential to track an arbitrary number of interacting fluid phases. The paper concentrates on the ability of the method to accurately track fluid interfaces for two-phase flows by imposing solenoidal velocity fields which translated, rotated, stretched and deformed fluid bodies. In order to perform error measurements based on an exact solution the individual flows were time-reversed so that fluid velocities reversed in sign returning the flow to its initial position after a period T. This allowed the known initial condition to be compared with the time-reversed result so that typical error measures such as global grid mass and volume conservation as well as interface transition width and spatial order of accuracy could be computed.

The results have shown that, provided a sufficient number of particles are used, the method is able to accurately track fluid interfaces with relative percentage errors no greater than 1–2%. This remains true for flows which translate, rotate, stretch and severely deform fluid bodies in the presence of strong vorticity for all of the grid sizes tested. In addition, the width of fluid interfaces, as demonstrated by grid volume fractions, does not increase. Interface width is constant at approximately two grid cell lengths for the translation test while never exceeding three cell widths for any of the other tests. It is also clear that the method is consistently of second-order accuracy for all of the flow tests studied no matter how much the fluid bodies were deformed. This compares more than favourably to the well-known PLIC method currently in common use.

While some doubts remain concerning the method's ability to accurately conserve mass the results have shown that this does not appear to be a severe problem for the tests so far conducted. Nonetheless, the construction of interpolation functions which strictly adhere to solenoidality would ensure a superior method to the one presented here. Although the MP method performs better than most purely grid-based methods it does incur a cost in the number (and storage) of particles required to ensure good performance.

Appendix A

A.1. Interpolation weighting function

Given a computational grid, defined earlier by Eq. (1) and functional values $X_{i,j}$ at each cell centre, we wish to be able to interpolate data from these known values to any point (x, y) lying in the domain, not necessarily on a cell centre, vertex or edge.

Now, construct the interpolate $X(x, y)$ which is linear in x and y, i.e.:

$$X(x,y) = a_0 + a_1x + a_2y + a_3xy$$

with the a_i's real constants. This value will lie somewhere within a region which has as its vertices the grid cell values $X_{i,j} = X(x_i, y_j)$, $X_{i+1,j} = X(x_{i+1}, y_j)$, $X_{i,j+1} = X(x_i, y_{j+1})$ and $X_{i+1,j+1} = X(x_{i+1}, y_{j+1})$, as shown in Fig. 10. Now the data at each cell centre is known so that we may write down four equations in the four unknowns a_0, a_1, a_2, a_3:

$$X_{i,j} = a_0 + a_1x_i + a_2y_j + a_3x_iy_j$$
$$X_{i+1,j} = a_0 + a_1x_{i+1} + a_2y_j + a_3x_{i+1}y_j$$
$$X_{i,j+1} = a_0 + a_1x_i + a_2y_{j+1} + a_3x_iy_{j+1}$$
$$X_{i+1,j+1} = a_0 + a_1x_{i+1} + a_2y_{j+1} + a_3x_{i+1}y_{j+1}$$

construct the matrix equation with solution vector $(a_0, a_1, a_2, a_3)^{\mathrm{T}}$:

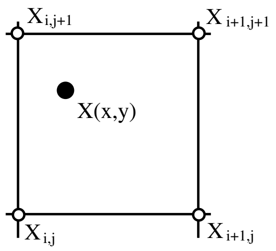

Fig. 10. Interpolation of a function value $X = X(x,y)$ at the point (x,y) using the functional data located at cell centres $X_{i,j}$, $X_{i+1,j}$, $X_{i,j+1}$ and $X_{i+1,j+1}$.

$$\begin{pmatrix} 1 & x_i & y_j & x_iy_j \\ 1 & x_{i+1} & y_j & x_{i+1}y_j \\ 1 & x_i & y_{j+1} & x_iy_{j+1} \\ 1 & x_{i+1} & y_{j+1} & x_{i+1}y_{j+1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} X_{i,j} \\ X_{i+1,j} \\ X_{i,j+1} \\ X_{i+1,j+1} \end{pmatrix}$$

solving for the a_i's we get:

$$\begin{aligned} X(x,y) = {} & \left[1 - \left(\frac{x - x_i}{\Delta x}\right)\right]\left[1 - \left(\frac{y - y_j}{\Delta y}\right)\right]X_{i,j} \\ & + \left(\frac{x - x_i}{\Delta x}\right)\left(\frac{y - y_j}{\Delta y}\right)X_{i+1,j+1} \\ & + \left(\frac{x - x_i}{\Delta x}\right)\left[1 - \left(\frac{y - y_j}{\Delta y}\right)\right]X_{i+1,j} \\ & + \left[1 - \left(\frac{x - x_i}{\Delta x}\right)\right]\left(\frac{y - y_j}{\Delta y}\right)X_{i,j+1} \end{aligned} \tag{13}$$

this may be written using a weighting function S as

$$X(x,y) = \sum_{j=1}^{J}\sum_{i=1}^{I} S(x - x_i, y - y_j)X_{i,j}$$

where S is restricted so that only the four cell vertices above are selected, i.e.:

$$S = \begin{cases} \left(1 - \left|\frac{x-x_i}{\Delta x}\right|\right)\left(1 - \left|\frac{y-y_j}{\Delta y}\right|\right) & \text{if } 0 \leqslant \left|\frac{x-x_i}{\Delta x}\right|, \left|\frac{y-y_j}{\Delta y}\right| \leqslant 1 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

References

[1] Bierbrauer F, Soh WK, Yuen WYD. Application of the Eulerian–Lagrangian method to the water-jet cooling of a hot moving strip. In: Proceedings of the ICHMT international symposium on advances in computational heat transfer, Palm Cove, Queensland, Australia, 20–25 May 2001.

[2] Bierbrauer F. Mathematical modeling of water-droplet impact on hot galvanised steel surfaces. PhD thesis, University of Wollongong, Australia, 2004.

[3] Kim MS, Park JS, Lee WL. A new VOF-based numerical scheme for the simulation of fluid flow with free surface. Part II: Application to the cavity filling and sloshing problems. Int J Numer Meth Fluids 2003;42:791–812.

[4] Liow J-L. Splash formation by water drops. In: Rein M, editor. Drop-surface interactions. New York: Springer; 2002. p. 299–302.

[5] Chen G, Kharif C, Zaleski S, Li J. Two dimensional simulations of breaking waves. Phys Fluids 1999;11:121–33.

[6] Hanratty TJ, Theofanus T, Delhaye J-M, Eaton J, McLaughlin J, Prosperetti A. Workshop findings. Int J Multiphase Flow 2003;29:1047–59.

[7] Prosperetti A, Tryggvason G. Appendix 3: Report of study group on computational physics. Int J Multiphase Flow 2003;29:1089–99.

[8] Kothe DB. Perspectives on Eulerian finite volume methods for incompressible interfacial flows. In: Kuhlmann HC, Rath H-J, editors. Free surface flows. New York: Springer; 1999. p. 267–331.

[9] Lakehal D, Meier M, Fulgosi M. Interface tracking towards the direct simulation of heat and mass transfer in multiphase flows. Int J Heat Fluid Flow 2002;23:242–57.

[10] Gresho PM. Incompressible fluid dynamics: some fundamental formulation issues. Ann Rev Fluid Mech 1991;23:413–53.

[11] Liu M, Ren Y-X, Zhang H. A class of fully second order accurate projection methods for solving the incompressible Navier–Stokes equations. J Comput Phys 2004;200:325–46.

[12] Bierbrauer F, Zhu S-P. A numerical model for multiphase flow based on the GMPPS formulation. Part II: Dynamics. Comput Fluids, submitted for publication.

[13] Kothe DB, Rider WJ. Practical considerations for interface tracking methods. In: Proceedings of the 6th international symposium on computational fluid dynamics, Lake Tahoe, CA, 4–8 September 1995.

[14] Scardovelli R, Zaleski S. Interface reconstruction with least-square fit and split Eulerian–Lagrangian advection. Int J Numer Meth Fluids 2003;41:251–74.

[15] Floryan JM, Rasmussen H. Numerical methods for viscous flows with moving boundaries. Appl Mech Rev 1989;42:323–41.

[16] Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. J Comput Phys 1981;39:201–25.

[17] Noh WF, Woodward P. SLIC (simple-line-interface-calculation). In: van Dooren AI, Zandbergen PJ, editors. Lecture notes in physics, vol. 59. New York: Springer; 1976. p. 330–40.

[18] Youngs DL. Time-dependent multi-material flow with large fluid distortion. In: Morton KW, Baines MJ, editors. Numerical methods for fluid dynamics. New York: Academic; 1982. p. 273–85.

[19] Pilliod JE, Puckett EG. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. J Comput Phys 2004;199:465–502.

[20] Černe G, Petelin S, Tiselj I. Numerical errors of the volume-of-fluid interface tracking algorithm. Int J Numer Meth Fluids 2002;38:329–50.

[21] Welch JE, Harlow FW, Shannon JP, Daly BJ. The MAC method: a computing technique for solving viscous, incompressible, transient fluid flow problems involving free surfaces. Los Alamos Scientific Laboratory Report LA-3425, 1966.

[22] Harlow FH. PIC and its progeny. Comp Phys Comm 1988;48:1–11.

[23] Rider WJ, Kothe DB, Mosso SJ, Cerutti JH, Hochstein JI. Accurate solution algorithms for incompressible multiphase flows. In: 33rd Aerospace sciences meeting, Reno, NV, 1995, AIAA paper no. 95-0699.

[24] Brackbill JU, Ruppel HM. FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. J Comput Phys 1986;65:314–43.

[25] Brackbill JU. The ringing instability in particles-in-cell calculations of low speed flow. J Comput Phys 1988;75:469–84.

[26] Rider WJ, Kothe DB. Stretching and tearing interface tracking methods. In: 12th AIAA CFD conference, June 20th, San Diego, CA, 1995, AIAA paper no. 95-1717.

[27] Rider WJ, Kothe DB. A marker particle method for interface tracking. In: Proceedings of the 6th international symposium on CFD, Lake Tahoe, CA, 4–8 September 1995.

[28] Rider WJ, Kothe DB. Reconstructing volume tracking. J Comput Phys 1998;141:112–52.

[29] Rudman M. Volume-tracking methods for interfacial flow calculations. Int J Numer Meth Fluids 1997;24:671–91.

[30] LeVeque R. Finite difference methods for differential equations, AMath 585-6 Notes, University of Washington, 1998.