Available online at www.sciencedirect.com
SCIENCE DIRECT
Journal of Computational Physics 214 (2006) 858–877
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

An auxiliary grid method for computations of multiphase flows in complex geometries

Metin Muradoglu *, Arif Doruk Kayaalp

Department of Mechanical Engineering, Koc University, Rumelifeneri Yolu, Sariyer 34450, Istanbul, Turkey

Received 23 February 2005; received in revised form 19 October 2005; accepted 20 October 2005
Available online 15 December 2005

Abstract

A method is developed for computations of interfacial flows in complex geometries. The method combines a front-tracking method with a newly developed finite volume (FV) scheme and utilizes an auxiliary grid for computationally efficient tracking of interfaces in body-fitted curvilinear grids. The tracking algorithm reduces particle tracking in a curvilinear grid to tracking on a uniform Cartesian grid with a look up table. The algorithm is general and can be used for other applications where Lagrangian particles have to be tracked in curvilinear or unstructured grids. The spatial and temporal errors are examined and it is shown that the method is globally second order accurate both in time and space. The method is implemented to solve two-dimensional (planar or axisymmetric) interfacial flows and is validated for a buoyancy-driven drops in a straight tube and the motion of buoyancy-driven drops in a periodically constricted channel.

Keywords: An auxiliary grid method; Particle tracking; A finite-volume/front-tracking method; Interfacial flows; Complex geometries

1. Introduction

Simulation of multiphase flows is notoriously difficult mainly due to the presence of deforming phase boundaries. A variety of numerical methods have been developed and successfully applied to a wide range of multifluid and multiphase flow problems [16,20,21,23,26]. In spite of this success, significant progress is still needed especially for accurate computations of multiphase flows involving strong interactions with complex solid boundaries. It is of great importance to be able to accurately model strong interactions between bubbles/drops and curved solid boundaries in many engineering and scientific applications such as microfluidic systems [22], pore-scale multi-phase flow processes [13,14] and biological systems [7,19]. It has been recently shown that adaptive grid methods combined with a level-set approach can be successfully used to solve interfacial flow problems in complex geometries [1,6,28]. Here a front-tracking approach is taken to account for the effects of the interfacial tension and change in material properties in different phases.

* Corresponding author. Tel.: +90 607 254 5402; fax: +90 607 255 1222/212 338 1548.
E-mail address: mmuradoglu@ku.edu.tr (M. Muradoglu).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2005.10.024

The front-tracking method developed by Unverdi and Tryggvason [25] has proved to be an effective tool for computations of interfacial flows and it has been successfully applied to a wide range of multiphase flow problems but almost all in relatively simple geometries [23] except for the cut-cell Cartesian method developed by Udaykumar et al. [24]. The front-tracking method has many advantages such as its conceptual simplicity and small numerical diffusion. However, its main disadvantage is probably the difficulty to maintain the communication between the Lagrangian marker points and Eulerian body-fitted curvilinear or unstructured grids. In the cut-cell method [24,27], the interface is tracked explicitly on a regular Cartesian grid and the grid cells cut by the interface are modified according to their intersections with the interface. The interface cells are then treated specifically in order to accurately discritize the governing equations. Although the method has been successfully applied to a variety of two-dimensional problems [24,27], the main difficulty with this method arises from the large number of possible intersections between the fixed grid and the interface resulting in various types of interface cells each requiring a special treatment. In addition, in the cases of highly deformed interfaces, the interface cells may have unavoidable irregular shapes with very small volumes and very large aspect ratios, which adversely impacts the accuracy and stability of the flow solver. Furthermore, it is not straightforward to incorporate the method in existing flow solvers as it requires to modify the basic solution algorithm. The present method overcomes these difficulties while maintaining the main advantages of the front-tracking method. In this approach, the front-tracking methodology is extended to body-fitted curvilinear grids and is combined with a newly developed finite-volume method to facilitate accurate and efficient modeling of strong interactions between the phases and complex solid boundaries. The method incorporates an efficient and robust tracking algorithm developed for tracking the front marker points in body-fitted curvilinear grids. The tracking algorithm utilizes an auxiliary uniform Cartesian grid and it can be easily adapted to unstructured grids as well. The algorithm reduces particle tracking in a curvilinear grid to tracking on a uniform Cartesian grid with a look up table. Furthermore, it can be used in other applications where Lagrangian particles have to be tracked on curvilinear or unstructured grids such as the particle-based Monte Carlo method widely used for solving the PDF equations of turbulent reacting flows [10,18]. The finite-volume method is based on the concept of dual (or pseudo) time-stepping method. The dual time-stepping method uses sub-iterations in pseudo time and has a number of advantages including direct coupling of the continuity and momentum equations for incompressible flows, the elimination of factorization error in factored implicit schemes, the elimination of errors due to approximations made in the implicit operator to improve numerical efficiency, the elimination of errors due to lagged boundary conditions at the solid and internal fluid boundaries, and ability to use non-physical, preconditioned iterative methods for more efficient convergence of the sub-iterations [5].

The main advantages of the present method that make it attractive compared to alternative approaches can be summarized as follows:

1. It retains all the advantages of the front-tracking method [23] while treating complex geometries in a natural way using a body-fitted curvilinear grid without substantial increase in computational cost.
2. It does not require any major modification to the basic flow solver so that it can be easily incorporated into virtually all existing flow solvers including commercial CFD packages through user defined functions (UDFs).
3. It is straightforward to extend the present approach to unstructured grids and to three-dimensional geometries.
4. The tracking algorithm is very robust and computational efficient. It reduces particle tracking in a curvilinear grid to tracking on a uniform Cartesian grid with a look up table and can be used in other applications where Lagrangian particles have to be tracked on a curvilinear or unstructured grid as mentioned above.

The method is implemented to compute two-dimensional (planar or axisymmetric) interfacial flows in complex geometries and has been successfully applied to compute the motion and breakup of viscous drops in complex geometries [11,15] and mixing in a plug moving through a serpentine channel [12]. In the present study, the performance of the tracking algorithm is tested and its temporal and spatial accuracies are quantified in a simple setting of a rigid body rotation of fluid in a two-dimensional circular channel. The method is

then applied to compute the motion of the drops falling due to gravity in a straight channel studied earlier by Han and Tryggvason [8]. It is found that the present results are in a very good agreement with the results obtained by Han and Tryggvason [8]. Finally, the method is applied to buoyancy-driven motion of drops in constricted channels studied experimentally by Hemmat and Borhan [9].

In Section 2, the governing equations are briefly reviewed and are transformed into an arbitrary curvilinear coordinate system. Then the tracking algorithm and the finite-volume/front-tracking method are described in Section 3. The results are presented and discussed in Section 4 and some conclusions are drawn in Section 5.

2. Mathematical formulation

The mathematical formulation is briefly described in this section for an axisymmetric incompressible flow of a Newtonian fluid. The incompressible flow equations for an axisymmetric flow can be written in the cylindrical coordinates in the vector form as

$$\frac{\partial \mathbf{q}}{\partial t}+\frac{\partial \mathbf{f}}{\partial r}+\frac{\partial \mathbf{g}}{\partial z}=\frac{\partial \mathbf{f}_v}{\partial r}+\frac{\partial \mathbf{g}_v}{\partial z}+\mathbf{h}_v+\mathbf{f}_{\mathrm{b}}, \tag{1}$$

where

$$\mathbf{q}=\begin{Bmatrix}0\\ r\rho v_r\\ r\rho v_z\end{Bmatrix},\quad \mathbf{f}=\begin{Bmatrix}rv_r\\ r(\rho v_r^2+p)\\ r\rho v_r v_z\end{Bmatrix},\quad \mathbf{g}=\begin{Bmatrix}rv_z\\ r\rho v_r v_z\\ r(\rho v_z^2+p)\end{Bmatrix}, \tag{2}$$

and

$$\mathbf{f}_v=\begin{Bmatrix}0\\ \tau_{rr}\\ \tau_{zr}\end{Bmatrix},\quad \mathbf{g}_v=\begin{Bmatrix}0\\ \tau_{zr}\\ \tau_{zz}\end{Bmatrix},\quad \mathbf{h}_v=\begin{Bmatrix}0\\ p-\frac{2}{r}\frac{\partial}{\partial r}(r\mu v_r)-\frac{\partial}{\partial z}(\mu v_z)\\ -\frac{\partial}{\partial z}(\mu v_z)\end{Bmatrix}. \tag{3}$$

In Eqs. (1)–(3), r and z are the radial and axial coordinates and t is the physical time; ρ, μ and p are the fluid density, the dynamic viscosity and pressure; v_r and v_z are the velocity components in r and z coordinate directions, respectively. The viscous stresses appearing in the viscous flux vectors are given by

$$\tau_{rr}=2\mu\frac{\partial v_r}{\partial r},\quad \tau_{zz}=2\mu\frac{\partial v_z}{\partial z},\quad \tau_{zr}=\mu\left(\frac{\partial v_r}{\partial z}+\frac{\partial v_z}{\partial r}\right). \tag{4}$$

The last term in Eq. (1) represents the body forces resulting from the buoyancy and surface tension and is given by

$$\mathbf{f}_{\mathrm{b}}=-r(\rho_{\mathrm{o}}-\rho)\mathbf{G}-\int_S r\sigma\kappa\mathbf{n}\delta(\mathbf{x}-\mathbf{x}_{\mathrm{f}})\,\mathrm{d}s, \tag{5}$$

where the first term represents the body force due to buoyancy with ρ_{o} and $\mathbf{G}$ being the density of ambient fluid and the gravitational acceleration, respectively. The second term in Eq. (5) represents the body force due to the surface tension, and δ, $\mathbf{x}_{\mathrm{f}}$, σ, κ, $\mathbf{n}$, S and $\mathrm{d}s$ denote the Dirac delta function, the location of the front, the surface tension coefficient, the twice of the mean curvature, the outward unit normal vector on the interface, the surface area of the interface and the surface area element of the interface, respectively.

In Eq. (1), the fluids are assumed to be incompressible so that the density of a fluid particle remains constant, i.e., $\frac{\mathrm{D}\rho}{\mathrm{D}t}=0$, where the substantial derivative is defined as $\frac{\mathrm{D}}{\mathrm{D}t}=\frac{\partial}{\partial t}+\mathbf{u}\cdot\nabla$. It is also assumed that the viscosity in each fluid particle remains constant $\frac{\mathrm{D}\mu}{\mathrm{D}t}=0$. As can be seen in Eq. (1), the continuity equation is decoupled from the momentum equations since it does not have any time derivative term. In order to overcome this difficulty and to be able to use a time-marching solution algorithm, artificial time derivative terms are added to the flow equations in the form

$$\Gamma^{-1}\frac{\partial \mathbf{w}}{\partial \tau}+\mathbf{I}^1\frac{\partial \rho\mathbf{w}}{\partial t}+\frac{\partial \mathbf{f}}{\partial r}+\frac{\partial \mathbf{g}}{\partial z}=\frac{\partial \mathbf{f}_v}{\partial r}+\frac{\partial \mathbf{g}_v}{\partial z}+\mathbf{h}_v+\mathbf{f}_{\mathrm{b}}, \tag{6}$$

where τ is the pseudo time. The solution vector $\mathbf{w}$, the incomplete identity matrix I^1 and the preconditioning matrix Γ^{-1} are given by

$$\mathbf{w} = \begin{Bmatrix} rp \\ rv_r \\ rv_z \end{Bmatrix}, \quad I^1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \Gamma^{-1} = \begin{bmatrix} \frac{1}{\rho\beta^2} & 0 & 0 \\ 0 & \rho & 0 \\ 0 & 0 & \rho \end{bmatrix}, \tag{7}$$

where β is the preconditioning parameter to be determined and it has dimensions of velocity. Note that the relation $\mathbf{q} = \rho I^1 \mathbf{w}$ has been used in Eq. (6). Based on a simple analysis, a nearly optimal value of the preconditioning parameter β is given by

$$\beta^2 = \kappa_\beta U_{\text{ref}}^2 \left[1 + \frac{1}{4Re_\ell^2}\right], \tag{8}$$

where κ_β is a constant of order of unity, U_{ref} and ℓ are the velocity and length scales, respectively, and the Reynolds number is defined as $Re_\ell = U_{\text{ref}}\ell/\mu$.

Eq. (6) can be transformed into a general, curvilinear coordinate system

$$\xi = \xi(r,z), \quad \eta = \eta(r,z), \tag{9}$$

and the resulting equations take the form

$$\Gamma^{-1}\frac{\partial h\mathbf{w}}{\partial \tau} + \mathbf{I}^1\frac{\partial \rho h\mathbf{w}}{\partial t} + \frac{\partial h\mathbf{F}}{\partial \xi} + \frac{\partial h\mathbf{G}}{\partial \eta} = \frac{\partial h\mathbf{F}_v}{\partial \xi} + \frac{\partial h\mathbf{G}_v}{\partial \eta} + h(\mathbf{h}_v + \mathbf{f}_b), \tag{10}$$

where $h = r_\xi z_\eta - r_\eta z_\xi$ represents the Jacobian of the transformation. The transformed inviscid and viscous flux vectors in Eq. (10) are defined as

$$\begin{aligned} h\mathbf{F} &= z_\eta \mathbf{f} - r_\eta \mathbf{g}, \quad h\mathbf{G} = -z_\xi \mathbf{f} + r_\xi \mathbf{g}, \\ h\mathbf{F}_v &= z_\eta \mathbf{f}_v - r_\eta \mathbf{g}_v, \quad h\mathbf{G}_v = -z_\xi \mathbf{f}_v + r_\xi \mathbf{g}_v. \end{aligned} \tag{11}$$

3. Numerical method

Three types of grid used in the present method are sketched in Fig. 1. A fixed curvilinear grid is used to solve the conservation equations (Eq. (10)) while a Lagrangian grid of lower dimension is used to track the interface separating different phases. An auxiliary uniform Cartesian grid is used to maintain computationally efficient communication between the curvilinear and Lagrangian grids.

3.1. Tracking algorithm

The interface between different phases is represented by a Lagrangian grid which consists of connected marker points as sketched in Fig. 1. The marker points can be considered as fluid particles moving with local flow velocity. In order to maintain communication between the Lagrangian and fixed curvilinear grids, it is necessary to determine the locations of the marker points in the curvilinear grid at every physical time step. Although it is a simple task to determine locations of the marker points in a uniform Cartesian mesh, it is substantially more difficult to track them in a general curvilinear or in an unstructured grid. To overcome this difficulty and to keep tracking computationally feasible, a new tracking algorithm is developed. The present tracking algorithm utilizes an auxiliary uniform Cartesian grid as sketched in Fig. 1. The overall algorithm can be summarized as follows: At the beginning of each simulation, a uniform Cartesian grid is generated such that it covers the entire computational domain. The cell size of the uniform grid is typically taken as the half of the size of the smallest curvilinear grid cell. It is then found which uniform Cartesian grid nodes reside in each curvilinear grid cell and this information is stored in an array. Referring to the sketch in Fig. 2, for example, the nodal point Q is found to be in the curvilinear grid cell $ABCD$ by performing the vector operations $\vec{k}\cdot(|\overrightarrow{AB}|\times|\overrightarrow{AQ}|) \geqslant 0$, $\vec{k}\cdot(|\overrightarrow{BC}|\times|\overrightarrow{BQ}|) \geqslant 0$, $\vec{k}\cdot(|\overrightarrow{CD}|\times|\overrightarrow{CQ}|) \geqslant 0$, and $\vec{k}\cdot(|\overrightarrow{DA}|\times|\overrightarrow{DQ}|) \geqslant 0$, where $\vec{k}$ is the outward unit vector perpendicular to the cell $ABCD$. The same procedure is used for all other nodal points

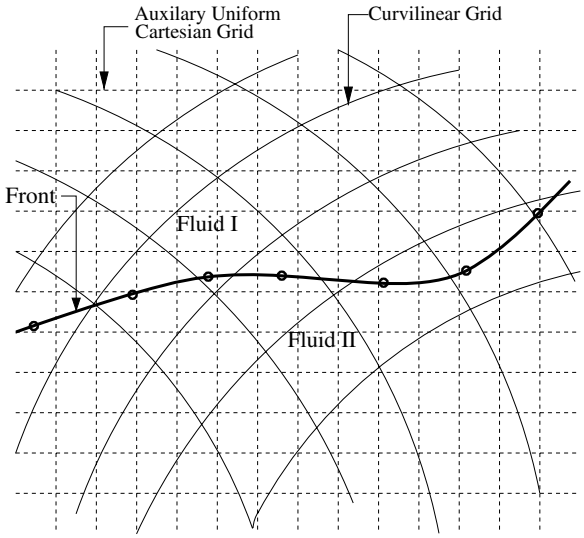


Fig. 1. Three types of grids used in the computations. The governing equations are solved on a fixed Eulerian curvilinear grid and the interface between different phases is represented by a Lagrangian grid consisting of connected marker points. An auxiliary uniform Cartesian grid is used to maintain communication between the curvilinear and Lagrangian grids.

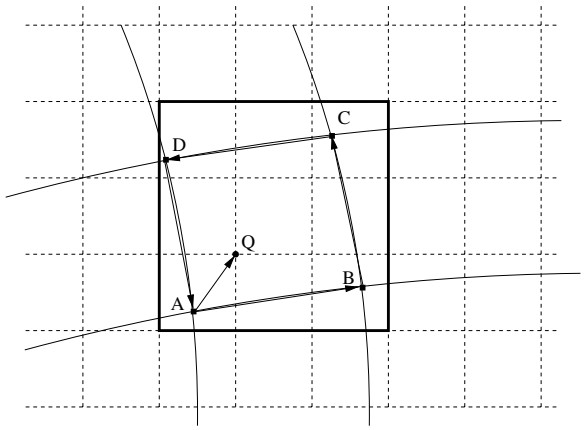


Fig. 2. Preprocessing of the uniform Cartesian grid. Vector algebra is used to determine which uniform Cartesian grid nodes reside in each curvilinear grid cell.

enclosed by the rectangle consisting of the uniform grid cells that enclose $ABCD$, i.e., the uniform grid cells in the rectangle shown by thick solid borderline in the sketch, and the entire process is repeated for all other curvilinear grid cells. It is emphasized that all these computations are done only once at the beginning of each simulation as a preprocessing. Then, in each physical time step, it is first determined where the front points reside in the uniform grid. Referring to the sketch in Fig. 3, for instance, it is first found that the front point P is in (I,J) cell of the uniform grid and then it is determined that the nodes of (I,J) cell reside in the curvilinear grid cells (i,j), $(i,j-1)$ and $(i-1,j-1)$ as shown by the dashed line in the sketch. As a result, we conclude that the front point P resides in the region consisting of the curvilinear grid cells $(i-1{:}i,j-1{:}j)$ and this region is expanded to include the cells $(i-2{:}i+1,j-2{:}j+1)$. Finally, the cells $(i-2{:}i+1,j-2)$ and $(i-2,j-2{:}j+1)$ are eliminated based on the relative distance of their outer nodes to the point P compared to the cells $(i-2{:}i+1,j+1)$ and $(i+1,j-2{:}j+1)$, respectively. At the end of this process, it is determined that the front point P resides in the domain composed by the cells $(i-1{:}i+1,j-1{:}j+1)$. The front properties

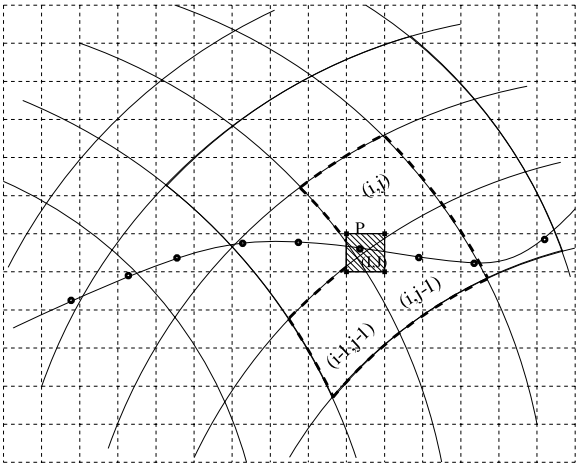


Fig. 3. The tracking algorithm for curvilinear grids.

evaluated at the point P are distributed onto these cells and flow variables such as velocity field are interpolated onto point P from these cells. It is found that this tracking algorithm is very robust and computationally efficient as it essentially reduces the particle tracking in curvilinear grid to a particle tracking on a uniform grid with a look up table. Since the present tracking algorithm is general, it can also be used to track Lagrangian points in other methods such as the particle-based Monte Carlo method widely used in solving the probability density function (PDF) model equations of turbulent reacting flows [10,18]. In addition, it can also be easily adapted for unstructured grids.

3.2. Front-tracking method

The effects of the surface tension forces and variable material properties in different phases are accounted for using the front-tracking method similar to that of Unverdi and Tryygvason [25]. The interface is represented by connected Lagrangian marker points. Each piece of the interface between two neighboring marker points is called an inteface (or front) element. The surface tension force is computed at the centroids of the front elements using a third-order Lagrange polynomial fit in a similar way as described by Tryggvason et al. [23] and is distributed over 16 neighboring curvilinear grid cells in a conservative manner using a tensor-product kernel. For instance, the kernel function used to distribute the surface tension force computed on lth interface element whose centroid is located at (r_c^l, z_c^l) onto (i,j) grid point is defined as

$$w_{ij}^l = K(r_{ij}^l)K(z_{ij}^l), \tag{12}$$

where $r_{ij}^l = |r_c^l - r_{ij}|/r_{\max}$ and $z_{ij}^l = |z_c^l - z_{ij}|/z_{\max}$. The length scales $r_{\max}$ and $z_{\max}$ are selected as the maximum distances between the centroid of the interface element and grid nodes on which the surface tension force $\mathbf{f}_b^l$ is to be distributed in r and z directions, respectively. The weighting functions given by Eq. (12) are normalized in order to satisfy the consistency condition, i.e.,

$$\sum_i \sum_j w_{ij}^l = 1, \tag{13}$$

where the summation is carried over all the grid points associated with the interface element l. A linear kernel function is used in this study in the form

$$K(\hat{r}) = \begin{cases} 1.0 - \hat{r} & \text{if } \hat{r} \leqslant 1.0, \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

Note that the same kernel function is also used for interpolating the velocity field from the curvilinear grid onto the locations of the marker points.

The material properties such as density and viscosity are computed according to

$$\begin{aligned}\rho &= \phi\rho_{\mathrm{d}} + (1-\phi)\rho_{\mathrm{o}},\\ \mu &= \phi\mu_{\mathrm{d}} + (1-\phi)\mu_{\mathrm{o}},\end{aligned} \tag{15}$$

where the subscripts o and d refer to the ambient and the drop fluids, respectively. The indicator function ϕ is defined such that it is unity inside and zero outside of the drops and, following Tryggvason et al. [23], it is obtained by solving the Poisson equation

$$\nabla^2\phi = \nabla_h \cdot \nabla_h\phi, \tag{16}$$

where ∇_h is the discrete version of the gradient operator. The jump $\nabla_h\phi$ is distributed on the neighboring grid cells using the Peskin distribution [17] and Eq. (16) is then solved on the uniform grid in the vicinity of each drop. After computing the indicator function on the uniform grid, it is interpolated onto the curvilinear grid using bilinear interpolations. Note that it is possible to efficiently solve the Poisson equation on the curvilinear grid but the above procedure seems to be robust and produces sufficiently smooth solutions for the problems studied in the present work.

3.3. Integration of the flow equations

Once the material properties and surface tension forces have been determined, any standard time-marching algorithm can, in principle, be used to solve Eq. (10) since it is in the same form as the usual continuum flow equations. Following Caughey [5], a two-parameter family of numerical scheme to solve Eq. (10) can be written as

$$\begin{aligned}&\Gamma^{-1}h\frac{\mathbf{w}^{p+1}-\mathbf{w}^{p}}{\Delta\tau} + I^1 h\frac{(2+\varphi)(\rho\mathbf{w})^{p+1} - 2(1+\varphi)(\rho\mathbf{w})^{n} + \varphi(\rho\mathbf{w})^{n-1}}{2\Delta t}\\ &\quad = -\theta\left[\frac{\partial h(\mathbf{F}-\mathbf{F}_v)}{\partial\xi} + \frac{\partial h(\mathbf{G}-\mathbf{G}_v)}{\partial\eta}\right]^{p+1} + \theta h(\mathbf{h}_v^{p} + \mathbf{f}_{\mathrm{b}}^{n+1}) - (1-\theta)\left[\frac{\partial h(\mathbf{F}-\mathbf{F}_v)}{\partial\xi} + \frac{\partial h(\mathbf{G}-\mathbf{G}_v)}{\partial\eta} - h(\mathbf{h}_v+\mathbf{f}_{\mathrm{b}})\right]^{n},\end{aligned} \tag{17}$$

where $(\cdots)^p$ and $(\cdots)^n$ denote the pseudo and physical time levels, respectively. The parameter φ governs the approximation to the physical time-derivative and θ determines the level of implicitness of the method in pseudo time. Note that when a steady state is reached in pseudo time, we have $\mathbf{w}^p \to \mathbf{w}^{n+1}$. Three combinations of φ and θ are of particular interest and correspond to different approximations in the physical time [5]. In the present study, both φ and θ are set to unity, which corresponds to a second-order three-point backward implicit method. Note that the terms $\mathbf{h}_v$ and $\mathbf{f}_{\mathrm{b}}$ are treated explicitly in the pseudo time in the present formulation although it is possible to include $\mathbf{h}_v$ into the implicit operator. The increment $\Delta\tau$ represents the time step for sub-iteration while Δt represents the physical time step which is usually different. Using the standard techniques [2,3], Eq. (17) can be approximated and put in the factorized form as follows:

$$\left\{S + \frac{\theta}{h}\left(\frac{\partial}{\partial\xi}A - \frac{\partial}{\partial\xi}A_v\frac{\partial}{\partial\xi}\right)\right\}S^{-1}\left\{S + \frac{\theta}{h}\left(\frac{\partial}{\partial\eta}B - \frac{\partial}{\partial\eta}B_v\frac{\partial}{\partial\eta}\right)\right\}\Delta\mathbf{w}^{p} = \mathbf{R}^{p}, \tag{18}$$

where the residual vector $\mathbf{R}^p$ is given by

$$\begin{aligned}\mathbf{R}^{p} = &-I^1\frac{(2+\varphi)(\rho\mathbf{w})^{p} - 2(1+\varphi)(\rho\mathbf{w})^{n} + \varphi(\rho\mathbf{w})^{n-1}}{2\Delta t} - \frac{\theta}{h}\left[\frac{\partial h(\mathbf{F}-\mathbf{F}_v)}{\partial\xi} + \frac{\partial h(\mathbf{G}-\mathbf{G}_v)}{\partial\eta} - h\mathbf{h}_v\right]^{p}\\ &- \theta\mathbf{h}_{\mathrm{b}}^{n+1} - \frac{(1-\theta)}{h}\left[\frac{\partial h(\mathbf{F}-\mathbf{F}_v)}{\partial\xi} + \frac{\partial h(\mathbf{G}-\mathbf{G}_v)}{\partial\eta} - h(\mathbf{h}_v+\mathbf{f}_{\mathrm{b}})\right]^{n},\end{aligned} \tag{19}$$

and the Jacobian matrices are defined as

$$A^{p} = \left\{\frac{\partial h\mathbf{F}}{\partial\mathbf{w}}\right\}^{p}, \quad B^{p} = \left\{\frac{\partial h\mathbf{G}}{\partial\mathbf{w}}\right\}^{p}, \quad A_v^{p} = \left\{\frac{\partial h\mathbf{F}_v}{\partial\mathbf{w}_\xi}\right\}^{p}, \quad B_v^{p} = \left\{\frac{\partial h\mathbf{G}_v}{\partial\mathbf{w}_\eta}\right\}^{p}, \tag{20}$$

where $\mathbf{w}_\xi = \frac{\partial \mathbf{w}}{\partial \xi}$ and $\mathbf{w}_\eta = \frac{\partial \mathbf{w}}{\partial \eta}$. In Eq. (18), we define $\mathbf{S} = \frac{\Gamma^{-1}}{\Delta\tau} + I^1 \frac{(2+\varphi)}{2\Delta t}\rho^{p+1}$, where ρ^{n+1} is approximated as $\rho^{p+1} \cong \rho^{n+1}$ in the solution process. Eq. (18) can be solved in two steps using a block tridiagonal solver when the spatial derivatives are discretized by three-point approximations.

The spatial derivatives are approximated using a cell-centered finite-volume method that is equivalent to second order central differences on a uniform Cartesian grid. A blend of second and fourth order numerical dissipation terms similar to that of Caughey [4] are added to the right-hand side of Eq. (18) to prevent the odd–even decoupling and to damp excessive oscillations in the vicinity of discontinuities or sharp gradients. Note that the numerical dissipation terms are treated explicitly in the pseudo time. Since the accuracy in pseudo time is not of interest, in addition to the preconditioning method, a multigrid method similar to that of Caughey [5] and a local time-stepping method are used to further accelerate the convergence rate in pseudo time.

3.4. The overall solution procedure

The finite-volume and front-tracking methods described above are combined as follows. In advancing solutions from physical time level n ($t_n = n \cdot \Delta t$) to level $n+1$, the locations of the marker points at the new time level $n+1$ are first predicted using an explicit Euler method, i.e.,

$$\widetilde{\mathbf{X}}_p^{n+1} = \mathbf{X}_p^n + \Delta t \mathbf{V}_p^n, \tag{21}$$

where $\mathbf{X}_p$ and $\mathbf{V}_p$ denote the position of front marker points and the velocity interpolated from the neighboring curvilinear grid points onto the front point $\mathbf{X}_p$, respectively. Then the material properties and surface tension are evaluated using the predicted front position as

$$\rho^{n+1} = \rho(\widetilde{\mathbf{X}}_p^{n+1}), \quad \mu^{n+1} = \mu(\widetilde{\mathbf{X}}_p^{n+1}), \quad \mathbf{f}_{\mathrm{b}}^{n+1} = \mathbf{f}_{\mathrm{b}}(\widetilde{\mathbf{X}}_p^{n+1}). \tag{22}$$

The velocity and pressure fields at new physical time level $n+1$ are then computed by solving the flow equations (Eq. (10)) by the FV method for a single physical time step and finally the positions of the front points are corrected as

$$\mathbf{X}_p^{n+1} = \mathbf{X}_p^n + \frac{\Delta t}{2}(\mathbf{V}_p^n + \mathbf{V}_p^{n+1}). \tag{23}$$

After this step the material properties and the body forces are re-evaluated using the corrected front position. The method is second order accurate in time and it has second order spatial accuracy in the smooth regions when a uniform Cartesian grid is used. However, the method is only first order in space near the interface due to smearing of the discontinuities.

Perfect reflection boundary conditions are used at the solid boundary for the front marker points, i.e., the front marker points crossing the solid boundary due to numerical error are reflected with respect to the inward normal vector back into the computational domain. If the front marker point is close to the boundary as sketched in Fig. 4, the front properties are distributed onto curvilinear cells in a conservative manner, i.e., the weights are defined only for the cells within the computational domain (shown by the thick solid line in the sketch) and are normalized to satisfy the consistency condition. The grid properties are interpolated onto the front point in a similar manner.

The Lagrangian grid is initialized such that element sizes are comparable with the local Eulerian grid cells and is kept so throughout the computations by deleting small elements and splitting the large elements. The initial front element size is typically set to $0.75\Delta l_m$, where Δl_m is the average size of the curvilinear grid cells in which the front element resides. The size of a curvilinear cell is defined as $\Delta l = 4A/P$, where P and A are the perimeter and the area of the grid cell, respectively. During the simulation, in each physical time step, the elements that are smaller than $0.5\Delta l_m$ are deleted and the elements that are larger than $1.2\Delta l_m$ are split in order to keep the Lagrangian grid comparable to the local Eulerian grid and to prevent the formation of wiggles much smaller than the grid size. Note that the restructuring procedure described above takes the curvature of the interface into account using third order interpolations. However, it does not necessarily conserve the mass and may contribute significantly to the change in drop volume unless care is taken. It is found that the interface restructuring is nearly optimal when the element sizes are kept comparable to the local curvilinear grid

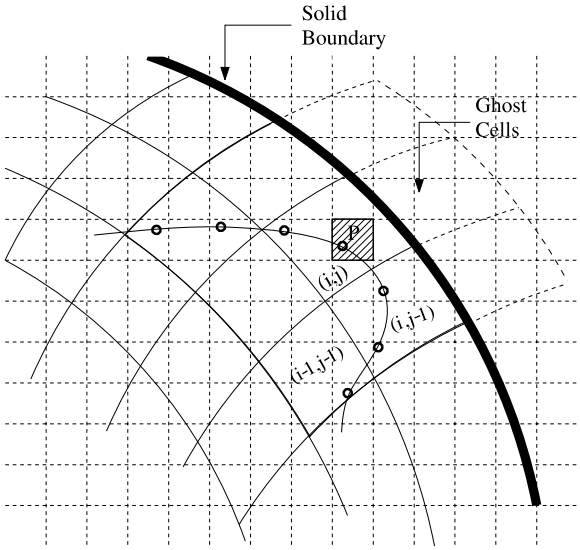


Fig. 4. Distribution of front properties from the Lagrangian grid onto the curvilinear grid and interpolation of flow quantities from the curvilinear grid onto the Lagrangian grid near a solid boundary.

size and does not contribute significantly to the change in drop volume. Total change in drop volume is found to be less than 10% in all the cases studied in the present paper. Although not employed here, a volume correction algorithm has been recently devised that iteratively corrects the positions of the marker points such that the global mass conservation of the drop is satisfied within a prescribed error tolerance [12].

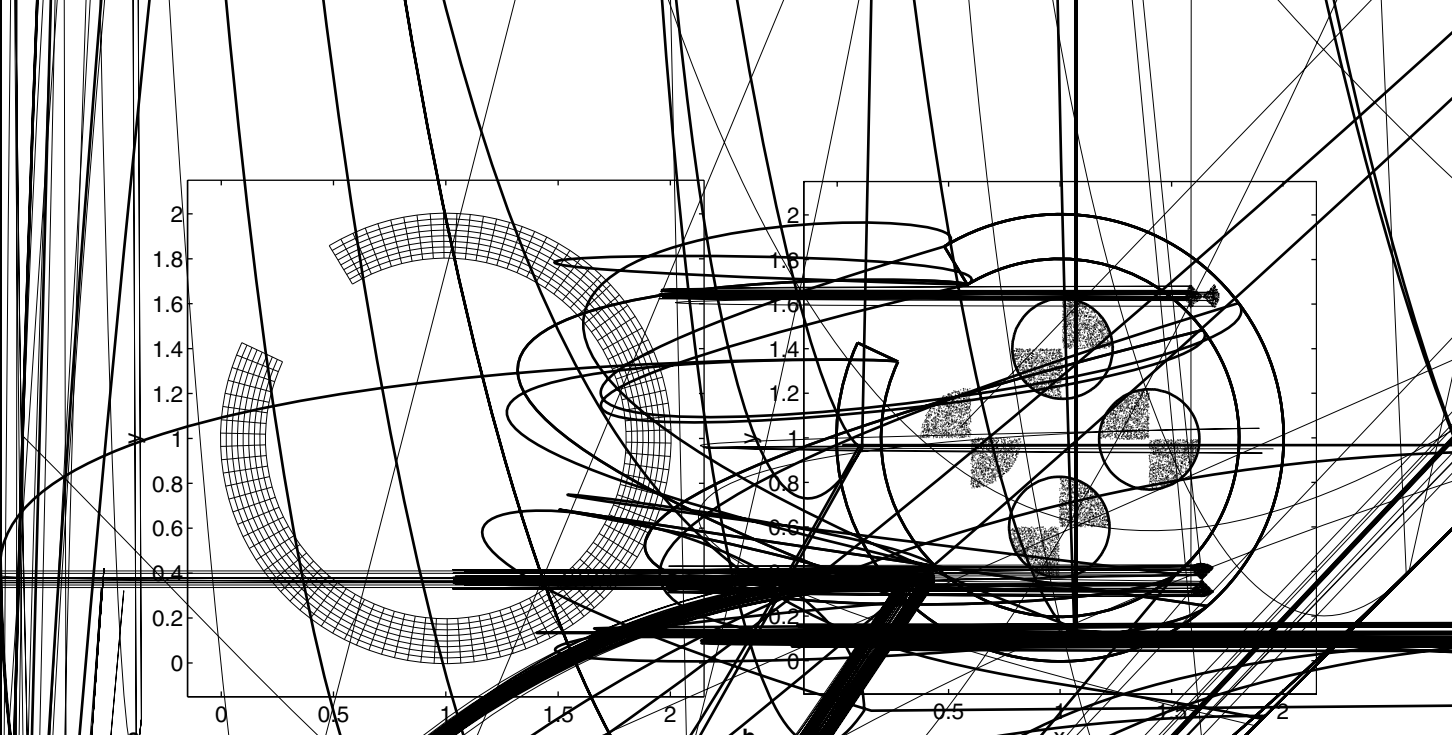
4. Results and discussion

4.1. Particle tracking

First the particle tracking algorithm is tested in a simple setting of a rigid body rotation of fluid in a circular channel as shown in Fig. 5. The radius of the outer boundary and the width of the channel are set to $R_c = 1$ and $w_c = 0.2$, respectively. The velocity field is specified as

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -y + y_0 \\ x - x_0 \end{pmatrix}, \tag{24}$$

where x and y are the components of the two-dimensional Cartesian coordinates; x_0 and y_0 are the centroid of the circular channel; and u and v are the x and y components of the velocity vector, respectively. A two-dimensional drop of diameter $d_d = 0.15$ centered at $(x, y) = (0.1, 1.0)$ is set into motion by the fluid and tracer marker particles are used to visualize its motion as shown in Fig. 5(b). The tracer particles are distributed inside the drop at random and the particles occupying the first and third quarters of the drop are identified as "red" while the rest are "blue". Only blue particles are shown in all the results presented here for clarity. The particles are moved with the local flow velocity interpolated from the neighboring computational grid points using the same scheme as used for advecting the interface marker points. A coarse version of the curvilinear grid used in the simulations is shown in Fig. 5(a). To demonstrate the accuracy of the tracking algorithm, snapshots of the drop are shown in Fig. 5(b) at six different locations in the channel. The initial location of the drop is denoted by "Location 1" and the other locations are denoted in the counter clockwise direction by "Location 2",...,"Location 6", respectively. In Fig. 5(b), the drops are enlarged three times and are plotted outside the channel at four different positions to better show the evolution of marker points. This figure clearly shows the rigid body motion of the drop indicating the accuracy of the tracking algorithm. The temporal and spatial accuracies of the tracking algorithm are quantified in Figs. 6(a) and (b), respectively. The error is defined as the difference between the computed and exact locations of the drop centroid and is plotted



at "Location 3" and "Location 6", corresponding to the bottom and top positions of the drop shown in Fig. 5(b). Note that error is scaled by the initial drop radius. In Fig. 6(a), the error is plotted against the time step Δt. This figure clearly shows that the tracking algorithm is second order accurate in time. In Fig. 6(b), the error is plotted against the inverse of the total number of grid cells M^{-2}. As can be seen in this figure, the tracking algorithm is also second order accurate in space as expected. Note that no grid restructuring is performed for the 9-node elements and, although not shown here, the average temporal and spatial errors based on the tracer particles are found to scale in the same way as shown in Figs. 6(a) and (b). The effects of the uniform Cartesian grid size relative to the curvilinear grid size are also examined. For this purpose, the error is plotted in Fig. 7 against the the ratio of grid sizes defined as $\zeta = \frac{\Delta x_u}{\Delta x_c}$, where Δx_u and Δx_c are the size of the uniform Cartesian grid and the size of the smallest curvilinear grid cell, respectively. As can be seen in this figure, the error is not very sensitive to the uniform Cartesian grid size if the uniform grid size is of the order of the curvilinear grid size or smaller.

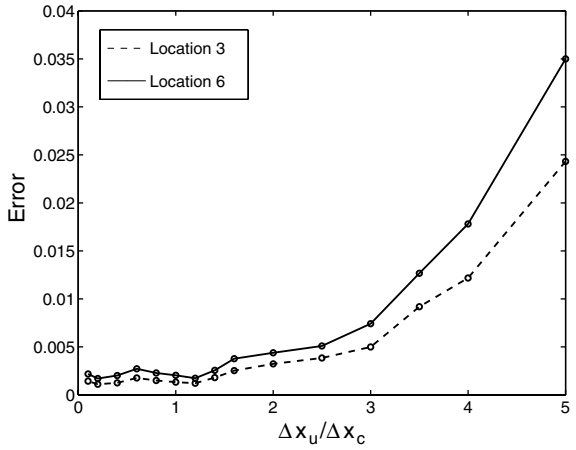


Fig. 7. Error in the position of the drop centroid against the ratio of the average size of the uniform Cartesian grid cells (Δx_u) to the average size of the curvilinear grid cells (Δx_c).

The same tracking algorithm is also used to distribute the surface tension forces onto the neighboring curvilinear grid cells. To test performance of the distribution algorithm, a 2D circular drop is placed in a sinusoidal channel in a hydrostatic equilibrium. A portion of the channel is shown in Fig. 8(b) and full description of the channel can be found in Ref. [12]. A curvilinear grid containing 1024×64 grid cells is used. The material properties and drop size are set such that the Laplace number, $La = 50$, where the Laplace number is defined as $La = \sigma \rho d_d / \mu^2$, and according to the Young–Laplace equation, the pressure difference between the inside and outside of the drop is $\Delta P = \sigma / R = 10$. Note that density and viscosity ratios are set to unity. The pressure distribution is plotted in Fig. 8(a) in the channel. As can be seen in this figure, the pressure jump exactly matches with the theoretical value of 10. The contour plot of the magnitude of the parasite current is shown in Fig. 8(b) to illustrate the distribution and magnitude of "parasitic current". The maximum magnitude of the parasitic velocity is found to be 1.16×10^{-4}. Although not shown here, for the same material properties and the drop size, the present method and the FD/FT method of Tryggvason et al. [23] yield the maximum magnitude of the parasitic current about 2.91×10^{-5} and 6.73×10^{-5}, respectively, in a rectangular domain using a regular Cartesian grid.

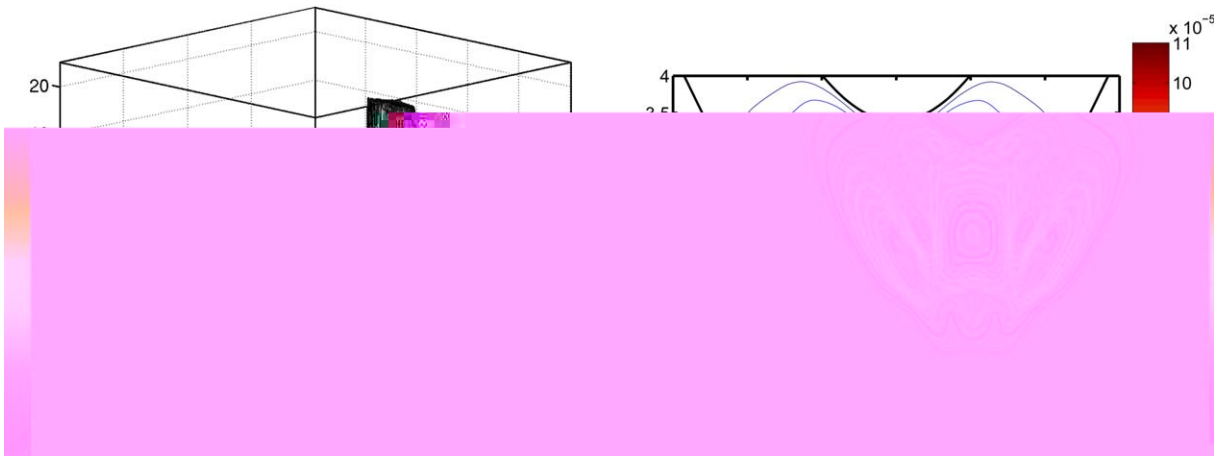


Fig. 8. Pressure and velocity fields for a drop in a hydrostatic equilibrium. (a) Pressure field, (b) contour plot of magnitude of parasitic velocity. The maximum magnitude of the parasitic velocity is about 1.16×10^{-4}.

4.2. Gravity-driven falling drop in a straight channel

The second test case concerns gravity-driven falling drops in a straight channel studied earlier by Han and Tryggvason [8]. The physical problem and computational domain are sketched in Fig. 9(a). As can be seen in this figure, the ambient fluid completely fills the rigid cylinder and the drop that is denser than the ambient fluid accelerates downward due to gravitational body force. The problem is governed by four non-dimensional parameters [8], namely the Eötvös number $E_0 = \frac{g_z \Delta\rho d^2}{\sigma}$ (interchangeably called the Bond number, Bo), the Ohnesorge number $Oh_d = \frac{\mu_d}{\sqrt{\rho_d d\sigma}}$, the density ratio $\rho^* = \frac{\rho_d}{\rho_o}$, and viscosity ratio $\mu^* = \frac{\mu_d}{\mu_o}$, where $\Delta\rho = \rho_d - \rho_o$ is the density difference between the drop and the ambient fluids, g_z is gravitational acceleration and d is the initial drop diameter. The Ohnesorge number based on the ambient fluid is defined similarly as $Oh_o = \frac{\mu_o}{\sqrt{\rho_o d\sigma}}$. The subscripts d and o denote the properties of the drop and ambient fluids, respectively. The non-dimensional time is defined as $t^* = \frac{t}{\sqrt{d/g_z}}$.

The computational domain is $5d$ in radial direction and is $15d$ in the axial direction. No-slip boundary conditions are applied on the cylinder walls and axisymmetry conditions are applied on the centerline. The drop centroid is initially located at $(r_c, z_c) = (0, 12d)$. The computational domain is resolved by a 128×768 uniform Cartesian grid. The grid is stretched in the radial direction to have more grid points close to the centerline. The Ohnesorge number, the density and viscosity ratios are kept constant at $Oh_d = 0.0466$ ($Oh_o = 0.05$), $\rho_d/\rho_o = 1.15$ and $\mu_d/\mu_o = 1$ in all the results presented here. In Fig. 9(b), the evolution of the drop for $Eo = 24$ is presented together with the results obtained with the finite-difference/front-tracking (FD/FT) method [8]. As can be seen in these figures, the present results are in a very good qualitative agreement with the results obtained with FD/FT method. To quantify the accuracy of the present method, the velocity is non-dimensionalized by $\sqrt{g_z d}$ and the non-dimensional velocity of the drop centroid V_c is plotted in Fig. 10(a) together with the FD/FT results. It is clearly seen in this figure that the present results are also in a very good quantitative agreement with those of the FD/FT method demonstrating the accuracy of the present method.

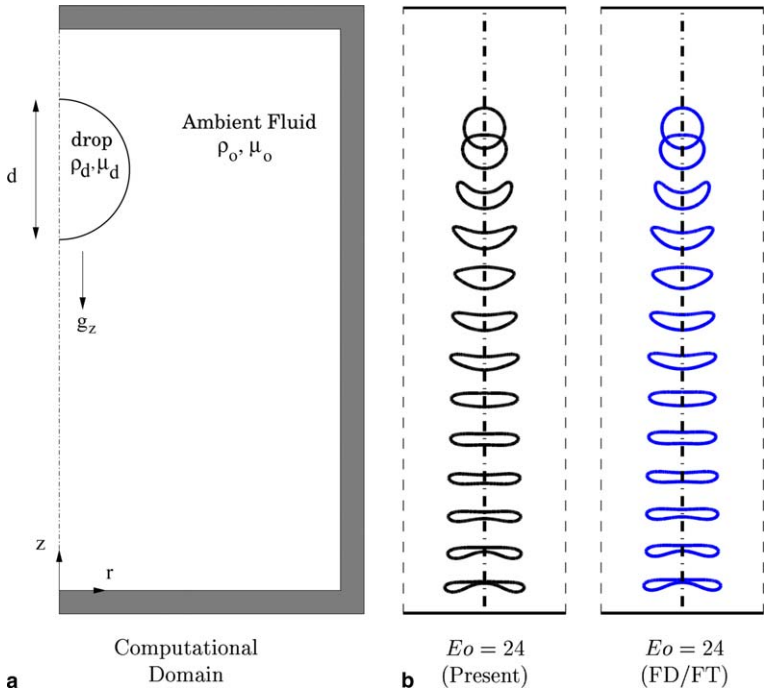


Fig. 9. (a) Schematic illustration of the physical problem and computational domain for a gravity-driven falling drop in a straight channel. (b) Evolution of drop for $Eo = 24$. The gap between two successive drops in each column represents the distance the drop travels at a fixed time interval and the last interface is plotted at $t^* = 44.01$.

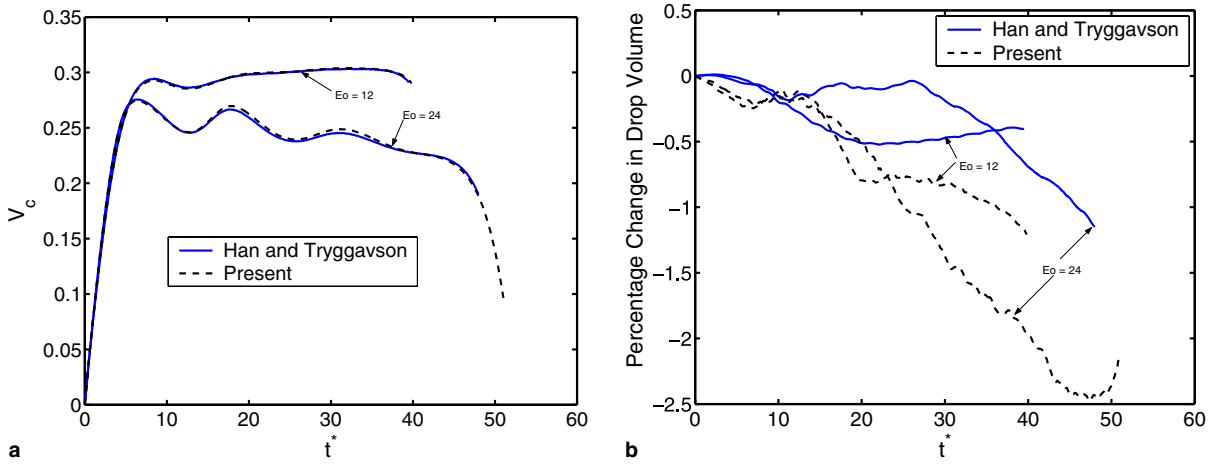


Fig. 10. (a) Velocity of drop centroid and (b) percentage change in drop volume versus t^* for $Eo = 12$ and $Eo = 24$ cases. The dashed lines are the present results and solid lines are the FD/FT results.

Finally, the percentage change in drop volume is plotted in Fig. 10(b) as function of non-dimensional time both for $Eo = 12$ and $Eo = 24$ cases and are again compared with the results of the FD/FT method. This figure shows that the volume changes in the present and FD/FT methods are of the same order and the maximum volume change at the end of the simulation is smaller than 2.5% in the present method. The relatively large volume change in the present method compared to the FD/FT method may be attributed to the larger numerical error in the present results partly due to larger physical time steps and partly due to the interpolation and distribution algorithms.

4.3. Buoyancy-driven rising drops in a continuously constricted channel

The previous test case has confirmed the accuracy of the present method. The final test case concerns the buoyancy-driven motion of viscous drops through a vertical capillary with periodic corrugations studied experimentally by Hemmat and Borhan [9]. The computational setup is sketched in Fig. 11(a). The capillary tube consists of a 26 cm long, periodically constricted cylindrical tube with six corrugations. The average internal radius of the tube is $R = 0.5$ cm, and the wavelength and amplitude of the corrugations are $h = 4$ cm and $A = 0.07$ cm, respectively. The suspending fluid is diethylene glycol–glycerol mixtures. The properties of the drop and suspending fluids are summarized in Table 1 where the same label is used as that used by Hemmat and Borhan [9]. A complete description of the experimental setup can be found in [9]. A portion of a coarse grid containing 8×416 grid cells is plotted in Fig. 11(b) to show the overall structure of the body-fitted grid used in the simulations. The average rise velocity of buoyant drops as well as the drop shapes are computed and the results are compared with the experimental data [9] for a range of the dimensionless drop size, κ, defined as the ratio of the equivalent spherical drop radius to the average capillary radius. In all the results presented in this section, the drops are initially located at $z = 1.5h$ in the ambient fluid that fully fills the cylindrical tube and is initially in the hydrostatic conditions. Symmetry boundary conditions are applied along the centerline and no-slip boundary conditions are used at top, bottom and lateral surfaces of the cylindrical tube. Drops are initially stationary and start rising due to buoyancy. The drops are initially spherical if the initial drop radius is smaller than the capillary tube and are ellipsoidal otherwise. Note that d represents the equivalent drop diameter if the initial shape of the drop is not spherical.

The results are expressed in terms of non-dimensional quantities denoted by superscript "*". The dimensionless coordinates are defined as $z^* = z/h$ and $r^* = r/R$. Time and velocity are made dimensionless with $T_{\text{ref}} = \frac{\mu_o}{\Delta\rho g_z R}$ and $V_{\text{ref}} = \frac{\Delta\rho g_z R^2}{\mu_o}$, respectively.

First a qualitative analysis of the shapes of the drops is shown in Fig. 11(c). In this figure, a sequence of images for the evolution of the shapes of viscous drops through constricted channel are plotted for the non-dimensional drop sizes $\kappa = 0.54$, 0.78 and 0.92. The computations are performed on a 32×1664 grid,

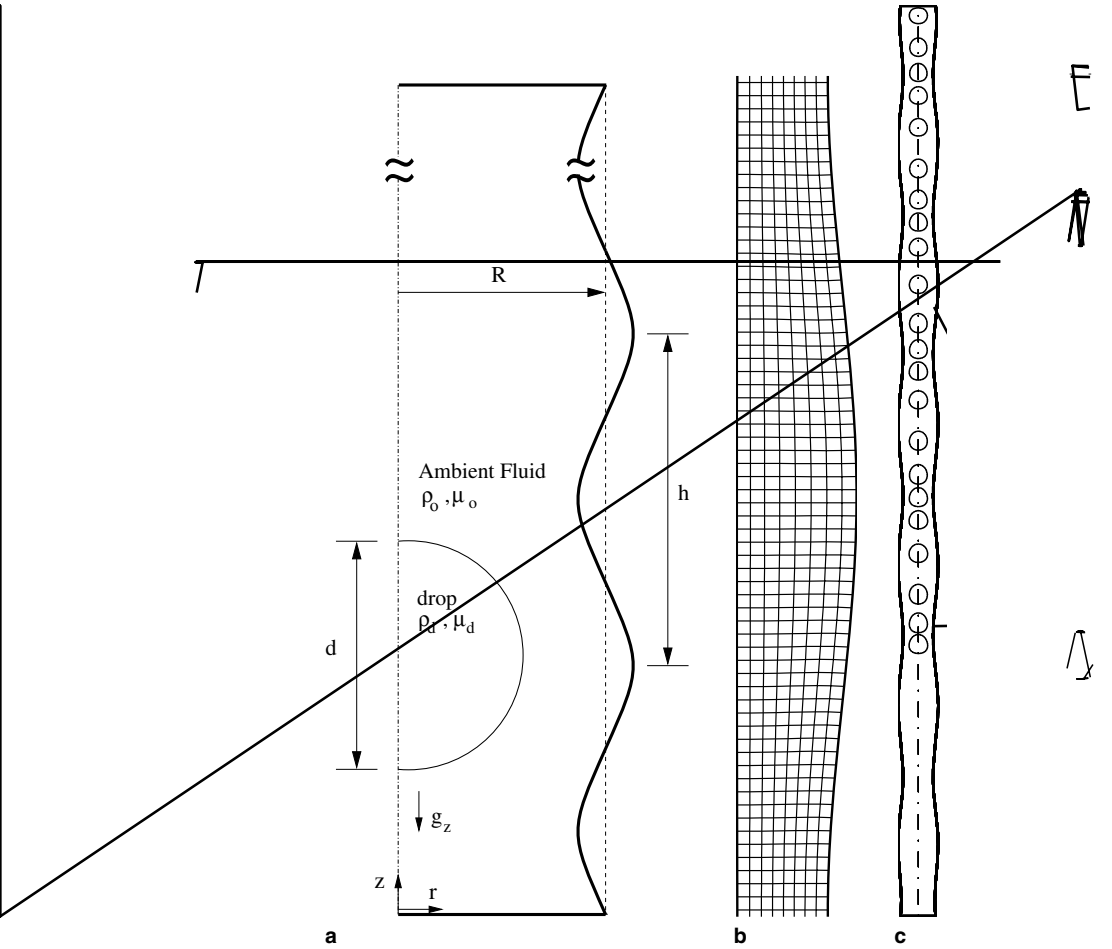


Fig. 11. (a) Schematic illustration of the computational setup for a buoyancy-driven rising drop in a constricted channel. (b) A portion of a coarse computational grid containing 8×416 cells. (c) Snapshots of buoyant drops of DEGG12 system for drops sizes $\kappa = 0.54$, 0.78 and 0.92 from left to right, respectively. The gap between two successive drops in each column represents the distance the drop travels at a fixed time interval and the last interface is plotted from left to right at $t^* = 2831.3$, 3693.0 and 5416.4, respectively.

Table 1
Two-phase system used in the computations

System	Suspending fluid	Drop fluid	μ_o (mPa s)	μ_d (mPa s)	ρ_o (kg/m^3)	ρ_d (kg/m^3)	σ (N/m)
DEGG12	Diethylene glycol–glycerol (63.8 wt%)	UCON-285	87	115	1160	966	0.0042

the physical time step is $\Delta t^* = 1.641$ and the residuals are reduced by three orders of magnitude in each FV sub-iteration. As can be seen in these figures, when a large drop ($\kappa > 0.7$) reaches a constriction, its leading edge follows the capillary wall contour and squeezes through the throat. Once the leading meniscus clears the throat, its rise velocity increases as it enters the diverging cross-section while the trailing edge of the drop remains trapped behind the throat similar to the experimental observations [9]. To better show the effects of the constrictions, the snapshots of the computed and experimental drop shapes before and after the throat of the constriction are shown in Figs. 12 and 13, respectively, for drop sizes $\kappa = 0.54$, 0.65, 0.78, 0.85 and 0.92. Note that the quality of the experimental images is not very good since they are not the original images but are scanned from the experimental paper [9]. As can be seen in these figures, the drop shapes are smooth in all the cases indicating accuracy of the computations and computed drop shapes qualitatively compare well with the experimental results of Hemmat and Borhan [9]. The drop shapes, the velocity field and pressure contours in

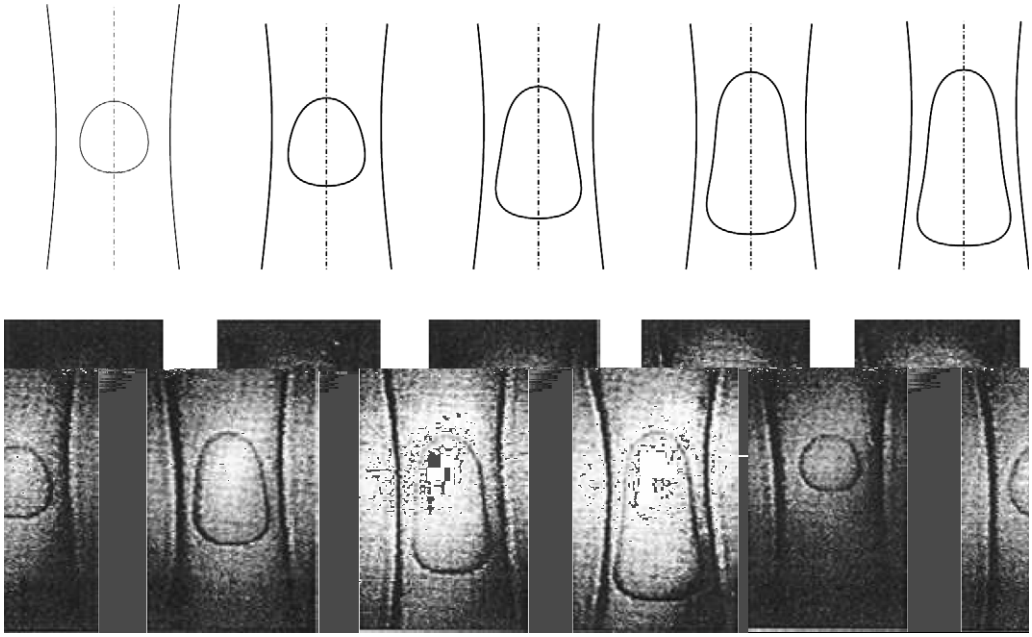


Fig. 12. Snapshots of the drops at the throat of the constriction for the DEGG12 system for drop sizes (from left to right) $\kappa = 0.54$, 0.65, 0.78, 0.85 and 0.92, respectively. Top plots are the computational results and the lower plots are the experimental results of Hemmat and Borhan [9]. *Grid*: 32×1664, $\Delta t^* = 1.641$. Copyright (1996) from buoyancy-driven motion of drops and bubbles in a periodically constricted capillary by Hemmat and Borhan. Reproduced by permission of Taylor & Francis, Inc., http://www.taylorandfrancis.com.

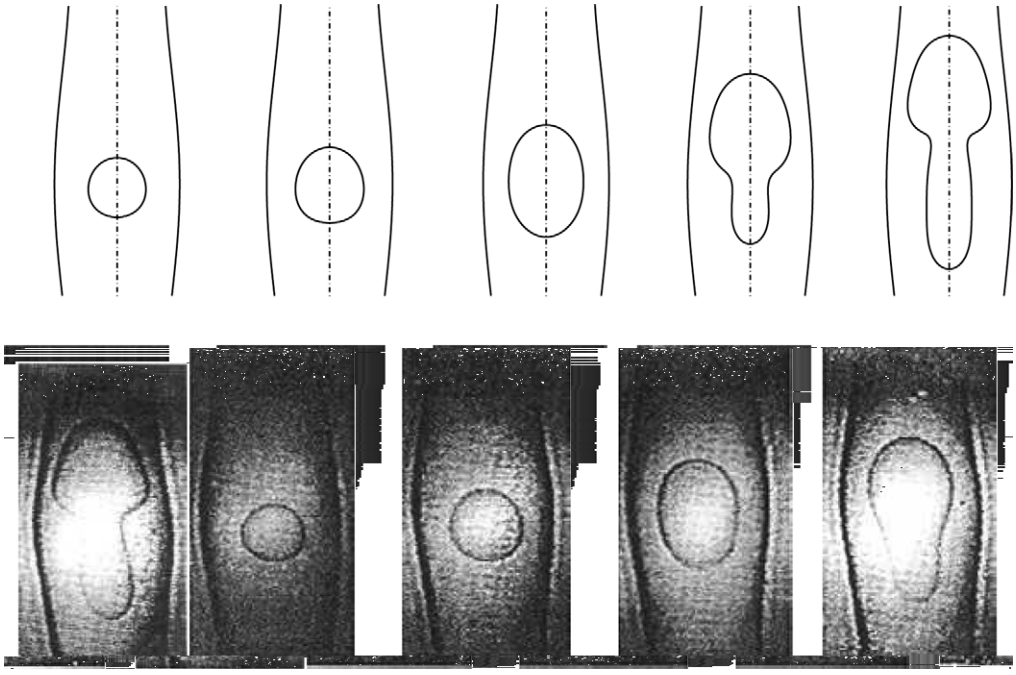


Fig. 13. Snapshots of the drops at the expansion of the constriction for the DEGG12 system for drop sizes (from left to right) $\kappa = 0.54$, 0.65, 0.78, 0.85 and 0.92, respectively. Top plots are the computational results and the lower plots are the experimental results of Hemmat and Borhan [9]. *Grid*: 32×1664, $\Delta t^* = 1.641$. Copyright (1996) from buoyancy-driven motion of drops and bubbles in a periodically constricted capillary by Hemmat and Borhan. Reproduced by permission of Taylor & Francis, Inc., http://www.taylorandfrancis.com.

the vicinity of the drop are plotted in Fig. 14 for a drop with $\kappa = 0.92$ while it passes through the throat and just after the throat to better show the overall quality of the solution.

Next the numerical accuracy and convergence of the method are examined. The numerical error can be decomposed into spatial and time-stepping errors. The spatial error results from the spatial discretization in the FV method and the interpolation and distribution schemes used in the front-tracking method due to finite size of mesh cells. The leading spatial error is expected to be globally second order accurate. The non-dimensional rise velocity of the drop centroid versus the non-dimensional axial length is plotted in Fig. 15 to show the overall dependence of the calculated results on grid refinement. The physical time step is taken as $\Delta t^* = 1.641$ and the residuals are reduced by three orders of magnitude in each FV inner iteration. The dashed vertical lines shown in Fig. 15 are drawn to mark the locations where the spatial error is quantified. To quantify the spatial error and to verify the expected second-order spatial accuracy, the rise velocity of the drop centroid is plotted against the inverse of the total number of grid cells M^{-2} in Fig. 16. The symbols

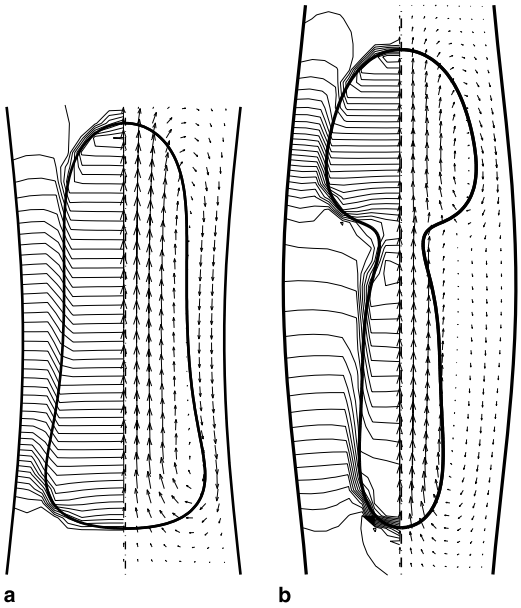


Fig. 14. Velocity vectors (right portion) and pressure contours (left portion) in the vicinity of the DEGG12 drop with $\kappa = 0.92$ while it passes through (a) the throat and (b) the expansion regions. *Grid*: 32×1664, $\Delta t^* = 1.641$.

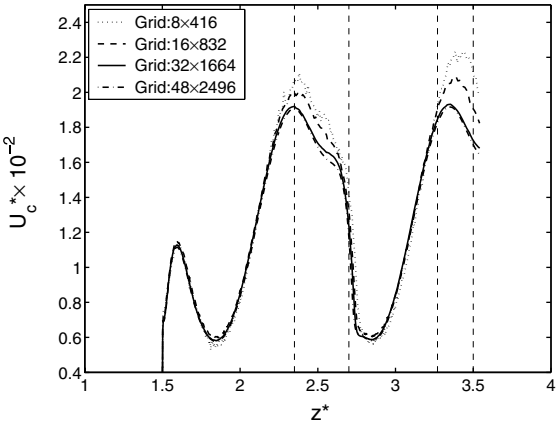


Fig. 15. The non-dimensional centroid velocity of a DEGG12 drop with $\kappa = 0.92$ plotted against the non-dimensional axial distance computed on 8×416, 16×832, 32×1664, and 48×2496 grids. $\Delta t^* = 1.641$.

indicate the numerical results and the lines are the linear least-squares fits to the numerical data. The approximate linear relationship between the rise velocity of drop centroid and M^{-2} in the asymptotic range confirms the expected second order accuracy of the method in space. To show the time-stepping error convergence, the non-dimensional rise velocity of the drop centroid is plotted against the non-dimensional time in Fig. 17 for various time steps. The computations are performed on a 32×1664 grid and the residuals are reduced by three orders of magnitude. As can be seen in this figure, the small difference between results obtained with different time steps indicates small time-stepping error in the present method. Note that larger time steps are found to result in instabilities on drop surface mainly due to explicit treatment of surface tension. The vertical dashed lines are again drawn to mark the locations where the time-stepping error is quantified. Fig. 18 shows the rise velocity of the drop centroid against Δt^2. The symbols are the numerical results and the solid lines are the linear least-squares fits to the numerical data. This figure shows that the slopes of the linear least-squares fits are small indicating the small time-stepping error in the simulations. Finally, the effects of the error tolerance are analyzed to determine the required reduction in averaged residuals in FV inner iterations. The error tolerance is denoted by ϵ_{res} and is defined as the reduction in average residuals in FV inner iterations. Fig. 19 shows the non-dimensional velocity of the drop centroid plotted against the non-dimensional time for various

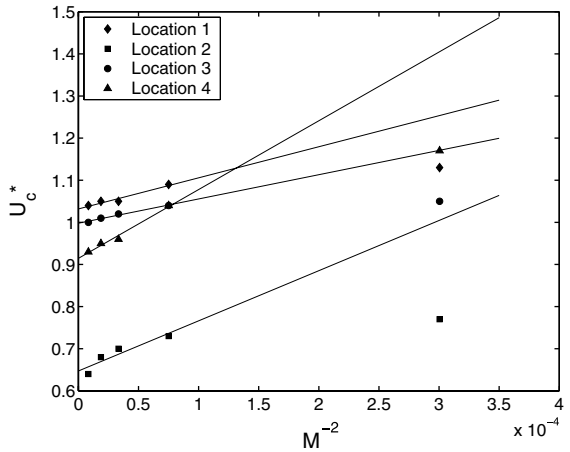


Fig. 16. The non-dimensional centroid velocities of a DEGG12 drop with $\kappa = 0.92$ against the inverse of the total number of grid cells M^{-2} at the selected locations showing the expected second order accuracy of the method. The solid lines are linear least-square fits to the numerical data. $\Delta t^* = 1.641$.

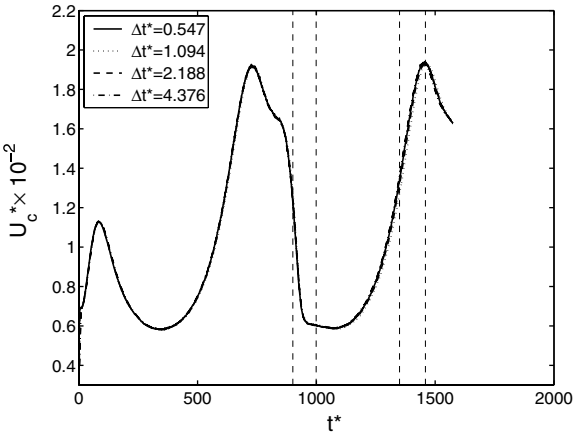


Fig. 17. The non-dimensional centroid velocity of a DEGG12 drop with $\kappa = 0.92$ plotted against the non-dimensional time computed with $\Delta t^* = 0.547$, 1.094, 2.188, and 4.376. *Grid*: 32×1664.

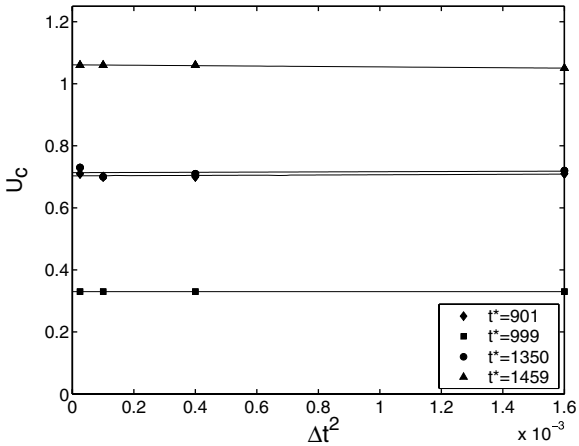


Fig. 18. The centroid velocities of a DEGG12 drop with $\kappa = 0.92$ against the time step Δt at the selected time frames. The solid lines are linear least-square fits to the numerical data. *Grid*: 32×1664.

error tolerances. It is clearly seen in this figure that $\epsilon_{\text{res}} = 10^{-3}$ is sufficient for this test case. Although not shown here, $\epsilon_{\text{res}} = 5 \times 10^{-3}$ is also found to be sufficient but $\epsilon_{\text{res}} = 10^{-3}$ is used in all the computations presented in this study.

Finally, the numerical results are compared with the experimental data. All the results are obtained on a 32×1664 grid, the physical time step is set to $\Delta t^* = 1.641$ and the error tolerance for sub-iterations is fixed at $\epsilon_{\text{res}} = 10^{-3}$. In order to qualitatively characterize the evolving shapes of drops as they pass through the corrugations, a deformation parameter denoted by D is defined as the ratio of the perimeter of the deformed drop profile to that of the equivalent spherical drop. The variations of the deformation parameter as a function of the axial position of the drop within one period of corrugation are plotted in Fig. 20 and are compared with the experimental data. It can be seen in this figure that the general trend for the deformation parameter is well captured by the present computations. The deformation is negligibly small for small drops, i.e., drops with $\kappa < 0.60$, and increases rapidly as the drop size gets larger. The discrepancy between the computed and the experimental results for D are partly attributed to the uncertainties in the experimental data and the inconsistency between the Bond number reported by Hemmat and Borhan [9] and the Bond number computed from the material properties and the average tube radius. Then the computed average rise velocities are compared with the experimental data. The non-dimensional average rise velocity U_m is plotted against the non-

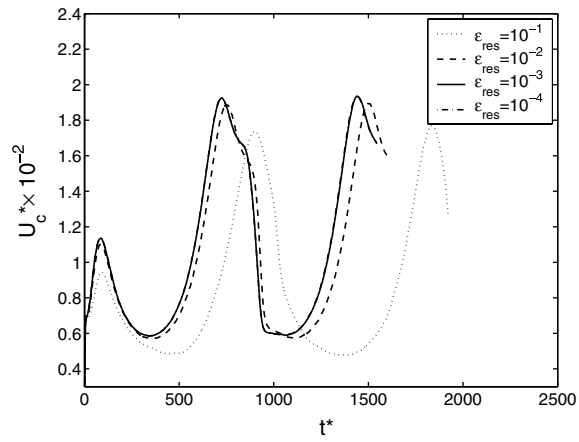


Fig. 19. The non-dimensional centroid velocity of a DEGG 12 drop with $\kappa = 0.92$ against the non-dimensional time computed with the FV inner iteration error tolerances of $\epsilon_{\text{res}} = 10^{-1}$, 10^{-2}, 10^{-3} and 10^{-4}. *Grid*: 32×1664 and $\Delta t^* = 1.641$.

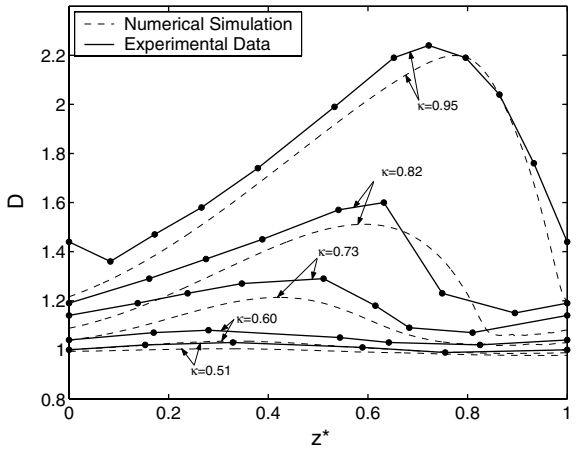


Fig. 20. The variations of the deformation parameter D with axial position of the advancing meniscus within one period of corrugation for DEGG12 system. The dashed curves are the numerical results and the solid lines are the experimental data.

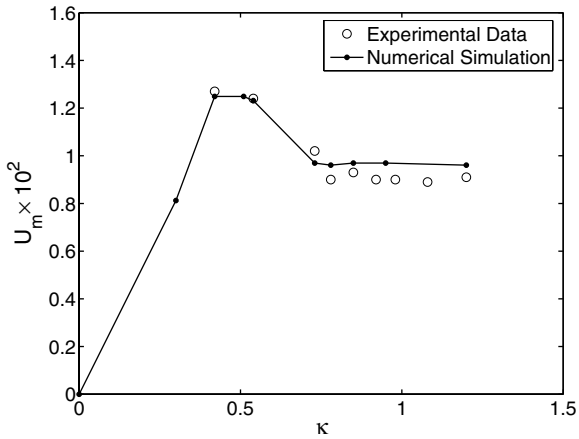


Fig. 21. Dimensionless average rise velocity as a function of drop size. The solid curve denotes the numerical results and the open symbols are the experimental data.

dimensional drop size κ in Fig. 21. The average rise velocity is defined as the average velocity of the drop centroid in the periodic motion. As can be seen in this figure, the numerical results are in a good agreement with the experimental data, i.e., the trend is well captured and the maximum error is less than a few percent for all the cases. In Fig. 21, the retardation effect of the constrictions is clearly seen and the non-dimensional velocity remains essentially constant for large drops, i.e., $\kappa > 0.90$.

5. Conclusions

An auxiliary grid method has been developed for computations of interfacial flows in complex geometries. The method combines a front-tracking method with a finite-volume algorithm and utilizes an auxiliary grid for computationally efficient tracking of interfaces in complex geometries. The flow equations are solved by a FV method on a body-fitted curvilinear grid and a separate Lagrangian grid is used to represent the interfaces between different phases.

The method is implemented to solve two-dimensional (planar or axisymmetric) interfacial flows. The numerical accuracy of the tracking algorithm has been assessed using simple settings of a two-dimensional drop set into motion by rigid body rotation of fluid and a drop placed in a sinusoidal channel in equilibrium

conditions. The tracking algorithm is found to be very robust and second order accurate both in time and space. The method is then applied to the motion of gravity-driven falling drops in a straight channel and buoyancy-driven rising drops in a continuously constricted channel. It is demonstrated that the method is convergent in terms of grid refinement, time-stepping error and residual reduction in the inner iteration of the FV method. The results computed by the present method are found to be in a good agreement with the FD/FT computations as well as with the available experimental data. It is demonstrated that the present method is a viable tool for accurate modeling of interfacial flows in complex geometries.

Acknowledgments

We thank Prof. G. Tryggvason for providing us with the finite-difference/front-tracking codes. We also thank Dr. S. Nas for useful discussions.

References

[1] A. Anderson, X. Zheng, V. Cristini, Adaptive unstructured volume meshing – I: the method, J. Comput. Phys. 208 (2) (2005) 616.
[2] R.M. Beam, R.F. Warming, An implicit factored scheme for the compressible Navier–Stokes equations, AIAA J. 16 (1978) 393.
[3] W.R. Briley, H. McDonald, Solution of the three-dimensional compressible Navier–Stokes equations by an implicit techniqueLecture Notes in Physics, vol. 35, Springer, New York, 1974, p. 105.
[4] D.A. Caughey, Diagonal implicit multigrid algorithm for the Euler equations, AIAA J. 26 (1988) 841.
[5] D.A. Caughey, Implicit multigrid computation of unsteady flows past cylinders of square cross-section, Comput. Fluids 30 (2001) 940.
[6] M. Dai, D.P. Schmidt, Adaptive tetrahedal meshing in free-surface flow, J. Comput. Phys. 208 (1) (2005) 228.
[7] L. Fauci, S. Gueron (Eds.), Computational Modeling in Biological Fluid Dynamics, Springer, New York, 2001.
[8] J. Han, G. Tryggvason, Secondary breakup of axisymmetric liquid drops: I. Acceleration by a constant body force, Phys. Fluids 11 (12) (1999) 3650.
[9] M. Hemmat, A. Borhan, Buoyancy-driven motion of drops and bubbles in a periodically constricted capillary, Chem. Eng. Commun. 150 (1996) 363.
[10] M. Muradoglu, P. Jenny, S.B. Pope, D.A. Caughey, A consistent hybrid finite-volume/particle method for the PDF equations of turbulent reactive flows, J. Comput. Phys. 154 (2) (1999) 342.
[11] M. Muradoglu, S. Gokaltun, Implicit multigrid computations of buoyant light drops through sinusoidal constrictions, J. Appl. Mech. 71 (2004) 1.
[12] M. Muradoglu, H.A. Stone, Mixing in a drop moving through a serpentine channel: a computational study, Phys. Fluids 17 (2005) 073305.
[13] W.L. Olbricht, L.G. Leal, The creeping motion of immiscible drops through a converging/diverging tube, J. Fluid Mech. 134 (1983) 329.
[14] W.L. Olbricht, Pore-scale prototypes of multiphase flow in porous media, Annu. Rev. Fluid Mech. 28 (1996) 187.
[15] U. Olgac, A.D. Kayaalp, M. Muradoglu, Buoyancy-driven motion and breakup of viscous drops in constricted capillaries, Int. J. Multi Flow (submitted).
[16] S. Osher, R.P. Fedkiw, Level set methods: an overview, J. Comput. Phys. 169 (2) (2001) 463.
[17] C. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220.
[18] S.B. Pope, PDF methods for turbulent reactive flows, Prog. Energy Combust. Sci. 11 (1985) 119.
[19] C. Pozrikidis (Ed.), Modeling and Simulation of Capsules and Biological Cells, Chapman & Hall/CRC Press, 2003.
[20] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annu. Rev. Fluid Mech. 31 (1999) 567.
[21] J.A. Sethian, P. Smereka, Level set methods for fluid interfaces, Annu. Rev. Fluid Mech. 35 (2003) 341.
[22] H.A. Stone, A.D. Stroock, A. Ajdari, Engineering flows in small devices: microfluidics toward lab-on-a-chip, Annu. Rev. Fluid Mech. 36 (2004) 381.
[23] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front-tracking method for the computations of multiphase flow, J. Comput. Phys. 169 (2) (2001) 708.
[24] H.S. Udaykumar, H.-C. Kan, W. Shyy, R. Tran-Son-Tay, Multiphase dynamics in arbitrary geometries on fixed Cartesian grids, J. Comput. Phys. 137 (1997) 366.
[25] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous incompressible multiphase flows, J. Comput. Phys. 100 (1992) 25.
[26] T. Yabe, F. Xiao, T. Utsumi, The constrained interpolation profile (CIP) method for multiphase analysis, J. Comput. Phys. 169 (2) (2001) 556.
[27] T. Ye, R. Mittal, H.S. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, J. Comput. Phys. 156 (1999) 209.
[28] X. Zheng, J. Lowengrub, A. Anderson, V. Cristini, Adaptive unstructured volume remeshing – II: application to two- and three-dimensional level-set simulations of multiphase flow, J. Comput. Phys. 208 (2) (2005) 626.