Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2007) 1428–1469
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A direct comparison between volume and surface tracking methods with a boundary-fitted coordinate transformation and third-order upwinding

Maria Zacharioudaki, Charalampos Kouris [1], Yannis Dimakopoulos, John Tsamopoulos *

Laboratory of Computational Fluid Dynamics, Department of Chemical Engineering, University of Patras, Patras 26500, Greece

Received 2 August 2006; received in revised form 3 September 2007; accepted 7 September 2007
Available online 15 September 2007

Abstract

A Volume Tracking (VT) and a Front Tracking (FT) algorithm are implemented and compared for locating the interface between two immiscible, incompressible, Newtonian fluids in a tube with a periodically varying, circular cross-section. Initially, the fluids are stationary and stratified in an axisymmetric arrangement so that one is around the axis of the tube (core fluid) and the other one surrounds it (annular fluid). A constant pressure gradient sets them in motion. With both VT and FT, a boundary-fitted coordinate transformation is applied and appropriate modifications are made to adopt either method in this geometry. The surface tension force is approximated using the continuous surface force method. All terms appearing in the continuity and momentum equations are approximated using centered finite differences in space and one-sided forward finite differences in time. In each time step, the incompressibility condition is enforced by a transformed Poisson equation, which is linear in pressure. This equation is solved by either direct LU decomposition or a Multigrid iterative solver. When the two fluids have the same density, the former method is about 3.5 times faster, but when they do not, the Multigrid solver is as much as 10 times faster than the LU decomposition. When the interface does not break and the Reynolds number remains small, the accuracy and rates of convergence of VT and FT are comparable. The well-known failure of centered finite differences arises as the Reynolds number increases and leads to non-physical oscillations in the interface and failure of both methods to converge with mesh refinement. These problems are resolved and computations with Reynolds as large as 500 converged by approximating the convective terms in the momentum equations by third-order upwind differences using Lagrangian Polynomials. When the volume of the core fluid or the Weber number decrease, increasing the importance of interfacial tension and leading to breakup of the interface forming a drop of core fluid, the FT method converges faster with mesh refinement than the VT method and upwinding may be required. Finally, examining the generation of spurious currents around a stationary "bubble" in the tube for Ohnesorge numbers between 0.1 and 10 it is found that the maximum velocity remains approximately the same in spite mesh refinements when VT is applied, whereas it is of the same order of magnitude for the coarsest mesh and monotonically decreases with mesh refinement when FT is applied.

* Corresponding author. Tel.: +30 2610 997 203; fax: +30 2610 996 178.
E-mail addresses: kourisc@lafarge.com (C. Kouris), tsamo@chemeng.upatras.gr (J. Tsamopoulos).
[1] Current address: Heracles Group of Companies, Halkis Plant, 34100, Greece.

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.09.004

Keywords: Multiphase flow; Volume tracking; Front tracking; Upwinding; Multigrid

1. Introduction

Flows of two or more immiscible fluids exhibit numerous phenomena with intrinsic scientific interest and important practical applications. Such applications from the oil and plastics industries include the production of oil from the ground (flow with water or steam in porous media), its water-facilitated transportation through pipelines, its chemical transformation in packed-bed reactors (flow with gaseous reactants in porous media) and the production of bicomponent and composite plastics and layered films (flow inside and out of channels). The overall applicability and performance of operations involving flows of two fluids are strongly dependent on their interface shapes. So, we must develop an understanding of the effects of the bulk properties and flow conditions on the interface evolution. In turn, a difficulty in developing reliable theoretical models for flows involving more than one fluid lies in the non-uniqueness of solutions, which is intimately related to the interface shape and its evolution. This strong coupling between the interface deformation and bulk motion necessitates computing them simultaneously. Our motivation for studying the two-phase flow in a circular tube with sinusoidal variation in its cross-section is that this is the simplest of the shearing flows of two fluids, which can adequately approximate the flow that takes place in a packed-bed reactor and, more generally in porous media. This model geometry of the tube wall has been used extensively for simulating single-phase flow of both Newtonian [1,2] and non-Newtonian fluids [3] and, although it cannot describe latitudinal dispersion in an actual packed-bed or exchange of fluids between nearby conduits, it simulates the converging–diverging character of the flow. In such reactors inertial, viscous and capillary forces are all comparable in magnitude and none can be neglected *a priori*. The fundamental mechanisms that give rise to the different flow regimes are rooted at the scale of the pore, where the two-phases flow and compete for the available space. Among the flow regimes that have been observed experimentally in a packed-bed reactor are: (a) The trickling regime, in which both phases are continuous, (b) The pulsing regime, in which one phase forms large discontinuous slugs within the other one which remains continuous, and (c) The spray and the bubbling regimes, in which the dispersed phase forms drops or bubbles much smaller than the typical constriction, which are surrounded by the continuous phase.

The numerical simulations of two-phase flows involve the solution of the Navier–Stokes equations for the two fluids with specified boundary and interfacial conditions. Several approaches to solve these moving interface problems are available in the literature, which have their own advantages and disadvantages. In particular, the free surface can be dealt with either an Eulerian or a Lagrangian numerical approach. The Eulerian approach involves the use of a discretization mesh that remains fixed in space with both fluids moving through it, while in the Lagrangian approach the mesh is convected by the flow in an effort to follow the moving interface and, thus, only simple nonintersecting interfaces can be represented due to limitations on the amount of mesh distortion allowed. Most finite element as well as spectral methods are developed around a Lagrangian approach and have been used successfully in calculating time dependent solutions of both single [4] and two-phase flows [5–8] in which free surfaces are involved. More recently, we advanced a mesh generation method based on solving a set of elliptic differential equations that can follow very large interface distortions, but still additional programming effort is needed to allow for interface breaking and merging [9,10]. Because we are further interested in readily predicting the above mentioned flow regimes and how they evolve, using either FEM or spectral methods in combination with a Lagrangian approach is precluded, since they cannot, at the present time, model such complex breaking and reforming interfaces [11].

In this paper, we utilize two basic Eulerian approaches for computing the evolving interface, namely a Volume Tracking (VT) which captures the interface after calculating the velocity field at each time step and a Front Tracking (FT) method, which explicitly advances the interface by following the motion of Lagrangian particles placed on it, [11]. In both methods the bulk fluids are treated as a single fluid properly accounting for the different fluid properties in each phase and the entire domain is partitioned into stationary computational cells irrespective of the changing topology of each phase, while interfacial tension is incorporated as a body force in the momentum balances multiplied by a delta function. In VT a marker function is introduced in

the entire domain to identify the fluid/fluid interface, while in FT a string of particles is used to follow the moving front. More specifically, in VT methods [12–20] the marker function indicates the volume fraction that one of the two fluids occupies in each computational cell, in our case this is the volume fraction of the core fluid in the initial core-annular arrangement. Although given the shape of the interface, the computation of the above function is straightforward, the inverse problem, i.e. the interface reconstruction given the values of the volume function is not a trivial matter and different methodologies of varying complexity and accuracy in two and three dimensions have been developed and applied. In increasing accuracy, the interface was assumed to be made up of a sequence of segments aligned with the grid, being either piecewise constant (SLIC) or constant ''stair-stepped'' (VOF) or piecewise linear segments (PLIC) or parabolic ones, an example of which is the PROST method [17–24]. Having reconstructed the interface it must be advected employing either a Lagrangian or a Eulerian scheme [17,19,20]. It is worth noting that the operations of interface reconstruction and advection are crucial in determining the overall accuracy of the computations. Furthermore, the application of the interfacial boundary conditions is not straightforward [25] and various approaches have been proposed, such as the continuous surface force (CSF) and the continuous surface stress (CSS) methods [22,25–27].

In Front Tracking (FT), the interface is explicitly discretized through a separate moving (Lagrangian) mesh, which moves with it as it deforms [28–36]. Front tracking is somewhat more complex to implement than VT, but gives the precise location and geometry of the interface with a continuous interelement approximation. An incentive for using an FT method is that it does not suffer from disadvantages that result from a non-smooth or discontinuous surface, like most VT methods. In principle, this gives a more accurate approximation of surface-related forces, like surface tension. Here the interfacial tension force is computed on the interfacial mesh and is transferred to the grid points by carefully constructed functions [28–30,34]. The variation of the fluid properties around the interface is computed using the same ''transfer'' function. Alternatively, fluid properties can be computed after calculating the volume fraction of each fluid in the cells containing the interface in a manner similar to VT methods, [33]. A striking feature of most Eulerian methods is the so-called spurious currents, which create oscillations strong enough to destroy the interface if it is stationary or convection is weak enough. These numerical artifacts result from the difficulty in approximating the delta function and its derivatives while computing the surface tension terms or the associated pressure jump and a number of remedies for this have been proposed [23,33]. The major incentive for using the VT method is that it allows for the description of highly complex interfaces such as those encountered in multiphase flows similarly to interface tracking methods, but at the same time it can readily handle topological changes. For example, in surface tracking methods break up of the interface cannot automatically take place, but must be allowed under some prespecified, presently purely numerical, conditions requiring additional programming effort. On the other hand, most VT studies allow the interface to be discontinuous between adjacent cells and generally achieve lower accuracy for the same discretization mesh.

For the numerical solution, we discretize all equations in space using conventional centered finite differences on a staggered mesh. The time integration is performed using the explicit Euler method. However, the approximation of the convective terms by central finite differences leads to numerical instability at high enough Reynolds numbers, which is manifested by the development of non-physical oscillations on the interface and failure to obtain a converged solution with mesh refinement. It is also well known that first-order upwinding introduces artificial viscosity resulting in overdamped (inaccurate) solutions, whereas second-order upwinding yields even larger oscillations. Thus it has been recognized that the most appropriate choice is third-order upwind approximations. Among such discretizations of the convective terms, we adopt the scheme developed by Tabata and Fujima [37] for finite element solutions of single phase flows at high Reynolds numbers. The basic idea of this upwind technique is properly choosing the locations of two upwind and two downwind points from the location where the momentum balances are written and along the local flow direction. Then a third-order approximation to the first derivative along this line is written based on Lagrange interpolation polynomials including a parameter to control the level of upwinding. Two choices of this parameter have been used and evaluated herein [37,38]. We enforce fluid incompressibility by solving a transformed Poison equation for the pressure by either a direct LU decomposition or an iterative multigrid method [39].

Interfacial forces are introduced into the equations of motion as body forces acting on fluid elements lying within a small transition region around the surface, instead of imposing them as boundary conditions on a discontinuous surface. This is the continuous-surface-force (CSF) method [25]. Moreover, boundary condi-

tions in both the radial and axial directions should be specified. In the axial direction we impose that the flow field is periodic while in the radial direction the flow field is bounded at the axis of symmetry and at the solid wall no-slip, no-penetration conditions are enforced. In order to accurately impose these boundary conditions a body-fitted coordinate transformation is employed. Because of this transformation an orthogonal mesh is generated in the transformed domain allowing direct application of finite differences, although it corresponds to a mesh in the physical domain that is neither orthogonal nor conformal. On the other hand, extensive modifications to algorithms for the interface calculation and advection must be introduced. The governing equations have been integrated for sufficiently long times so that the solution reaches a stable limit cycle, when this is possible.

The remaining of this paper is organized as follows: The dimensionless equations and boundary conditions as well as the body fitted coordinate transformation are presented in the following section. The discretization schemes used to solve the Navier–Stokes equations are presented in Section 3. Sections 4 and 5 give a short description of the VT and FT algorithms, respectively, with emphasis on the modifications and improvements we introduced herein. The solution methods of the transformed Poisson equation are discussed in Section 6. Section 7 presents selective results to demonstrate the numerical accuracy, convergence and efficiency of our calculations and a comparison of all the methods. In the final section conclusions are drawn.

2. Problem definition and domain transformation

We study the flow of two incompressible, immiscible and Newtonian fluids in a pipe with periodically varying and circular cross-section, initially having an axisymmetric core-annular arrangement. The first fluid with viscosity and density $(\hat{\mu}_1, \hat{\rho}_1)$ occupies the core region, the second fluid with properties $(\hat{\mu}_2, \hat{\rho}_2)$ occupies the surrounding annulus, while the density and the viscosity of either phase are scaled using the physical properties of the core fluid. $\widehat{T}$ denotes the interfacial tension between the two fluids. The circular tube has an inner radius, $\widehat{R}_2$, which varies sinusoidally with spatial period $\widehat{L}$ between its maximum, $\widehat{R}_{\max}$, and its minimum value, $\widehat{R}_{\min}$. The symbol (^) denotes dimensional quantities. Using as characteristic length scale in both the radial and axial directions the maximum radius of the tube, the axial variation of the solid wall is described by the following dimensionless equation:

$$R_2(z) = \frac{1+\alpha}{2} + \frac{1-\alpha}{2}\cos\left(2\pi\frac{z}{\Lambda}\right), \quad 0 \leqslant z \leqslant \Lambda \tag{1}$$

In Eq. (1) α denotes the constriction ratio, $\alpha = \widehat{R}_{\min}/\widehat{R}_{\max}$; Λ denotes the aspect ratio of the geometry and $\Lambda = \widehat{L}/\widehat{R}_{\max}$. The two-phase flow is driven by an externally imposed pressure gradient, $\hat{f}$, which remains constant and is used to obtain other characteristic scales of our system. The velocity field is scaled by $(\hat{f}\widehat{R}_{\max}/\hat{\rho}_1)^{1/2}$, the time variable by $(\hat{\rho}_1\widehat{R}_{\max}/\hat{f})^{1/2}$ and the pressure as well as the viscous stresses by $(\widehat{R}_{\max}\hat{f})$. The mass and momentum balances for the entire flow domain in dimensionless form are:

$$\nabla \cdot \underline{U} = 0 \tag{2}$$

$$\rho\partial_t(\underline{U}) + \rho\nabla\cdot(\underline{U}\underline{U}) = -\nabla P + \frac{1}{Re}\nabla\cdot(2\mu\underline{\underline{D}}) + \frac{\kappa}{We}\tilde{\underline{n}}\frac{\delta(\underline{r}-\underline{R}_f)}{r} \tag{3}$$

where $\underline{U} = U\underline{e}_r + 0\underline{e}_\theta + W\underline{e}_z$ denotes the axisymmetric velocity vector; $\underline{e}_r$, $\underline{e}_\theta$, $\underline{e}_z$ are the unit normal vectors associated with the cylindrical coordinate system (r,θ,z), $\underline{r} = r\underline{e}_r + z\underline{e}_z$ the position vector, $\underline{R}_f$ its value on the interface, t denotes the time variable and $\rho = \hat{\rho}_i/\hat{\rho}_1$ and $\mu = \hat{\mu}_i/\hat{\mu}_1$, where $i = 1, 2$. Generally, we will indicate all dependent variables on the interface by a subscript f and its coordinates by capital letters, whereas the coordinates in the bulk by lower case letters. The gradient operator is expressed in the cylindrical coordinate system and the rate of deformation tensor $\underline{\underline{D}}$ in the Newtonian case is defined as $\underline{\underline{D}} = ((\nabla\underline{U}) + (\nabla\underline{U})^{\mathrm{T}})/2$. It is worth noting that the effect of interfacial tension has been incorporated into the momentum equation according to the CSF method [25] using the Dirac delta function, $\delta(\underline{r} - \underline{R}_f)/r$. This function is zero everywhere within either fluid except for the fluid/fluid interface (front), which is defined by $\underline{r} = \underline{R}_f$; the unit vector normal to the interface is denoted by $\tilde{\underline{n}}$ and κ is the curvature of the interface. The dimensionless numbers that appear in Eq. (3) are the Reynolds, Re, and the Weber, We, which are defined by

$$Re = \frac{\widehat{R}_{\max}(\hat{\rho}_1 \hat{f} \widehat{R}_{\max})^{1/2}}{\hat{\mu}_1}, \quad We^{-1} = \frac{\widehat{T}}{\hat{f}\widehat{R}^2_{\max}} \tag{4}$$

Given that the flow is induced by a constant pressure gradient, it is convenient to decompose the dimensionless pressure field, P into a periodic part, p, and a term that varies linearly in the axial distance, i.e.:

$$P = p - z \tag{5}$$

In order (i) to impose correctly the appropriate boundary conditions on the momentum equation, Eq. (3), at the centerline as well as at the solid wall and (ii) to generate an orthogonal mesh to implement finite differences readily, the following body-fitted coordinate transformation is employed:

$$x = \frac{r}{R_2(z)}, \quad y = z, \quad \tau = t \tag{6}$$

This transformation maps the solid surface, which is described in the cylindrical coordinates by the equation $r = R_2(z)$, to the $x = 1$ plane; the centerline $r = 0$ to the $x = 0$ plane, while the other transformed variables (y,τ) coincide with the original axial distance and the time variable (z,t), respectively. Therefore the bounds of the new variables are:

$$0 \leqslant x \leqslant 1, \quad 0 \leqslant y \leqslant \Lambda \quad \text{and} \quad \tau \geqslant 0 \tag{7}$$

Introducing the transformations given in Eqs. (5) and (6) into Eqs. (2),(3), and using the chain rule to transform the derivatives from the physical to the computational domain we obtain the following equations, expressed in conservative form:

$$R_2 \partial_x(x(U - xR_2'W)) + x\partial_y(R_2^2 W) = 0 \tag{8}$$

$$\begin{aligned}\partial_\tau(U) = &-\frac{\partial_x(xU^2)}{xR_2} + \frac{R_2'\partial_x(xUW)}{R_2} - \frac{\partial_y(R_2UW)}{R_2} \\ &- \frac{\partial_x(p)}{\rho R_2} - \frac{\tau_{\theta\theta}}{\rho Re x R_2} + \frac{\partial_x(x\tau_{rr})}{\rho Re x R_2} - \frac{R_2'\partial_x(x\tau_{rz})}{\rho Re R_2} + \frac{\partial_y(R_2\tau_{rz})}{\rho Re R_2} + \frac{\kappa\delta(\underline{r} - \underline{R}_f)\tilde{\underline{n}} \cdot \underline{e}_r}{xR_2 We\rho}\end{aligned} \tag{9}$$

$$\begin{aligned}\partial_\tau(W) = &-\frac{\partial_x(xUW)}{xR_2} + \frac{R_2'\partial_x(xW^2)}{R_2} - \frac{\partial_y(R_2W^2)}{R_2} + \frac{R_2'\partial_x(xp)}{\rho R_2} - \frac{\partial_y(R_2p)}{\rho R_2} + \frac{\partial_x(x\tau_{rz})}{\rho Re x R_2} \\ &- \frac{R_2'\partial_x(x\tau_{zz})}{\rho Re R_2} + \frac{\partial_y(R_2\tau_{zz})}{\rho Re R_2} + \frac{\kappa\delta(\underline{r} - \underline{R}_f)\tilde{\underline{n}} \cdot \underline{e}_z}{xR_2 We\rho} + \frac{1}{\rho}\end{aligned} \tag{10}$$

where

$$\begin{aligned}&\tau_{\theta\theta} = \frac{2\mu U}{R_2 x}, \quad \tau_{rr} = \frac{2\mu\partial_x(U)}{R_2}, \quad \tau_{zz} = \frac{2\mu(\partial_y(R_2W) - R_2'\partial_x(xW))}{R_2}, \\ &\tau_{rz} = \frac{\mu(\partial_x(W) - R_2'\partial_x(xU) + \partial_y(R_2U))}{R_2}\end{aligned} \tag{11}$$

Eq. (8) is the transformed equation of continuity for incompressible fluids, Eqs. (9) and (10) are the transformed radial and axial components of the momentum equation, while Eq. (11) give the transformed stress components for Newtonian fluids. The scalar quantity R_2', which appears in the above equations, denotes the derivative of the solid surface with respect to the axial distance, $R_2' = \partial_y(R_2)$, and it is computed at axial distances located at the center between two consecutive coordinate nodes that form each computational cell. The values of the function itself are computed at the coordinate nodes according to Eq. (1).

The above partial differential equations are solved subject to no-slip and no-penetration conditions at the solid wall and that the velocity field remains bounded at the axis of symmetry, while in the axial direction we impose periodicity of both the velocity field and its derivative with respect to the axial distance.

$$U|_{x=0} = \partial_x(W)|_{x=0} = U|_{x=1} = W|_{x=1} = 0 \tag{12}$$

$$U|_{y=0} - U|_{y=\Lambda} = \partial_y(U)|_{y=0} - \partial_y(U)|_{y=\Lambda} = W|_{y=0} - W|_{y=\Lambda} = \partial_y(W)|_{y=0} - \partial_y(W)|_{y=\Lambda} = 0 \tag{13}$$

3. Centered finite differences and third-order upwinding

The mesh that results in physical space (r,z) by partitioning the computational domain (x,y) into equal orthogonal quadrilaterals of sides Δx, Δy in the x and y directions, respectively, is shown in Fig. 1. In the same figure, we give the points at which each variable is computed as well as the coordinate nodes that form each computational cell according to the marker-and-cell (MAC) method [40]. In particular, the radial, U, and axial, W, velocities are computed at the centers of the horizontal and vertical sides of each cell, respectively, whereas the pressure, p, and volume fraction, F, which will be introduced in Section 4, are computed at the centers of each cell. Each cell is indicated by the pair (i,j), characterizing its center in the x, y directions, respectively, while the coordinate nodes are translated by 1/2. It is worth mentioning that although the solid surface is a sinusoidal function in the axial distance, in our VT code only, it is also approximated by using piecewise continuous and linear segments and thus, it is computed with a second-order accuracy consistently with the fluid/fluid interface (see Section 4.1). Apparently, the resulting mesh is neither orthogonal nor conformal in physical space. In the above equations, all derivatives in space are approximated using second-order, centered finite differences, while the time derivatives are approximated using one-sided forward finite differences (explicit Euler method). The boundary conditions given in Eqs. (12) and (13) are imposed using as many rows of fictitious boundary nodes as needed on each boundary and therefore, our Navier–Stokes solver exhibits formal accuracy of order $O(\Delta x^2, \Delta y^2)$ in the x,y spatial directions, respectively and $O(\Delta\tau)$ in time.

For higher Reynolds numbers, of order 200 and above, we noticed that central differencing of the convective terms leads to non-physical oscillations on the interface and the numerical solution fails to converge with mesh refinement. It has been demonstrated that this problem can be resolved using upwinding, i.e. asymmetrically favoring upstream nodes while approximating the derivatives in the convective terms. However, first-order upwinding is overly diffusive, second and fourth-order upwiding exhibit increased oscillations, whereas fifth-order upwinding is significantly more complex, making third-order upwinding an optimal choice [41]. Therefore, in order to resolve this problem we use a third-order upwind scheme and in particular the one developed for finite element methods by Tabata and Fujima [37], henceforth referred to as TF. These authors demonstrated that their third-order scheme is very sensitive to increases in the Reynolds number (i.e. no excessive overdamping is generated) by comparing their results to a mesh with 10–40 times more nodes without upwiding. According to this scheme, we approximate the convective terms by taking into consideration the flow direction. This is a non-trivial procedure because the mesh in physical space is not orthogonal and we need to find the intersections of the local velocity vector with the sides of triangles generated from the mesh. In particular each rectangle of the original mesh is divided into 6 triangles by connecting the mid points of two consecutive sides with each other and the mid points of its upper and lower side with each other. The level of upwinding is controlled by the non-negative parameter β. For $\beta = 0$ we recover a fourth-order central difference scheme, whereas for $\beta > 0$ different upwinding

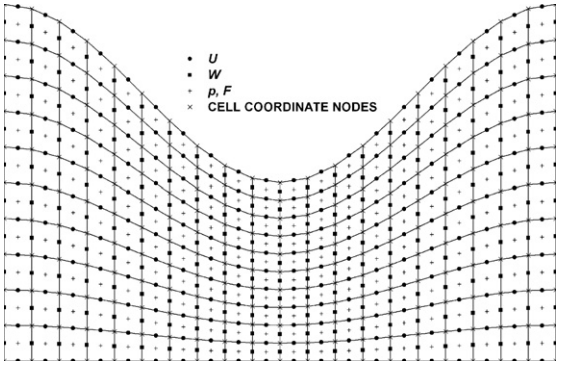

Fig. 1. Coarse mesh layout in the physical domain that results by partitioning the computational domain into $L = 10$ radial and $M = 20$ axial quadrilaterals. The position at which each dependent variable (U, W, p, F) is computed is indicated on the figure.

schemes are generated. We have tested the above algorithm with two values of β; the first one was proposed by Singh and Leal [38], henceforth mentioned as SL, after trial and error tests, whereas the second one was proposed by TF.

The strong, but at the same time weak point of the VT method is that the boundary conditions are not applied directly at the liquid/liquid interface. Therefore, the good resolution of the boundary conditions depends strongly on the quality of the mesh, especially in the vicinity of the fluid/fluid interface. However, the use of a fine grid imposes a restriction on the time step size for an explicit method in order to generate a numerically stable integration scheme. This restriction is that the time step must be less than the time scale for convective transport of a fluid particle through a cell (which is proportional to the grid size) and also the time scale of vorticity diffusion through a cell (which is proportional to the square of the grid size). Between the two stability criteria, the latter is much more restrictive than the former since it depends on the square of the grid size. Hence with any explicit method, simulations with a very low Reynolds number are subject to very strict stability limitations on the size of the time step and thus, are very expensive. For this reason in our multigrid solver (see Section 6.2) during time-integration of the radial (axial) momentum equation all stress components related to the radial (axial) velocity vectors are treated implicitly. Finally, both radial and axial velocities are checked at each time step as to whether there is need for upwinding according to the former criterion. This need arises as Re increases and then the following inequalities are violated:

$$W\Delta y Re\rho/\mu \leqslant 1.5 \tag{14}$$

$$(U - xR_2'W)R_2\Delta x Re\rho/\mu \leqslant 1.5 \tag{15}$$

4. The Volume Tracking method (VT)

In VT an additional variable denoted by F, $0 \leqslant F \leqslant 1$, is introduced, often called color or marker function. This function is associated with the volume fraction that one of the two fluids occupies in each computational cell, the core fluid in our case. As a result, $F = 1$, when the cell is completely filled with the core fluid and $F = 0$, when it is completely filled with the annular one. In fact, the marker function makes a smooth transition from zero to one in a thin region commensurate with the mesh size. The volume fraction of the core (and the annular) fluid is conserved and thus, the time evolution of F for incompressible materials is governed by the following equation:

$$\partial_t(F) + \nabla \cdot (F\underline{U}) = 0 \tag{16}$$

In the transition zone, the fluid properties change smoothly from their value on one side of the interface to their value on the other side. This interface thickness is a function of the mesh size and does not change during the calculations; therefore no additional smearing of the interface takes place. Since μ and ρ are constant within each fluid, the volume function, F, allows us to evaluate them at each computational cell using the following relationships:

$$\rho = F + (1 - F)\rho^{o}, \quad \mu = F + (1 - F)\mu^{o} \tag{17}$$

where ρ^{o} and μ^{o} are the density and viscosity ratios of the annular to the core fluid and, thus, when $F = 1(0)$, then $\rho = 1(\rho^{o})$ and $\mu = 1(\mu^{o})$, as they should. So, when a fluid property changes across the interface, we change the grid values of this property every time the interface moves using Eq. (17). Since the typical meshes we used are fine enough, we did not find advantageous to approximate the jump of these properties better, following the more complicated procedure described in [15]. The basic steps of our implementation of the VT method are discussed in detail in the following.

4.1. Interface reconstruction

The scalar function $F = F(x, y, \tau)$ enables us to deal with fluid/fluid interfaces of arbitrarily complex shape and continue the simulation past the point of pinch-off and the formation of drops or drop coalescence. This becomes feasible since the interface does not necessarily coincide with the sides of the computational cells and thus, mesh distortion is avoided. However, the marker function by itself locates only the cells intersected by the

interface and the interface needs to be reconstructed in each such cell. In order to minimize as much as possible the smearing of the sharp front between the two phases, the interface is reconstructed at each time step before being advanced in time. Earlier investigators developed such schemes which, unfortunately, are not accurate enough and may introduce submesh-size material of one phase in the other phase [20]. For the same reasons, spurious oscillations accompanied with severe distortion of the interface, which assumes a "step" shape, have been reported [15,16]. In their particular case, the problem has been resolved by introducing a moving with the velocity of the undisturbed interface coordinate system and a higher-order velocity interpolation scheme.

In order to avoid such problems we implemented the following PLIC-type method, which is multidimensional and second-order accurate [19,20]. In particular, we reconstruct the interface at a typical computational cell (i,j), provided that $0 < F_{i,j} < 1$ in it. However, an Eulerian scheme does not preserve this constraint on the marker function and we need to treat cells even with $F \leqslant \varepsilon$ *or* $F \geqslant 1-\varepsilon$ where $\varepsilon = 0.5 \times 10^{-8}$ as cells not intersected by the interface. In each of the remaining cells, the interface is approximated by a straight line expressed in the transformed variables as

$$G(x,y) \equiv Ax + By + C = 0 \tag{18}$$

and, thus, the three parameters of the equation of the line should be computed. Having computed the values of the volume function at the center of each computational cell, $F_{i,j}$, we first interpolate these values at the four corners of each cell. Then we use the interpolated values of F to compute the parameters A, B of the equation of the line according to the following relationships:

$$A = \partial_x(F) = \frac{F_{i+1/2,j+1/2} + F_{i+1/2,j-1/2} - F_{i-1/2,j+1/2} - F_{i-1/2,j-1/2}}{2\Delta x} \tag{19}$$

$$B = \partial_y(F) = \frac{F_{i+1/2,j+1/2} + F_{i-1/2,j+1/2} - F_{i+1/2,j-1/2} - F_{i-1/2,j-1/2}}{2\Delta y} \tag{20}$$

It is worth mentioning that although Eq. (18) describes the interface with a simple linear relationship in the transformed variables (x,y), this is not the case in the physical domain, since its dependence on the (r,z) variables becomes nonlinear given the transformation of Eq. (6).

The undetermined constant C of Eq. (18) is adjusted so that the resulting line that passes through the cell divides it in two sub-volumes while the ratio of the core fluid to the total cell volume equals the value of the volume function at this particular cell, $F_{i,j}$. Which of the two sub-volumes corresponds to the core fluid can be inferred by examining the sign of the function $G(x,y)$. By the definition of the constants A and B, the sub-volume that corresponds to the core fluid is the one the internal points of which make this function positive. Then, C is the root of the equation:

$$g(C) = V(C) - F_{i,j} = 0 \tag{21}$$

where $V = V(C)$ denotes the ratio of the volume of the core fluid to the total volume of the cell, $V_{i,j}$, for the previously determined values of the parameters A and B. For a given value of the parameter C the equation of the line, Eq. (18), is known and thus we can identify the points at which it crosses the four sides that form the computational cell. A line that crosses a quadrilateral divides it into two smaller sub-volumes, which can be either two smaller quadrilaterals or two triangles or one triangle and a pentangle. After identifying the case at hand, we compute both the total volume of the cell as well as the volume of one of the sub-cells, using the appropriate four-point Gaussian quadrature. After that, the value of the function $g = g(C)$ can be computed straightforwardly using Eq. (21). Of course, the dependence of the function g on the parameter C is nonlinear and thus an iterative method should be employed. We prefer finding the zero of the above equation using Brent's method [20,42], which invokes a combination of bisection and inverse quadratic interpolation, since it involves only computations of g for different values of C and its convergence is guaranteed on condition that we provide both an upper (C_1) and a lower (C_2) bound for parameter C such that $g(C_1)g(C_2) < 0$. Of course, its convergence rate is somewhat slower than Newton's method, but the convergence of the latter method is not guaranteed for any initial guess in addition to the fact that the sensitivity of g with respect to C is also needed. The error associated with this interface approximation is $O(\kappa\Delta x^2, \kappa\Delta y^2)$ where κ is the local curvature of the interface [20].

4.2. Advection of volume fraction

A number of ways have been proposed to advect the F function. The original SOLA-VOF algorithm [43] employed a donor-acceptor scheme around each cell in the mesh, but this has been found [15] to be of low-accuracy and should not be used. Renardy and co-workers [16] have advocated an alternative operator-split method. Instead of this, we perform an accurate multidimensional, time-integration of Eq. (16) assuming incompressible flow, whereby the core fluid volume fractions at discrete time level (n) are marched forward in time to level ($n+1$). Eq. (16) is reformulated as follows in the transformed variables:

$$\partial_\tau(xR_2^2F) + \partial_x(xR_2F(U - xR_2'W)) + \partial_y(xR_2^2FW) = 0 \tag{22}$$

The first term in the left-hand side of Eq. (22) accounts for the change of F in time, while the second and the third terms account for the volumes of the core fluid that enter and/or exit the cell through its four sides. Given the transformation that we have performed, the total volume of each computational cell equals:

$$V_{i,j} = 2\pi \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} xR_2^2 \,\mathrm{d}y\,\mathrm{d}x \tag{23}$$

By integrating Eq. (22) over both the time interval $\Delta\tau$ and the volume of the computational cell (i,j) and after dividing the resulting equation by the total volume of the cell and using geometric arguments, we obtain the following equation:

$$F_{i,j}^{(n+1)} - F_{i,j}^{(n)} = -\frac{\Delta V_{i+1/2,j} - \Delta V_{i-1/2,j}}{V_{i,j}} - \frac{\Delta V_{i,j+1/2} - \Delta V_{i,j-1/2}}{V_{i,j}} \tag{24}$$

where $F_{i,j}^{(n)}, F_{i,j}^{(n+1)}$ are the values of the volume function in the cell (i,j) at the time level (n) and $(n+1)$, respectively, and $\Delta V_{i+1/2,j}$, $\Delta V_{i-1/2,j}$, $\Delta\, V_{i,j+1/2}$, $\Delta V_{i,j-1/2}$ are the core-volume fluxes across the four sides of the computational cell (i, j) within the time interval $\Delta\tau$ given by

$$\Delta V_{i+1/2,j} = \int_{\tau^{(n)}}^{\tau^{(n+1)}} 2\pi \int_{y_{j-1/2}}^{y_{j+1/2}} R_2(xF(U - xR_2'W))\Big|_{x=x_{i+1/2}} \mathrm{d}y\,\mathrm{d}\tau \tag{25}$$

$$\Delta V_{i,j+1/2} = \int_{\tau^{(n)}}^{\tau^{(n+1)}} 2\pi \int_{x_{i-1/2}}^{x_{i+1/2}} x(FR_2^2W)\Big|_{y=y_{j+1/2}} \mathrm{d}x\,\mathrm{d}\tau \tag{26}$$

It is worth interpreting the different terms that arise in Eqs. (25) and (26). The velocities $(U - xR_2'W)/\sqrt{1 + x^2R_2'^2}$ and W that appear in the above equations correspond to the velocities normal to the sides $x = x_{i+1/2}$, $y = y_{j+1/2}$, respectively, while the terms $2\pi xR_2\sqrt{1 + x^2R_2'^2}\,\mathrm{d}y$ and $2\pi xR_2^2\,\mathrm{d}x$ define the corresponding infinitesimal cross-sections. Given a typical computational cell (i,j), shown in Fig. 2, i.e. the quadrilateral ABCDEFGHIJA, the reconstructed interface in it is such that the core fluid occupies the region ABKMGHIJA. The normal velocities to the sides $x = x_{i+1/2}$, $y = y_{j+1/2}$ are pointing outwards of the cell and thus the core fluid is fluxed outside from this cell to its neighboring ones. The total volumes that are fluxed across the sides $x = x_{i+1/2}$, $y = y_{j+1/2}$ within the time interval $\Delta\tau$ are given by the following equations:

$$\begin{aligned}&\int_{\tau^{(n)}}^{\tau^{(n+1)}} 2\pi \int_{y_{j-1/2}}^{y_{j+1/2}} R_2(x(U - xR_2'W))\Big|_{x_{i+1/2}} \mathrm{d}y\,\mathrm{d}\tau \approx 2\pi\Delta\tau \int_{y_{j-1/2}}^{y_{j+1/2}} R_2(x(U - xR_2'W))\Big|_{x_{i+1/2}} \mathrm{d}y \\ &= 2\pi \int_{y_{j-1/2}}^{y_{j+1/2}} \int_{x^*}^{x_{i+1/2}} xR_2^2 \,\mathrm{d}x\,\mathrm{d}y\end{aligned} \tag{27}$$

$$\int_{\tau^{(n)}}^{\tau^{(n+1)}} 2\pi \int_{x_{i-1/2}}^{x_{i+1/2}} x(R_2^2W)\Big|_{y_{j+1/2}} \mathrm{d}x\,\mathrm{d}\tau \approx 2\pi\Delta\tau \int_{x_{i-1/2}}^{x_{i+1/2}} x(R_2^2W)\Big|_{y_{j+1/2}} \mathrm{d}x = 2\pi \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y^*}^{y_{j+1/2}} xR_2^2 \,\mathrm{d}y\,\mathrm{d}x \tag{28}$$

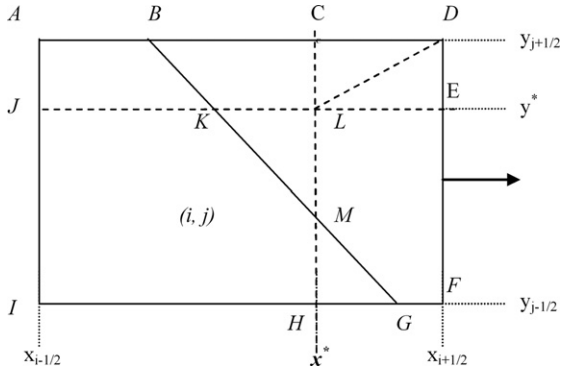

Fig. 2. Geometric representation for the development of the advection algorithm with VT.

Thus, the locations x^*, y^* that are depicted in Fig. 2 can be found explicitly for our geometry:

$$x^* = \begin{cases} D = 0, & \sqrt{x_{i+1/2}^2 - 2x_{i+1/2}\Delta\tau U_{i+1/2,j}} \\ D \neq 0, & \sqrt{x_{i+1/2}^2 - 3x_{i+1/2}\Delta\tau(U_{i+1/2,j} - x_{i+1/2}DW_{i+1/2,j})\frac{(Dy_{j+1/2}+E)^2-(Dy_{j-1/2}+E)^2}{(Dy_{j+1/2}+E)^3-(Dy_{j-1/2}+E)^3}} \end{cases} \tag{29}$$

$$y^* = \begin{cases} D = 0, & y_{j+1/2} - W_{i,j+1/2}\Delta\tau \\ D \neq 0, & \frac{\sqrt[3]{(Dy_{j+1/2}+E)^3 - 3D\Delta\tau W_{i,j+1/2}(Dy_{j+1/2}+E)^2} - E}{D} \end{cases} \tag{30}$$

The constants D and E arise in the linear approximation of the solid wall in the axial direction within this cell:

$$R_2 = Dy + E, \quad y_{j-1/2} \leqslant y \leqslant y_{j+1/2} \tag{31}$$

In order to derive these forms, it was necessary to approximate the tube boundary by linear segments, Eq. (31), which are of the same accuracy as those used to approximate the fluid/fluid interface. Since we are interested in computing the volume of only the core fluid and not the total volume that is fluxed outside of the computational cell what we must compute in this particular case are the volumes of the triangle HMGH and the volume of the trapezoid ABKJA, respectively. These volumes after being identified are computed using the appropriate Gauss formula as explained above.

The above method of computing the advected volume fraction is direct, but the fluid near the corners of each cell will be accounted for twice, for example, in Fig. 2 the fluid in the rectangle CDELC will exit the cell in both the x and y direction. In order to remedy this, we split this rectangle into two triangles by drawing the line LD and use an iterative method to compute x^*, y^*. The total volume that is fluxed across the sides $x = x_{i+1/2}$, $y = y_{j+1/2}$ within the time interval $\Delta\tau$ initially occupies the trapezoids DLMHGFD and JKLDBAJ, while the fluid exits through the same right and top boundary, respectively and with the same velocity, as before. Therefore, Eqs. (27) and (28) must be changed to:

$$\int\int_{DLMHGFD(x^*,y^*)} xR_2^2\,\mathrm{d}x\,\mathrm{d}y = \Delta\tau\int_{y_{j-1/2}}^{y_{j+1/2}} R_2(x(U - xR_2'W))\Big|_{x_{i+1/2}}^{}\mathrm{d}y = \Delta\tau\Delta y x R_2(U - xR_2'W)\big|_{x_{i+1/2}} \tag{32}$$

$$\int\int_{JKLDBAJ(x^*,y^*)} xR_2^2\,\mathrm{d}x\,\mathrm{d}y = \Delta\tau\int_{x_{i-1/2}}^{x_{i+1/2}} x(R_2^2W)\Big|_{y_{j+1/2}}^{}\mathrm{d}x = \Delta\tau\Delta x x R_2^2 W\big|_{y_{j+1/2}} \tag{33}$$

Of course, the fluxes in the right-hand sides of Eqs. (32) and (33) have remained the same as before, since the bounding surfaces and the normal velocities remain the same. Therefore, the coordinates x^*, y^* that are depicted in Fig. 2 can now be found using these integral equations since the right hand sides of these equations can be calculated. The procedure we follow to determine the coordinates x^*, y^* is described next. In the general case, as shown in Fig. 3, we introduce four unknowns, $x_1^*, x_2^*, y_1^*, y_2^*$ and the iteration starts assuming initial val-

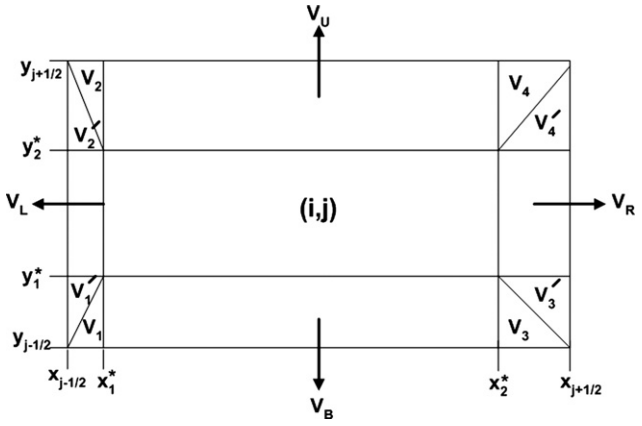

Fig. 3. Schematic showing the more accurate partitioning of the volumes fluxed across the four sides of a computational cell with VT.

ues for all these unknowns. Using Gaussian quadrature the volumes enclosed by the triangles near the corners, i.e. V_1, V_2, V_3 ,V_4, shown in the same figure, are calculated and the variables x_1^* and x_2^* are updated according to the following equations:

$$x_1^* = \left[x_{i-1/2}^2 + 2(V_L + V_1 + V_2)/\Gamma\right]^{1/2} \tag{34}$$

$$x_2^* = \left[x_{i+1/2}^2 - 2(V_R + V_3 + V_4)/\Gamma\right]^{1/2} \tag{35}$$

$$\Gamma = D^2(y_{j+1/2}^3 - y_{j-1/2}^3)/3 + DE(y_{j+1/2}^2 - y_{j-1/2}^2) + E^2(y_{j+1/2} - y_{j-1/2}) \tag{36}$$

where $V_L(V_R)$ stand for the volume as calculated by Eq. (32). Once the variables x_1^* and x_2^* are updated, the volumes enclosed by the triangles, i.e. V_1', V_2', V_3', V_4', are calculated and the variables y_1^* and y_2^* are updated according to the following equations:

$\underline{D \neq 0}$

$$y_1^* = [(E^3 + D\Delta)^{1/3} - E]/D \tag{37}$$

$$y_2^* = [(E^3 + DZ)^{1/3} - E]/D \tag{38}$$

$$\Delta = D^2 y_{j-1/2}^3 + 3DEy_{j-1/2}^2 + 3E^2 y_{j-1/2} + 6(V_B + V_1' + V_3')/(x_{i+1/2}^2 - x_{i-1/2}^2) \tag{39}$$

$$Z = D^2 y_{j+1/2}^3 + 3DEy_{j+1/2}^2 + 3E^2 y_{j+1/2} - 6(V_U + V_2' + V_4')/(x_{i+1/2}^2 - x_{i-1/2}^2) \tag{40}$$

$\underline{D = 0}$

$$y_1^* = y_{j-1/2} + 2(V_B + V_1' + V_3')/(x_{i+1/2}^2 - x_{i-1/2}^2) \tag{41}$$

$$y_2^* = y_{j+1/2} - 2(V_U + V_2' + V_4')/(x_{i+1/2}^2 - x_{i-1/2}^2) \tag{42}$$

where $V_U(V_B)$ stand for the volume as calculated by Eq. (33). This iteration stops when the maximum change of all unknowns between consecutive iterations is less than or equal to 0.5×10^{-12}. Since we are again interested in computing the volume of the core fluid and not the total volume that is fluxed outside of the computational cell, we compute the volumes of the triangle HMGH and of the trapezoid ABKJA using Gaussian quadrature.

4.3. Computation of the surface tension force

The reconstructed interface does not necessarily coincide with the mesh coordinates and as a result the effect of surface tension cannot be enforced as a boundary condition to the momentum equations, but as a continuous body force (CSF). The numerical scheme that we followed in order to introduce the effect of surface tension into the momentum equations is similar to the ALE scheme described by Brackbill et al. [24]. We

approximate the Dirac function appearing in Eqs. (9) and (10) using the gradient of the volume function. This function is indeed an approximation to the Dirac function, since it is zero everywhere in the computational domain except for thin transition regions of distance comparable to the mesh spacing $O(\Delta x, \Delta y)$ where the interface exists. In order to evaluate the curvature of the interface, κ, we need to define a unit vector normal to the interface, $\tilde{\underline{n}}$, using the gradient of the color function:

$$\tilde{\underline{n}} \equiv \frac{\underline{n}}{|\underline{n}|} = \frac{\nabla F}{|\nabla F|} \tag{43}$$

$$n_r = \underline{n} \cdot \underline{e}_r = \partial_r(F) = \frac{\partial_x(F)}{R_2}, \quad n_z = \underline{n} \cdot \underline{e}_z = \partial_z(F) = \frac{\partial_y(R_2 F) - R_2' \partial_x(xF)}{R_2}, \quad |\underline{n}| = \sqrt{n_r^2 + n_z^2} \tag{44}$$

The gradient of the volume function, $\underline{n}$, and thus, both components of the vector normal to the interface, n_r, n_z, as well as its modulus, $|\underline{n}|$, are computed at the mesh coordinate nodes. Upon application of the coordinate transformation introduced in Eq. (6), the curvature term, which is computed at the center of each computational cell where both the pressure and the volume function are evaluated, is given by

$$\kappa = \frac{1}{R_2 |\underline{n}|} \left(\frac{n_r \partial_x(|\underline{n}|)}{|\underline{n}|} + \frac{n_z \partial_y(R_2 |\underline{n}|)}{|\underline{n}|} - \frac{R_2' n_z \partial_x(x|\underline{n}|)}{|\underline{n}|} - \frac{\partial_x(x n_r)}{x} - \partial_y(R_2 n_z) + R_2' \partial_x(x n_z) \right) \tag{45}$$

These differentiations introduce the more serious errors in our computations. The application of a filter produces a smoother variation of the curvature, but introduces artificial diffusion and for this reason we did not use it. More recently, significant improvement in the accuracy of determining the surface tension term and of advecting the interface has been achieved using a parabolic reconstruction of the interface coupled with an optimization algorithm, which, however, could be quite time-consuming [23]. It is worth noting that whenever derivatives of the volume function, Eqs. (19), (20), (43)–(45), or its values for points not located at the center of the cell, Eq. (16), are needed, these are approximated by interpolation of the same accuracy. This way of approximation is preferred because it generates no oscillations around the front either in the velocity or in the pressure field.

5. The Front Tracking method (FT)

In FT, in addition to the Eulerian mesh, an unstructured grid of lower dimensionality is introduced on the moving interface, which is composed of Lagrangian connected particles, to explicitly follow the motion of the interface. Generally, this increases both the accuracy and the complexity of the calculations. The former increases because the interface is continuous by construction and derivatives along and normal to it can be readily and accurately calculated. The complexity increases because Lagrangian particles and Eulerian mesh points do not necessarily coincide. This necessitates transferring information from the nearby Eulerian grid points to the interface via 'interpolation'; and from the interface to nearby grid points via 'spreading', following Peskin's [34] immersed boundary method. The main idea is the introduction of a new finite-difference operator, which is constructed so that the interpolated velocity field on the interface is more nearly divergence-free. This is accomplished by allowing the interface to have a finite but small thickness through a smooth, approximating function, which is nonzero only on a prespecified small number of grid points around the interface [28–30]. We will describe next our adaptation of these ideas to our problem.

Initially at $\tau = 0$, the interface is assumed to be static and continuous and in the transformed coordinate system (x, y) is expressed as a single-valued function of y, $X = X(y)$. We discretize this interface in the interval (0,L) by introducing the points $Y_i = (i-1)h$, $i = 1, \ldots, N$, where $h = \min(\Delta x, \Delta y)$, and $N - 1 = \frac{L}{h}$. Since, Δx and Δy are the lengths of the sides of the computational cells in the x and y directions, respectively and $\Delta x < \Delta y$, this operation may introduce more than one points in each cell. This is something that occasionally introduced wiggles on the interface and we discuss it further in Section 5.3. Using the equation $X = X(y)$, we generate an ordered list of marker particles on the interface with coordinates (X_i, Y_i), $i = 1, \ldots, N$.

5.1. Advection of particles

Having constructed this list of marker particles, next we advance them in time. To this end, first we need to compute the instantaneous velocity of each particle, because the particles do not necessarily coincide with the grid points of the computational mesh where velocities are calculated by solving the NS equation. The velocity, $\underline{U}_f(\underline{X}_i) = (U_f, W_f)$, of a particle with coordinates $\underline{X}_i$ is interpolated using the bulk velocities of nearby grid points, $\underline{U}(\underline{x})$, as follows:

$$(U_f(X_i, Y_i), W_f(X_i, Y_i)) = \sum_{(x,y)} (U(x,y), W(x,y))\delta_{h_1}(xR_2(y) - X_i R_2(Y_i))\delta_{h_2}(y - Y_i), \quad i = 1, ..N \tag{46}$$

where the delta function is split into the product of one-dimensional functions, δ_h, which, for convenience, are assumed to be of identical form when discretized and must be non-singular and approach the delta function, δ, as the mesh size approaches zero and $h_1 = \Delta x R_2(y)$, $h_2 = \Delta y$. Moreover, it must be such that the interpolated velocity field of the interface is nearly divergence-free. The details of constructing this approximation are given in [29,34], where it is also shown that its rather complicated algebraic form is very well approximated by a much simpler trigonometric function:

$$\delta_h(q) = \begin{cases} \frac{1}{4}(1 + \cos\left(\frac{\pi q}{2h}\right)), & |q| \leqslant 2h \\ 0, & |q| > 2h \end{cases} \tag{47}$$

According to Eqs. (46) and (47) only the bulk velocities, $\underline{U}(\underline{x})$, at the nodes $\underline{x}$, which are away from the considered surface point i by less or equal to two times of the mesh size, $2h$, contribute to the interpolated velocity $\underline{U}_f(\underline{X}_i)$. Another possibility, which involves even less grid points closer to the front, has been proposed in [28,30].

$$\delta_h(q) = \begin{cases} (h - |q|)/h, & |q| < h \\ 0, & |q| \geqslant h \end{cases} \tag{48}$$

We used and compared these two representations of the delta function in our problem. The distances in the delta functions are initially in physical space and, in Eqs. (46)–(48), have been transformed to the computational one. Then, the velocity components of a surface point, i, in the transformed domain are equated to the time-derivatives of its coordinates (X_i, Y_i). The resulting equations are solved with the same accuracy as the NS equations, using an explicit Euler scheme to yield its new coordinates:

$$X_i^{n+1} = X_i^n + \Delta\tau(U_f^n - X_i^n R_2'(Y_i^n) W_f^n)/R_2(Y_i^n) \tag{49}$$

$$Y_i^{n+1} = Y_i^n + \Delta\tau W_f^n \tag{50}$$

where the interpolated velocity $\underline{U}_f^n$ is computed at the point (X_i^n, Y_i^n) at time τ, $\Delta\tau$ is the time step and the superscripts n and $n+1$ correspond to times τ and $\tau + \Delta\tau$ respectively.

After an advection step the interface may have come to a stage that breaking up is imminent, for example by coming very close to the axis of symmetry in which case a large drop of the core fluid is created in the center of the tube and is surrounded by the annular fluid. In order to proceed with this breaking of the interface, after the advection of the particles we examine every particle whether $X_i \leqslant \Delta x$. If this holds for the marker-particle k, the interface will break at that point. Then we delete this particle and, simultaneously, set the radial coordinates of its neighbors to zero $X_{k-1} = X_{k+1} = 0$. The new list of particles is $(X_{k+1}, Y_{k+1}), \ldots, (X_N, Y_N)$, $(X_2, Y_2), \ldots, (X_{k-1}, Y_{k-1})$, in which even the point (X_1, Y_1) has been deleted because it coincides with the point (X_N, Y_N) due to the periodicity between the two ends of the tube.

5.2. Parametric representation of the interface

When the points are advanced in time, the length of the interface may change and its shape may stop being a single valued function of y, so we need to introduce a parametric representation of the interface that is independent of such changes. This is easily accomplished by introducing the arc length of each marker particle, S_i, starting with the particle in the entrance of the tube, for which $S_1 = 0$. In order to calculate derivatives of the

interface, e.g. for obtaining its curvature, the new parametric representation of the interface, $(X(S), Y(S))$, is interpolated by cubic splines, which by construction ensure continuity of the function and its second derivative. Continuity of their first derivatives, $X'(S)$, $Y'(S)$, on the internal nodes and periodicity of the interface and its first two derivatives between the two ends of the tube are also imposed. A complication arises in taking derivatives to enforce such continuity on Y between the two end-points of the interface located at the entrance and the exit of the tube. To overcome it, we take advantage of the periodicity of the interface and transform Y_i, $i = 1, \ldots, N$ to TY_i, $i = 1, \ldots, N$ via:

$$TY(S) = Y(S) - L\frac{S - S_1}{S_N - S_1} \tag{51}$$

Upon simplification, two linear equation sets result of size $(N-1, N-1)$, which are tridiagonal, except for their first and last equation, which have off-diagonal terms because of the periodicity conditions. We solve them with $O(N-1)$ operations using the Sherman, Morrison and Woodbury formulas [42].

Even if before their advection the marker particles were such that the first and the last one resided on the left and right edges of the tube, i.e. $Y_1 = 0$ and $Y_N = L$, advection may move them according to one of the following two possibilities:

$$\text{(a) } Y_1^{\text{new}} > 0, \quad Y_N^{\text{new}} > L, \quad \text{(b) } Y_1^{\text{new}} < 0, \quad Y_N^{\text{new}} < L \tag{52}$$

In both these cases we must find the intersection points of the new interface, after advection, with the edges of the tube and use these intersection points instead of those resulting from simple advection in the ordered list of marker particles, $(X_i^{\text{new}}, Y_i^{\text{new}}), i = 1, 2, \ldots, N_{\text{new}}$. To this end we find the arclength of the intersection point between the interface and the exit or entrance plane of the tube using Brent's method. This step can be performed only after the cubic splines for it have been constructed and before proceeding with the redistribution of nodes which requires that the first and the last marker particle coincide with the left and right edge of the tube respectively, see Section 5.3.

If a drop has been formed at the center of the tube because of a breakup of the interface, its two ends are always on the axis of symmetry. Then the boundary conditions that we use at the edges of the drop interface in order to compute $X(S)$ and $Y(S)$ with cubic splines are:

$$X'(0) = 1, \quad X'(S_N) = -1 \quad \text{and} \quad Y'(0) = 0, Y'(S_N) = 0 \tag{53}$$

where S_N is the arc length of the drop surface. When the drop does not intersect the edges of the tube, we do not make use of the transformation (51) because we do not need a periodicity condition for the interface. Moreover, for the computation of $X(S)$ and $Y(S)$ via cubic splines we solve two (N, N) tridiagonal systems using the original set of particles. However, when the drop intersects the edges of the tube at the particle k, then we increase the axial coordinate of all subsequent particles by L, the length of the tube, to extend the drop surface and make it continuous.

5.3. Redistribution of particles on the front

As the interface evolves it deforms and its length increases or decreases or some parts of it may be depleted of or get crowed with Lagrangian particles, so we need to add or remove such particles accordingly. We have observed that decreasing the distance between consecutive marker particles does not improve the accuracy and, in some cases, leads to non physical oscillations. In order to ensure a nearly uniform distribution of marker particles along the interface we choose them so that the distance between two successive particles is of the order of the mesh size. We achieve this by choosing the new ordered list of particles, $(X_i^{\text{new}}, Y_i^{\text{new}})$, $i = 1, \ldots, N$ so that it consists of the two interface points that coincide with the edges of the tube (whenever such intersections exist) and the middle points of the segment between intersection points of the interface with the sides of each computational cell. To find the latter we use the ordered list (X_i, Y_i), $i = 1, \ldots, N$, the corresponding arc lengths S_i, $i = 1, \ldots, N$ and the parameterization $(X(S), Y(S))$ we generated previously. For every marker particle, i, after finding the computational cell to which it belongs, we determine which sides of the cell are intersected by the corresponding segment of interface, (S_{i-1}, S_i). For example, (S_{i-1}, S_i) intersects the vertical side $y = y_0$ of a cell, if $(Y(S_{i-1}) - Y_0)(Y(S_i) - Y_0) < 0$. Then we use Brent's method to find the arc length

$S \in (S_{i-1}, S_i)$ of the intersection point. At the end of this procedure we set the arc lengths of all the intersection points into an increasing order, and select as new marker particles the points at half the distance between two consecutive intersections. In this way we ensure that there is one and only one marker particle in each cell crossed by the interface.

5.4. Computation of surface tension force

The curvature and the normal to the interface, κ and $\underline{\tilde{n}}$ are needed to determine the capillary force in Eq. (3). They are computed first in physical space according to their definitions in cylindrical coordinates:

$$\underline{\tilde{n}} = \frac{Z_S \underline{e}_r - R_S \underline{e}_z}{\sqrt{R_S^2 + Z_S^2}} \tag{54}$$

and

$$\kappa = \frac{Z_S(R_S^2 + Z_S^2) + R(R_S Z_{SS} - Z_S R_{SS})}{R(R_S^2 + Z_S^2)^{3/2}} \tag{55}$$

and then they are transformed according to Eq. (6) in computational space:

$$\begin{aligned} &R(S) = X(S)R_2(Y(S)), \quad Z(S) = Y(S) \\ &R_S = X_S R_2(Y) + XY_S R_2'(Y), \quad Z_S = Y_S \\ &R_{SS} = X_{SS} R_2(Y) + 2X_S Y_S R_2' + X(Y_{SS} R_2' + Y_S^2 R_2''), \quad Z_{SS} = Y_{SS} \end{aligned} \tag{56}$$

In the above notation, a variable in subscript indicates partial differentiation with respect to this variable. In Eqs. (54) and (55), the factor $\kappa\underline{\tilde{n}}$ can be computed only at surface points, but it must be transferred (spread) to the Eulerian grid in order to use it in the Navier–Stokes equations. Therefore, the delta function representing the front must be approximated and smoothed out over the Eulerian grid. This smoothing can be done in several different ways [30,34], but it is always necessary to ensure that the quantity transferred is conserved. This is the opposite operation to the interpolation we employed in Section 5.1. To this end, if Ω is the control volume for one of the components of the momentum and AB, the interface segment inside this control volume, we must require that:

$$\int_\Omega \kappa \underline{\tilde{n}} \delta(\underline{r} - \underline{R}_f)\, \mathrm{d}\Omega = \int_A^B \kappa \underline{\tilde{n}} r\, \mathrm{d}S \tag{57}$$

Given the parametric representation of the interface, $(X(S), Y(S))$, we find its intersection points with the sides of the x and y control volumes (centered at the U and W velocity nodes) of the computational domain, respectively, constructing two ordered lists of those intersection points. Then we calculate the term $\kappa\ \underline{\tilde{n}}$ in the center of every two consecutive intersection points. In order to compute $\kappa\underline{\tilde{n}}\ \delta(\underline{r} - \underline{R}_f)$ at the grid point (i,j) and simultaneously satisfy Eq. (57) we write:

$$[\kappa \underline{\tilde{n}} \delta(\underline{r} - \underline{R}_f)]|_{ij} = \sum_f [\kappa \underline{\tilde{n}}]_f \omega_{ij}^f \frac{\Delta S_f}{V_\Omega} \tag{58}$$

where ΔS_f is the interface segment between two consecutive intersection points of the interface with the sides of a control volume and ω_{ij}^f is the weight of grid point (i,j) in this interface segment. Also $[\kappa\underline{\tilde{n}}]_f$ is estimated at the center of ΔS_f and V_Ω is the volume of the control volume Ω. The weights must satisfy the following condition:

$$\sum_{ij} \omega_{ij}^f = 1 \tag{59}$$

The weight for the grid point (i,j) for smoothing from the interface point (R_f, Z_f) is given by

$$\omega_{ij} = \delta_{\Delta r}(R_f - r_i)\delta_{\Delta z}(Z_f - z_j) \tag{60}$$

where $R_f = X_f R_2(Y_f)$, $Z_f = Y_f$, $r_i = x_i R_2(y_j)$, $z_j = y_j$ according to the transformation, Eq. (6). As discrete form of the delta function we use either Eq. (47) or Eq. (48) with arguments in the r-direction, $\delta_{\Delta r}(R_f - r_i)$, $q = R_f - r_i$ and $\Delta r = \Delta x\ R_2(z_j)$, while in the z-direction, $\delta_{\Delta z}(Z_f - z_j)$, $q = Z_f - z_j$ and $\Delta z = \Delta y$.

5.5. Computation of the physical properties

The physical properties of the two fluids are computed according to the same equation used for our VT algorithm, namely Eq. (17). However, in FT the function F, often here called indicator function, is not calculated along with the other unknowns in the bulk, but via the equation:

$$\nabla^2 F = \nabla \cdot \underline{G} \tag{61}$$

The vector $\underline{G}$ is an approximation of ∇F which is computed according to:

$$\underline{G}_{ij} = \sum_f \tilde{\underline{n}}_f \omega_{ij}^f \frac{\Delta S_f}{V_\Omega} \tag{62}$$

where $\underline{G}_{ij}$ is the value of $\underline{G}$ at grid point (i,j) and $\tilde{\underline{n}}_f$ is calculated at the center of ΔS_f. In order to compute the function F in the center of each computational cell, first we estimate the divergence of $\underline{G}$ and then we solve Eq. (61) with the appropriate boundary conditions for F. The Laplacian is approximated with centered finite differences of second order. Occasionally, over- or undershoots in fluid properties arise near the front, which are filtered out.

6. Solution of the pressure equation

The simultaneous solution of the momentum and mass balances in the bulk of both fluids is very costly. In order to reduce it, we decouple the momentum balance which we solve explicitly in time and then we impose the mass balance by the projection method proposed by Chorin [44]. In particular, in order to advance the flow field (U, W, p) from the time level (n) to $(n+1)$ we compute all the terms that appear in the right hand side of Eqs. (9) and (10) using the flow field obtained at the previous time level (n) and thus, using the left hand side of the same equations a first approximation of the radial and axial velocities (U^*, W^*) is obtained. However, these velocities do not satisfy the continuity equation and must be corrected accordingly. To this end, the flow field that corresponds to the time level $(n+1)$ is split into two parts as follows.

$$p^{n+1} = p^n + \frac{\Delta p}{\Delta \tau} \tag{63}$$

$$U^{n+1} = U^* - \frac{\partial_x(\Delta p)}{\rho R_2}, \quad W^{n+1} = W^* + \frac{R_2' \partial_x(x\Delta p)}{\rho R_2} - \frac{\partial_y(R_2 \Delta p)}{\rho R_2} \tag{64}$$

Eq. (64) expresses the sensitivity of the flow field to the changes of the pressure correction, Δp, and by forcing the velocity field at the time level $(n+1)$ to be divergence-free we obtain the following equation:

$$\begin{aligned} &R_2 \partial_x \left(\frac{x}{\rho R_2} (\partial_x(\Delta p) - x R_2' (\partial_y(R_2 \Delta p) - R_2' \partial_x(x \Delta p))) \right) + x \partial_y \left(\frac{R_2}{\rho} (\partial_y(R_2 \Delta p) - R_2' \partial_x(x \Delta p)) \right) \\ &\quad = R_2 \partial_x (x(U^* - x R_2' W^*)) + x \partial_y (R_2^2 W^*) \end{aligned} \tag{65}$$

It is worth noting that if one works in the original cylindrical coordinate system the resulting equation for Δp, is the Poisson equation, but this is not the case here because the coordinate transformation generates a more complicated differential equation. Clearly, in addition to the derivatives that arise in the Poisson equation expressed in the cylindrical coordinate system, cross derivatives of the pressure correction arise here as well. These additional terms are all multiplied with the first derivative of the solid surface with respect to the axial distance and, when discretized, they give rise to a non-symmetric Jacobian matrix for which library routines and conventional iterative methods often fail. This second order linear partial differential equation with non-constant coefficients is solved subject to the following boundary conditions:

$$\partial_x(\Delta p)|_{x=0} = \partial_x(\Delta p)|_{x=1} = 0 \tag{66}$$

$$\Delta p|_{y=0} - \Delta p|_{y=\Lambda} = \partial_y(\Delta p)|_{y=0} - \partial_y(\Delta p)|_{y=\Lambda} = 0 \tag{67}$$

Eq. (66) results from the fact that at these boundaries the radial velocity equals zero, while Eq. (67) as boundary conditions in the axial direction arise from the fact that the resulting pressure at the time level $(n+1)$ is periodic. The discretization of Eq. (65) is made using centered finite differences in space and the associated boundary conditions are enforced using fictitious boundary nodes as for the other variables. The pressure correction variable Δp is computed at the same grid points as the pressure, p, i.e. staggered with respect to the points used for the computation of the radial and axial velocities. The boundary conditions given in Eqs. (66),(67) are not sufficient for the solution of Eq. (65), since we must specify the pressure datum. For this reason, we drop the continuity equation at one grid point and we impose $\Delta p = 0$ instead and thus, the resulting problem is well defined.

6.1. Conventional solvers

The solution of Eq. (65) is the most time-consuming part of the code and for that reason we attempted to solve it in three different ways. First, we implemented the S.O.R method but we noticed that when topological changes occur, which are always associated with large temporal variations of the flow field, the number of required iterations to decrease the maximum error in imposing the divergence-free flow below 0.5×10^{-7} increases even above 10^3. Thus, we are obliged to decrease the time step, but the sequence of decreasing the time step in order to keep the required number of iterations to a reasonable limit, $O(10^2)$, leads to such a small time step, $O(10^{-7})$, that the computations become prohibitively expensive and hence this method of solution becomes inefficient. Then, we tried to solve Eq. (65) using the bi-conjugate gradient method along with the Cholesky decomposition as a preconditioner [21]. In that case we observed that, when the constriction ratio becomes smaller than 0.75, the iterative method does not converge and since we are interested in examining cases characterized by a wide range of constriction ratios we abandoned this method of solution as well.

Finally, we used a subroutine suitable for banded matrices that employs Gauss elimination. The benefit of this choice is that since Eq. (65) is linear, only one matrix inversion is needed to attain a solution and as a result no decrease of the time step is required irrespective of the complexity of the flow field. Moreover, the divergence of the resulting velocity field is zero within machine accuracy and not approximately in accordance with the requirements of our algorithm for interface advection. We have further accelerated the time integration by using the LU decomposed matrix that corresponds to a previous time step in order to find the solution at the current one. The need to recalculate the Jacobian matrix at regular time intervals stems from the fact that although the governing equation of the pressure correction is linear its coefficients depend on time through the density ratio. Of course, when $\rho = 1$, i.e. the two fluids are of equal densities, the Jacobian matrix that is formed in the first time step does not need updating and it can be used throughout the computations. In this case, having the LU decomposed matrix, only the back-substitution step is needed and thus, the solution at every time step can be calculated in only $O(m \cdot n)$ operations, where m is the bandwidth of the resulting matrix and n is the number of unknowns.

It is worth noting that the imposition of periodic boundary conditions in the axial direction introduces an additional problem to the solution of the pressure equation. This is due to the fact that in order to achieve the minimum storage requirements and the least computational time for solving Eq. (65), the resulting algebraic set of equations are numbered in the radial direction first. In this way the bandwidth of the resulting matrix is smaller, since more computational cells are used for resolving the axial direction than the radial one; the axial length of the domain is about 10 times the maximum radius of the tube. In order to make clear the additional complexity in imposing the periodicity in the axial direction to the numerical solution of the problem, assume that the computational domain is divided into L and M computational cells in the radial and axial directions, respectively. Including the fictitious boundary cells that are attached along the computational domain the total unknowns are $(L+2)(M+2)$ and the bandwidth of the resulting matrix, when the equations are numbered in the radial direction, equals $(2L+7)$. But the periodicity imposed between the two ends of the tube relates unknowns that are outside the bandwidth; the number of these equations is $(2L+4)$. After careful manipulations the resulting matrix resembles that shown in Fig. 4 in which the algebraic equations that impose the

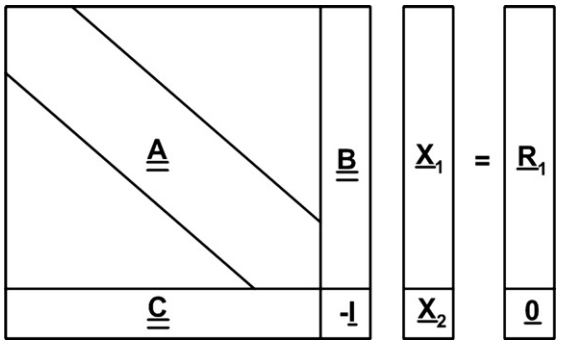

Fig. 4. Schematic representation of the type of matrix problem that arises after rearranging the algebraic set of equations that results by discretizing with finite differences a Poisson-like equation with periodic and Neumann boundary conditions in the axial and radial direction, respectively.

periodicity form the lower part of the matrix. The matrix $\underline{\underline{A}}$ has dimension $(L+2)M\times(2L+7)$, the matrix $\underline{\underline{B}}$ $(L+2)M\times(2L+4)$, the matrix $\underline{\underline{C}}$ $(2L+4)\times(L+2)M$ and $\underline{\underline{I}}$ which is the identity matrix is of dimensions $(2L+4)\times(2L+4)$. By $\underline{X}_1$,$\underline{X}_2$ and $\underline{R}_1$ we denote the associated solution and residual vectors respectively. The solution to this problem can be simplified as follows:

$$\{\underline{\underline{A}}\underline{X}_1+\underline{\underline{B}}\underline{X}_2=\underline{R}_1,\underline{\underline{C}}\underline{X}_1-\underline{\underline{I}}\underline{X}_2=\underline{0}\}\Longleftrightarrow\{(\underline{\underline{C}}\underline{\underline{A}}^{-1}\underline{\underline{B}}+\underline{\underline{I}})\underline{X}_2=\underline{\underline{C}}\underline{\underline{A}}^{-1}\underline{R}_1,\underline{X}_1=\underline{\underline{A}}^{-1}\underline{R}_1-\underline{\underline{A}}^{-1}\underline{\underline{B}}\underline{X}_2\} \tag{68}$$

The benefit of using the second form of Eq. (68) is that now the solution for the vector $\underline{X}_2$ is decoupled from the solution for the vector $\underline{X}_1$. Once we compute the LU decomposition of matrix $\underline{\underline{A}}$ what we need in order to compute the vector $\underline{X}_2$ is the products $\underline{\underline{A}}^{-1}\underline{\underline{B}}$ and $\underline{\underline{A}}^{-1}\underline{R}_1$, which are evaluated by applying the back-substitution step for multiple right hand sides which are formed by the columns of the matrix $\underline{\underline{B}}$ and the vector $\underline{R}_1$. The matrix-matrix multiplication $\underline{\underline{C}}(\underline{\underline{A}}^{-1}\underline{\underline{B}})$ is trivial and, once the matrix $\underline{\underline{C}}(\underline{\underline{A}}^{-1}\underline{\underline{B}})+\underline{\underline{I}}$ as well as the residual vector $\underline{\underline{C}}\underline{\underline{A}}^{-1}\underline{R}_1$ are formed, the computation of the vector $\underline{X}_2$ is straightforward using a dense matrix solver. Having computed the solution vector $\underline{X}_2$, it is used for computing the solution vector $\underline{X}_1$.

6.2. Multigrid solver

We developed our own multigrid solver following the ideas presented in Ref. [39]. It solves Eq. (65), which is not a simple Poisson equation for the reasons explained earlier, combining the efficiency of multigrid iteration with the certainty of a direct method. The algorithm employs banded Gaussian elimination in place of relaxation, whenever the coarsest subgrid is encountered within multigrid cycling. This provides additional grid size flexibility by eliminating the usual multigrid constraint that the coarsest grid should consist of very few points for effective error reduction within multigrid cycling. The solution region of all equations, i.e. radial and axial momentum as well as the continuity equations is subdivided into elements in every grid level. Since in our two-dimensional grid, the radial and axial velocities and the pressure are computed from their respective equations at different locations, the 1D cell- and vertex-centered restriction and prolongation operators are extended to 2D differently for each variable and in each equation. Special care is needed in the cell-centered prolongation operator when dealing with the boundary conditions.

The developed method is a recursive, linear, *W*-cycle, fixed-schedule, multigrid algorithm that employs 2 iterations in the pre-smoothing and 1 iteration in the post-smoothing phases using a four-color Gauss–Seidel relaxation method. Additionally, the solution vector at the coarsest grid level is calculated using banded Gaussian elimination. At the start of each time step the coarsest grid solution is utilized as an initial approximation of the solution vector and is refined (max absolute rhs error $\leqslant 0.5\times10^{-6}$) by utilizing additional multigrid iterations. When the number of multigrid iterations reaches the preset maximum allowed number of 20 and the solution error exceeds the pre-specified limit, the time step is reduced by 10%. On the contrary, when the error is reduced below the pre-specified threshold in less than 8 iterations the time step is increased by 5%, but at the same time is kept small enough ($\sim0.2\times10^{-4}$), so that the integration scheme does not turn unstable.

7. Results and discussion

In our earlier studies on Core-Annular-Flow (CAF) in an undulating tube [7,8], we had shown that it remains stratified with both phases continuous under specific, quite strict conditions, e.g. the outer fluid must be less viscous than the core fluid, its thickness below a certain limit, the value of which depends on the Reynolds and Weber numbers, etc. Otherwise the interface would become unstable and possibly break leading to a different arrangement of the two fluids. However, this evolution could not be studied with a Finite element or a Spectral method we employed before, but it can be studied with either the VT or FT algorithms we are presenting herein.

7.1. VT versus FT when the interface remains continuous and Re is low

We start the evaluation of our algorithms by presenting a case where linear theory [7] predicts that the steady Core-Annular-Flow is unstable because the two fluids are of equal viscosity. Nevertheless, we keep the Reynolds and Weber numbers at moderate values. In particular, the parameter values are $(\alpha, \Lambda, \mu^o, \rho^o, Re, We^{-1}, V) = (0.5, 10, 1, 1, 50, 1, 0.49)$. We have carried out computations using either the VT or the FT algorithms using various meshes, while keeping the time step at $\Delta\tau = 5 \times 10^{-5}$. Fig. 5 shows one period of the tube wall and the fluid/fluid interface at time $\tau = 10$ predicted with a mesh of $(NX, NY) = (125, 625)$ computational cells. Although nonlinear waves of larger amplitude and wavelength before the constriction than after it have developed on the interface, one can barely see a difference between the predictions obtained with the two methods. Next we examine the velocity field (streamlines and velocity vectors) under the same conditions. Again, we observe in Fig. 6 a most satisfying quantitative agreement, as the minimum, maximum and all intermediate values of the streamfunction coincide, see figure caption for specific values. As expected, the velocity is maximized near the centerline and around the tube constriction and minimized near the tube wall when the wall assumes its maximum radius.

The fact that the two methods converge to the same result can be demonstrated by computing the Euclidean norm of the difference between the velocity components or the pressure obtained either by VT or FT everywhere in the domain. To this end, we chose as coarsest mesh one with $(NX, NY) = (25, 125)$ computational cells within the tube (or 3125 pressure unknowns, not counting the unknowns in the fictitious cells outside the tube) and multiplied by the same factor of 2, 3, etc. the number of radial and axial cells until we arrived at the finest mesh which has $(NX, NY) = (125, 625)$ cells (or 78125 pressure unknowns). The CPU times required to compute these results up to $\tau = 10$ using our VT algorithm and

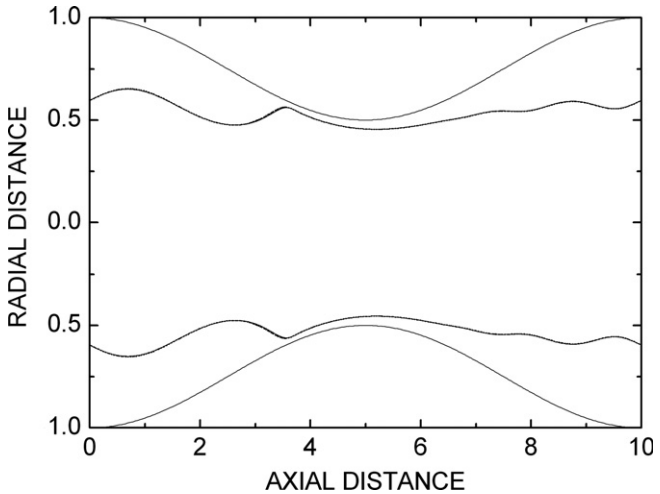

Fig. 5. A snapshot from the evolution of the interface in a period of the tube geometry for the following flow and geometric parameters: $(\alpha, \Lambda, \mu^o, \rho^o, Re, We^{-1}, V) = (0.5, 10, 1, 1, 50, 1, 0.49)$, whereas $(NX, NY, \Delta\tau, \tau) = (125, 625, 5 \times 10^{-5}, 10)$. The continuous line corresponds to the FT method and the dashed line to the VT one, but the two lines nearly coincide.

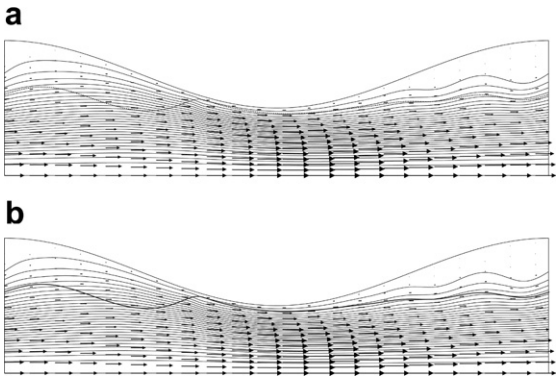

Fig. 6. The velocity vectors and streamlines for flow conditions identical with those in Fig. 5, computed using (a) the FT algorithm (streamfunction values between 0 and 0.4233 with a step of 0.01411) and (b) the VT algorithm (streamline values between 0 and 0.4236 with a step of 0.01412). The fluid/fluid interface is also shown.

the LU solver range from 20 min for the 25×125 mesh to 22 h for the 125×625 mesh, on a dual core XEON at 2.8 GHz. Increasing the number of cells in this way and keeping the time step constant as given in the figure caption may not be optimal, but will allow us to isolate the effect of mesh refinement from other factors. Fig. 7 shows that the differences of both velocity components and pressure converge. More specifically, the velocity norms are smaller by an order of magnitude than the corresponding pressure norm, but the latter converges faster with mesh refinement. The rate of convergence of each method can be assessed by computing the Euclidean norm of the differences of the values obtained with any mesh to those obtained by the finest mesh for the same variables. For the computation of the norms for this relative error, the values of all variables at meshes other than the coarsest one were interpolated at the locations corresponding to the coarsest mesh. The result is shown in Fig. 8 for either the axial velocity or the pressure variables. For the first three meshes the relative error with either method decreases almost quadratically with mesh refinement. Moreover, the relative error with VT in the axial velocity is slightly smaller than the relative error for FT, but at the finest mesh it decreases faster for FT, whereas the error in the axial velocity calculated with the VT method does not decrease further. When we decreased the time

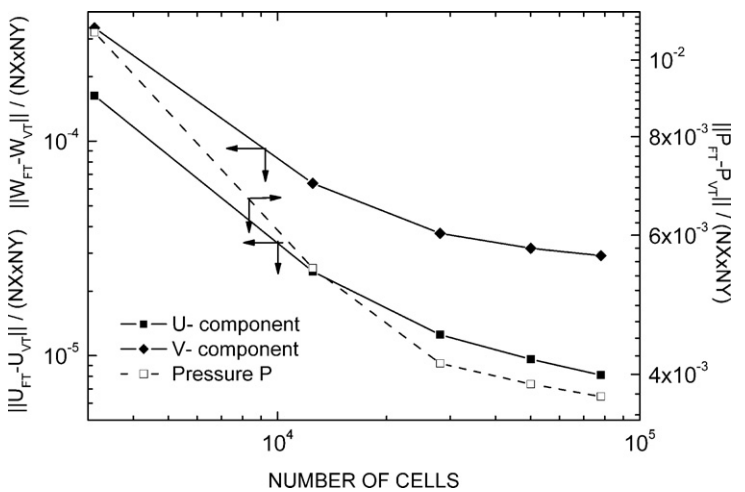

Fig. 7. The Euclidian norm of the difference between the solution vectors (velocity components with filled symbols or pressure with open symbols) corresponding to the FT and VT methods divided with the number of total cells in the coarsest grid. The computational cells used are: 25×125, 50×250, 75×375, 100×500 all with $\Delta\tau = 10^{-4}$ and 125×625 with $\Delta\tau = 5 \times 10^{-5}$. The flow conditions are identical with those in Fig. 5.

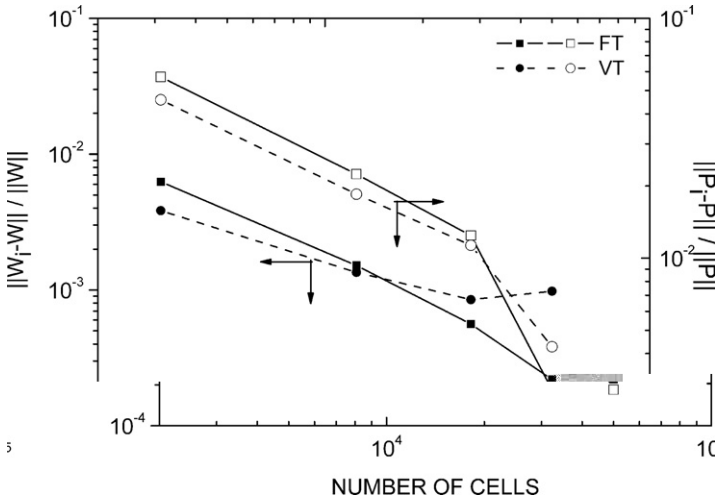

Fig. 8. The relative error of the axial velocity (filled symbols, left vertical axis) and the pressure (open symbols right vertical axis), for both FT (squares) and VT (circles) methods. As a reference value, W and P, the solution for a discretization with 125×625 computational cells and $\Delta\tau = 5 \times 10^{-5}$ is used, while the vectors W_i and P_i are calculated using 25×125, 50×250, 75×375 and 100×500 cells all with $\Delta\tau = 10^{-4}$. The flow conditions are identical with those in Fig. 5.

step to $\Delta\tau = 5 \times 10^{-5}$ for the 100×500 mesh or down to $\Delta\tau = 10^{-5}$ for the two finest meshes things did not change, indicating that the size of the time step did not play a role. Hence we attribute this cessation of decrease in error in VT with mesh refinement to the increased number of interface discontinuities between adjacent cells. Again the error in pressure is larger than the error in the velocity. This should have been expected, because not only pressure is always the most sensitive variable but also in its norm the constant pressure gradient (see Eq. (5)) has not been included. The latter would increase only the denominator of the relative error in pressure. Such numerical tests have been performed in other cases, when the Reynolds number increased or the interface broke or capillarity became important. They will be presented occasionally for comparison purposes in the following, otherwise only converged results with relative error in the axial velocity less than 10^{-3} will be presented.

One of the well-documented shortcomings of VT and FT algorithms are the so-called "spurious" or "parasitic" currents. They arise either because finite difference approximations of the delta function and its derivatives in calculating the curvature are not accurate enough (VT) or because the pressure is not calculated accurately enough (FT). They are observed as small velocities near the surface of an otherwise stationary drop and may lead to its non-physical deformation. Clearly, if the flow is strong enough they are not important. We tested the case of a stationary drop with radius $R = 0.25$ located in the middle of a straight tube of radius 1 and length 10, assuming that the fluid inside the drop has the same density and viscosity with the annular fluid. Without imposing an external pressure gradient, $f = 0$, the velocity should be zero everywhere and the drop should have a spherical shape. The absence of the pressure gradient necessitates introducing characteristic scales based on capillarity and results in a momentum balance similar to that given by Eq. (3), except that the inverse Weber number does not arise and instead of Re^{-1}, the Ohnesorge number multiplies the viscous terms, $Oh = \hat{\mu}_1/(\hat{T}\hat{\rho}_1\hat{R})^{1/2}$, where $\hat{R}$ is the radius of the drop. As capillary forces increase in comparison to viscous forces the Ohnesorge number will decrease and spurious currents should become more prominent. We used both our VT and FT codes and we report the results after 2000 time steps and for different mesh discretizations. Table 1 gives the maximum velocity anywhere in the tube. We observe that generally, (i) the spurious currents increase as the Oh decreases, (ii) FT gives more accurate results than VT and (iii) mesh refinement is effectively reducing them only when FT is used. Similar conclusions have been reached in Ref. [23], where the PROST algorithm decreased these currents quite effectively. It is important to determine if this numerical inaccuracy will affect the rest of our computations. Assuming that fluid properties are those of water (with its large surface tension), we find that for a drop of radius 1cm, the maximum velocity generated by the spurious currents is less than 10^{-3} m/s, whereas for 0.1 atm/m driving pressure gradient the maximum axial

Table 1
Variation of the maximum magnitude of the velocity vector with increasing mesh discretization for various values of $Oh = (\hat{\mu}_1^2/\hat{R}\hat{\rho}_1\hat{T})^{1/2}$ computed with both our VT and FT methods and generated around a stationary drop in the middle of a straight tube

Mesh size NX × NY	Oh = 0.1 $\lvert\underline{U}\rvert_{max}$	Oh = 1 $\lvert\underline{U}\rvert_{max}$	Oh = 10 $\lvert\underline{U}\rvert_{max}$
VT method			
5×25	1.4231×10^{-2}	1.1234×10^{-2}	3.0842×10^{-3}
10×50	1.7057×10^{-1}	8.0876×10^{-2}	1.2259×10^{-2}
20×100	3.7529×10^{-1}	8.8001×10^{-2}	1.0228×10^{-2}
40×200	3.2026×10^{-1}	3.5184×10^{-2}	3.8531×10^{-3}
80×400	1.6859×10^{-1}	3.3156×10^{-2}	3.7541×10^{-3}
FT method			
5×25	1.4867×10^{-2}	1.1708×10^{-2}	3.2207×10^{-3}
10×50	1.9160×10^{-2}	1.1507×10^{-2}	2.2134×10^{-3}
20×100	1.2213×10^{-2}	5.6054×10^{-3}	1.0088×10^{-3}
40×200	8.2268×10^{-3}	1.6451×10^{-3}	2.4312×10^{-4}
80×400	6.5711×10^{-3}	7.3645×10^{-4}	8.1651×10^{-5}

The velocity values are taken after 2000 time steps and $\Delta\tau = 5\times10^{-6}$.

velocity we obtain in our simulations is larger than 1 m/s. This means that our results are not affected by the spurious currents.

7.2. Centered differencing versus upwinding at large Re

Next we increase the Reynolds number and the relative viscosity of the outer fluid, moving in this way further away from the curve of marginal stability [7]. In Fig. 9, we show only the upper half of the tube and the time-evolution of the fluid/fluid interface computed using our VT algorithm with two different levels of mesh discretization among those we have tried in this study and for the parameter values $(\alpha, \Lambda, \mu^o, \rho^o, Re,$

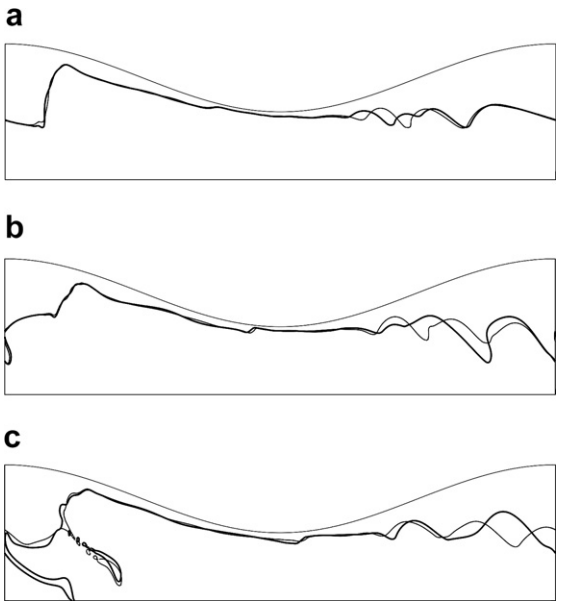

Fig. 9. Evolution of the interface when the convective terms are approximated with central finite differences. The thin line corresponds to the computational mesh of (75×350) cells in the radial and axial directions respectively and the thick line to (100×500) cells. The time step in both discretizations is $\Delta\tau = 5\times10^{-5}$, the parameters are $(\alpha, \Lambda, \mu^o, \rho^o, Re, We^{-1}, V) = (0.5, 10, 5, 1, 500, 0.1, 0.49)$. (a) $\tau = 6.4$, (b) $\tau = 6.8$ and (c) $\tau = 7.2$.

$We^{-1}, V) = (0.5, 10, 5, 1, 500, 0.1, 0.49)$. Here, in spite of the fact that the Reynolds number is fairly large, we have approximated the convective terms in the momentum equations with centered finite differences as in Section 7.1. We observe that very irregular variations arise in the interface with amplitude that increases with time. Even worse than this is the fact that the solution does not converge with mesh refinement, even when we tried finer meshes than the two shown in this figure. To remedy this problem, we introduced the upwind scheme discussed in Section 3 to approximate the convective terms. First, we tested for the upwind parameter β the one proposed by TF [37]. The obtained results are shown in Fig. 10. There is a significant reduction of the non-physical variations of the interface (and the flow field, which is not shown here) and convergence of the interface with mesh refinement. Now we can observe that waves do not appear on the interface for about half the tube length, around the tube constriction. However, looking closer at Fig. 10, we observe some persistent, small and irregular variations in part of the interface. They arise after the tube constriction as the fluids tend to expand. Further downstream the large inertia locally flattens the interface and decreases the deviation between the two meshes. This deviation reappears as the interface undergoes an abrupt change just before the next constriction as the less viscous core fluid tries to pass through it, but inertia delays the contraction in the interface. This, due to periodicity, appears on the left side of the figure. This deviation increases as time proceeds. For this reason, we also tested the second value for β proposed by SL [38]. The results are shown in Fig. 11 for the same two meshes and time instants as in the previous two figures. Clearly, now, there is complete elimination of the variations in the interface and convergence of the results with mesh refinement. This fluid arrangement in a packed-bed, where both phases are continuous is called trickling flow. To quantify the error and its dependence on the mesh and demonstrate that centered differences do not converge, to the extent we tested them, we prepared Fig. 12. It shows the same norm of the pressure, the more sensitive of the field variables, for the same four meshes used in Fig. 8 in comparison to the finest mesh which has $(NX, NY) = (125, 625)$ computational cells. When centered differences are used, the relative error norm oscillates as the mesh becomes finer, always remaining large. On the contrary, when upwinding is introduced, the relative error decreases with mesh refinement and is consistently smaller for all meshes used when the β value proposed by SL is used. The CPU times required to compute these results up to $\tau = 7.2$ using our VT algorithm range from 15 min for the 25×125 mesh to 16 h for the 125×625 mesh.

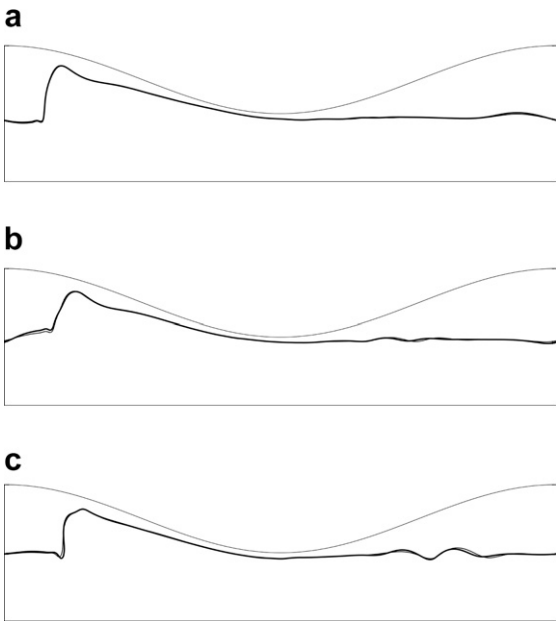

Fig. 10. Evolution of the interface when the convective terms are approximated with upwinding using the parameter β suggested by TB [37]. The thin line corresponds to computational mesh with (75×350) cells in the radial and axial directions, respectively and the thick line to (100×500) cells. The time step in both discretizations is $\Delta\tau = 5 \times 10^{-5}$. Parameters are the same as in Fig. 9 (a) $\tau = 6.4$, (b) $\tau = 6.8$ and (c) $\tau = 7.2$.

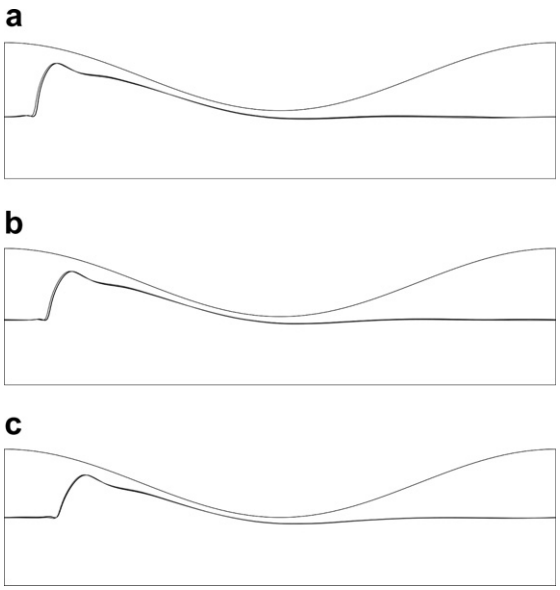

Fig. 11. Evolution of the interface when the convective terms are approximated with upwinding using the parameter β suggested by SL [38]. The thin line corresponds to computational mesh with (75×350) cells in the radial and axial directions respectively and the thick line to (100×500) cells. The time step in both discretizations is $\Delta\tau = 5 \times 10^{-5}$. Parameters are the same as in Fig. 9. (a) $\tau = 6.4$, (b) $\tau = 6.8$ and (c) $\tau = 7.2$.

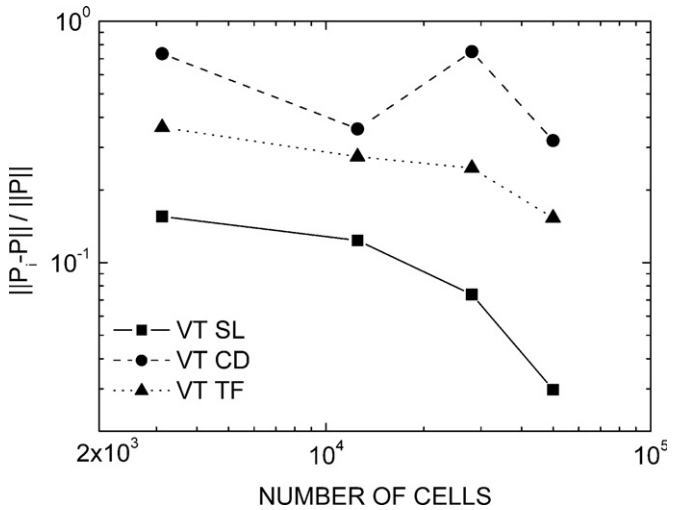

Fig. 12. The relative error of the pressure for the VT method using central differences (circles), and the upwinding approximations of SL (squares) and TF (triangles). The reference vector P is computed using 125×625 computational cells, while the pressure vectors P_i are calculated using 25×125, 50×250, 75×375 and 100×500 cells. The flow parameters parameters are as in Fig. 9 and $\tau = 7.2$.

Next we examine the interface evolution for the same Reynolds number, but the core fluid is more viscous and its volume much smaller than before, $(\mu^o, V) = (0.2, 0.04)$. The change in the viscosity ratio is expected to stabilize the interface, whereas the change in the volume of the core fluid may induce a capillary instability because it increases the importance of capillary forces. The results in Fig. 13 were obtained using VT and centered finite differences for the convective terms. Again results at 3 time instants with two different discretizations are shown. Here we observe that capillary forces break the interface forming a long drop of core fluid along the axis of symmetry of the tube, before $\tau = 4.0$. The large value of the Reynolds number, makes the

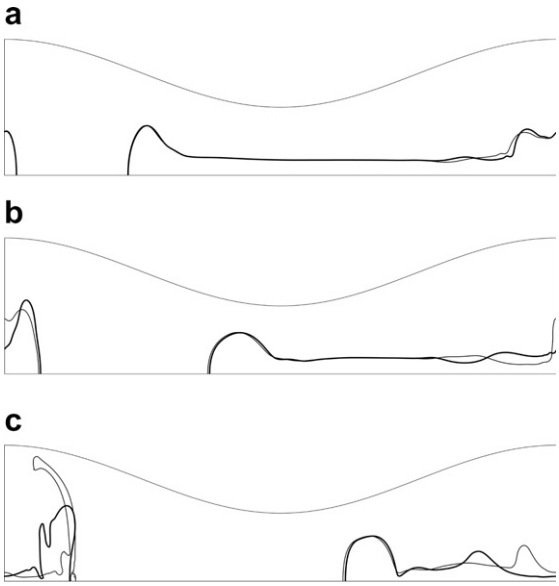

Fig. 13. Evolution of the interface when the convective terms are approximated with central finite differences. The thin line corresponds to the computational mesh of (75×350) cells in the radial and axial directions respectively and the thick line to (100×500). The time step in both discretizations is $\Delta\tau = 5 \times 10^{-5}$. The parameter values are $(\alpha, \Lambda, \mu^{o}, \rho^{o}, Re, We^{-1}, V) = (0.5, 10, 0.2, 1, 500, 0.1, 0.04)$ and the times (a) $\tau = 4.0$, (b) $\tau = 4.4$ and (c) $\tau = 4.8$.

centered finite difference approximation of the convective terms inaccurate, because large variations of the interface are seen which do not converge with mesh refinement, on the contrary, their magnitude increases with time, Fig. 13b and c. By approximating the convective terms in the momentum equations by the upwinding scheme of Ref. [37] and using the parameter β as proposed by TF, these variations are reduced significantly and convergence with mesh refinement is improved, Fig. 14. On the other hand, using the value of β proposed

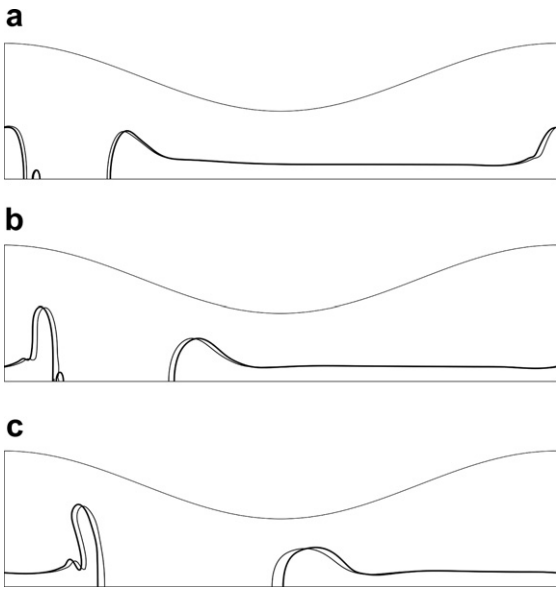

Fig. 14. Evolution of the interface when the convective terms are approximated with upwinding using β proposed by TF. The thin line corresponds to the computational mesh of (75×350) cells in the radial and axial directions respectively and the thick line to (100×500) cells. The time step in both mesh discretizations is $\Delta\tau = 5 \times 10^{-5}$. Parameters are the same as in Fig. 13 and the times (a) $\tau = 4.0$, (b) $\tau = 4.4$ and (c) $\tau = 4.8$.

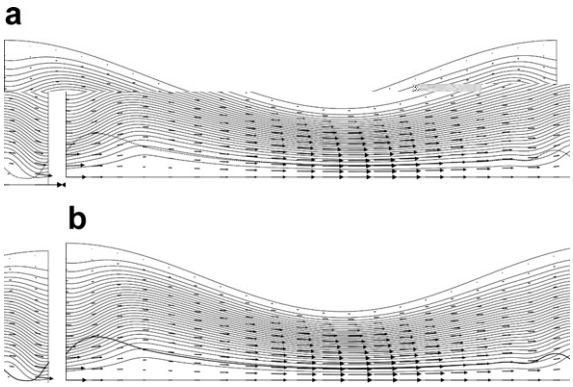

Fig. 15. Velocity field for $\tau = 4.4$ computed with (a) the FT algorithm and (b) the VT algorithm. The streamfunction values are between 0 and 0.11 with a step of 0.00367 in both figures. The parameter values are $(\alpha, \Lambda, \mu^{o}, \rho^{o}, Re, We^{-1}, V) = (0.5, 10, 1, 1, 10, 1, 0.04)$, whereas $(NX, NY, \Delta\tau) = (125, 625, 5 \times 10^{-5})$.

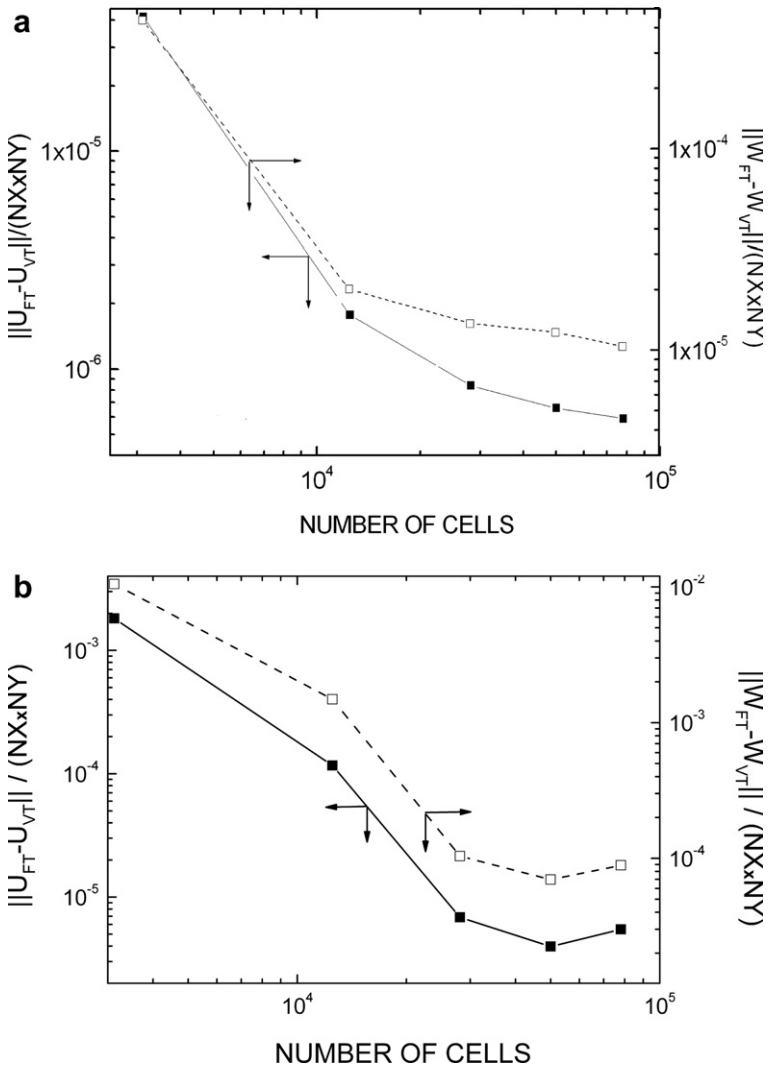

Fig. 16. The Euclidian norm of the difference between the solution vectors (velocity components) at (a) $\tau = 3.6$ and (b) $\tau = 4.4$ corresponding to the FT and the VT methods divided with the number of cells of the coarsest grid. The computational cells are: 25×125, 50×250, 75×375, 100×500 all with $\Delta\tau = 10^{-4}$ and 125×625 with $\Delta\tau = 5 \times 10^{-5}$. The flow parameters are as in Fig. 15.

by SL did not produce a converged solution, although this is not generally true. This flow topology where large drops of the inner phase travel within the outer phase, which remains continuous is called pulsing flow. The type of surface instability we observe here is the pressure-driven equivalent of the one that has been reported in two-phase Couette flow in the past [15,16,31,45]. Here we should mention that we are currently testing other upwinding schemes.

7.3. VT versus FT when the interface breaks at low Re

Next we present certain cases in which the two fluids have the same densities and viscosities, the Reynolds number is smaller than above, but the fraction of the volume of the core fluid is fairly small, $V = 0.04$, thus enhancing instability due to capillarity. In Fig. 15 we see the velocity field at time $\tau = 4.4$ as predicted either by FT (Fig. 15a) or VT (Fig. 15b). Similarly to Fig. 6, the minimum, maximum and intermediate values of the streamfunction computed by these two methods coincide to 3 significant digits, see values given in the figure caption. The interface is rather straight for its most part, but forms two maxima at the tube expansion in the left and right side of the picture separated by a deep minimum. Apparently the interface is about to break and the core fluid is about to loose its continuity. The convergence of both methods to the same solution is examined in Fig. 16. It shows the Euclidean norm of the difference of the two velocity components computed by our two algorithms first at $\tau = 3.6$, Fig. 16a, and then at $\tau = 4.4$, Fig. 16b. At the earlier time instant, it is demonstrated that the deviations in the results obtained with the two methods are very small even with the coarsest mesh and monotonically decrease with mesh refinement. At a later time however, when the interface has come very close to the axis of symmetry and the core fluid is about to loose its continuity forming a drop the deviations decrease monotonically with mesh refinement except for the finest mesh, where it increases ever so slightly always remaining of the order of 10^{-5} for both velocities. It is noteworthy that the two methods, VT and FT, calculate the interface force in a completely different way: VT according to Eq. (45) allowing for an error $O(\kappa\Delta x^2, \kappa\Delta y^2)$ where κ is the local curvature of the interface [20], i.e. the error increases with the curvature, and FT as explained in Section 5.4. Moreover, VT allows the interface to be discontinuous between adjacent cells. Both these facts introduce different errors to the two methods and a small deviation between the results obtained which increase when the interface force becomes important as the curvature increases significantly. This becomes clear at $\tau = 4.4$, but not at earlier times. We have made additional convergence tests by decreasing the time step for either the finest or the two finer meshes to $\Delta\tau = 10^{-5}$ and verified that what we observe in Fig. 16b is not

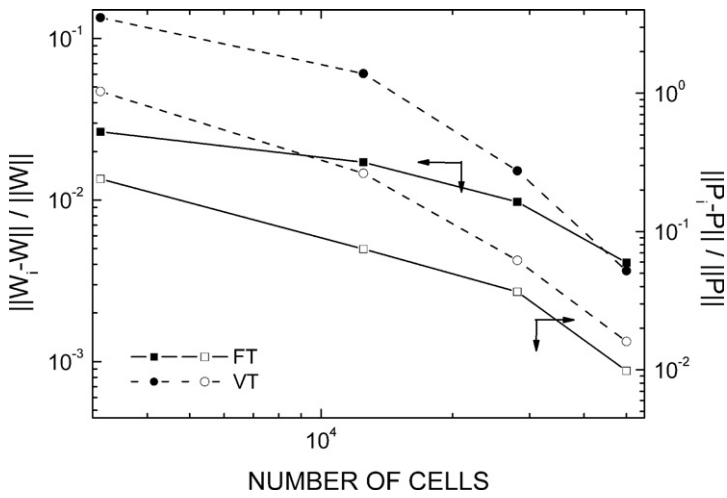

Fig. 17. The relative error of the axial velocity and the pressure, left and right vertical axis, respectively, for both FT and VT methods. As a reference value, W and P, the solution for a discretization with 125×625 computational cells is used and $\Delta\tau = 5 \times 10^{-5}$, while the vectors W_i and P_i are calculated using 25×125, 50×250, 75×375 and 100×500 cells all with $\Delta\tau = 10^{-4}$. Flow parameters are identical to those in Fig. 15.

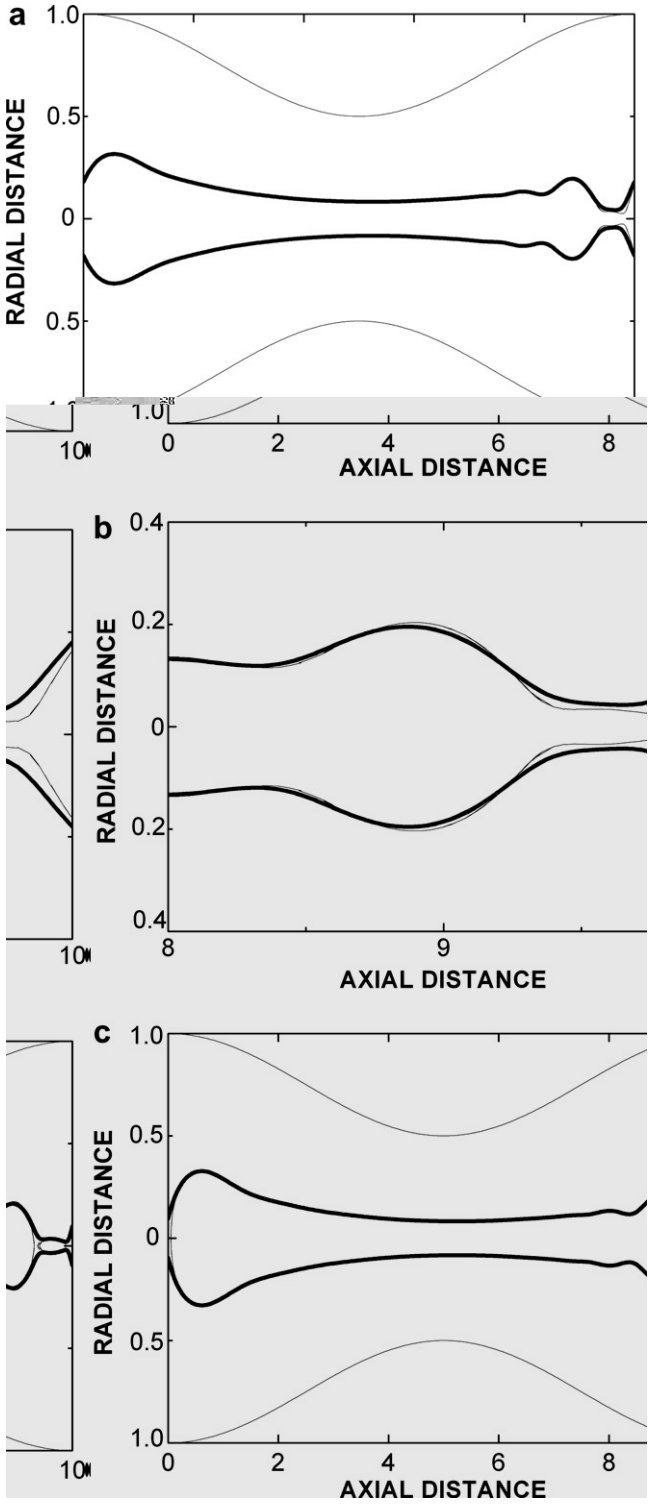

Fig. 18. Evolution of the interface for (a) $\tau = 4.4$, (b) $\tau = 4.4$ expanded view around the breaking point and (c) $\tau = 4.5$. The interface has been reconstructed by the VT method. The thick line corresponds to (75×350) computational cells in radial and axial directions respectively, while the thin line corresponds to (50×250) cells. The time step in both discretizations is $\Delta\tau = 10^{-4}$, and the parameter values identical to those in Fig. 15.

due to the choice of a larger than allowed time step. Fig. 17 shows the norm of the differences of the axial velocity and the pressure computed as described for Fig. 8. The error with FT is significantly smaller for the three coarser meshes than the error with VT, which with the coarser meshes gives rather inaccurate results. On the other hand, the error with VT drops faster in the fourth mesh. The CPU times required to compute

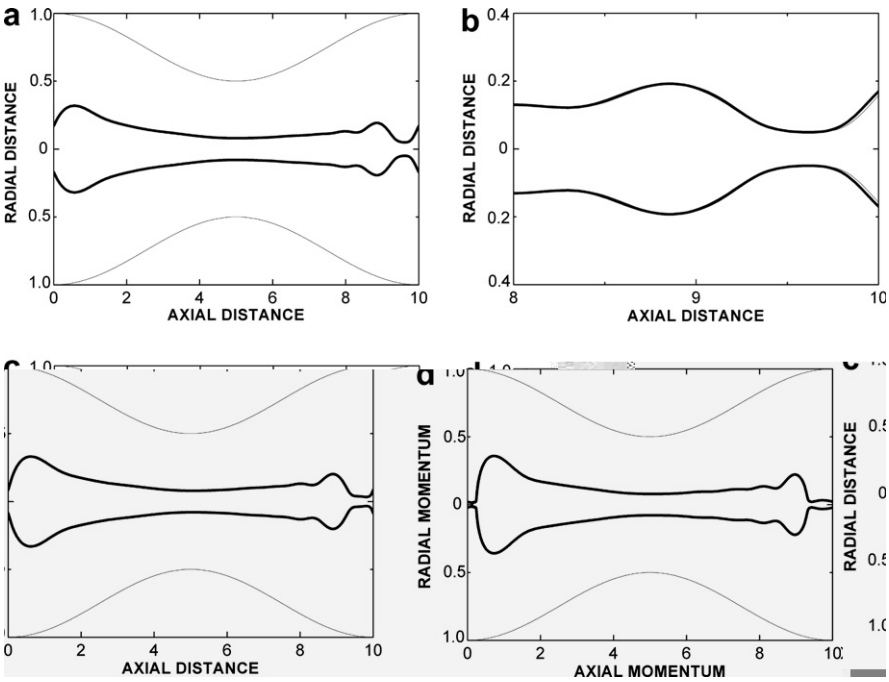

Fig. 19. Evolution of the interface for (a) $\tau = 4.4$, (b) $\tau = 4.4$ expanded view around the breaking point, (c) $\tau = 4.5$ and (d) $\tau = 4.66$. The interface has been reconstructed by the FT method and the dirac function is approximated according to Eq. (47). The thick line corresponds to (75×350) computational cells in radial and axial directions respectively and the thin line to (50×250) cells. The time step in both discretizations is $\Delta\tau = 10^{-4}$, the parameters are identical to those in Fig. 15.

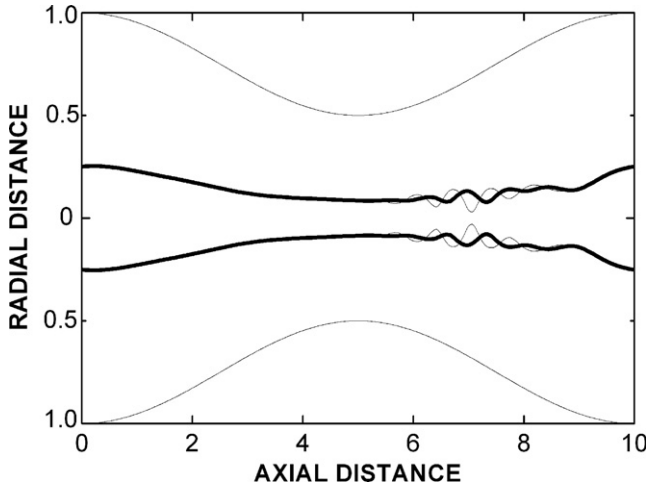

Fig. 20. Evolution of the interface for $\tau = 3.5$. The interface has been reconstructed by the FT method and the dirac function is approximated according to Eq. (48). The thick line corresponds to (75×350) computational cells in radial and axial directions respectively and the thin line to (50×250) cells. The time step in both discretizations is $\Delta\tau = 10^{-4}$, the parameters are the same with those in Fig. 15.

these results up to $\tau = 4.4$ using our VT algorithm range from 9 min for the 25×125 mesh to 10 h for the 125×625 mesh.

For the same parameter values, allowing the computations to proceed in time and closer examining the interface with intermediate meshes gives interesting information about the accuracy of the two methods. Fig. 18 gives two snapshots of the interface when it is reconstructed by the VT method. A long drop tends to form with larger radius at its two ends, which are away from the tube constriction. At time $\tau = 4.4$ and at axial distance $y = 9.6$ pinching is about to occur, but at that location convergence with mesh is less satisfactory than in the rest of the drop. This is more clearly seen in Fig. 18b, which is an expanded view of Fig. 18a, confirming once more the limited accuracy obtained with VT, when capillary forces increase. The less fine mesh, lighter line, predicts that the interface will breakup slightly earlier. Indeed at $\tau = 4.5$, Fig. 18c shows that the interface generated with the coarser mesh has broken, while the other one is still continuous. On the contrary, when we repeat the same calculation using our FT method and Eq. (47) to approximate the delta function, convergence with mesh refinement is better throughout the interface. Fig. 19 presents three such snapshots of the interface, where the two results for the two different meshes at the same time are indistinguishable, except for a small part of the blow up shown in Fig. 19b. Now we can approach the first

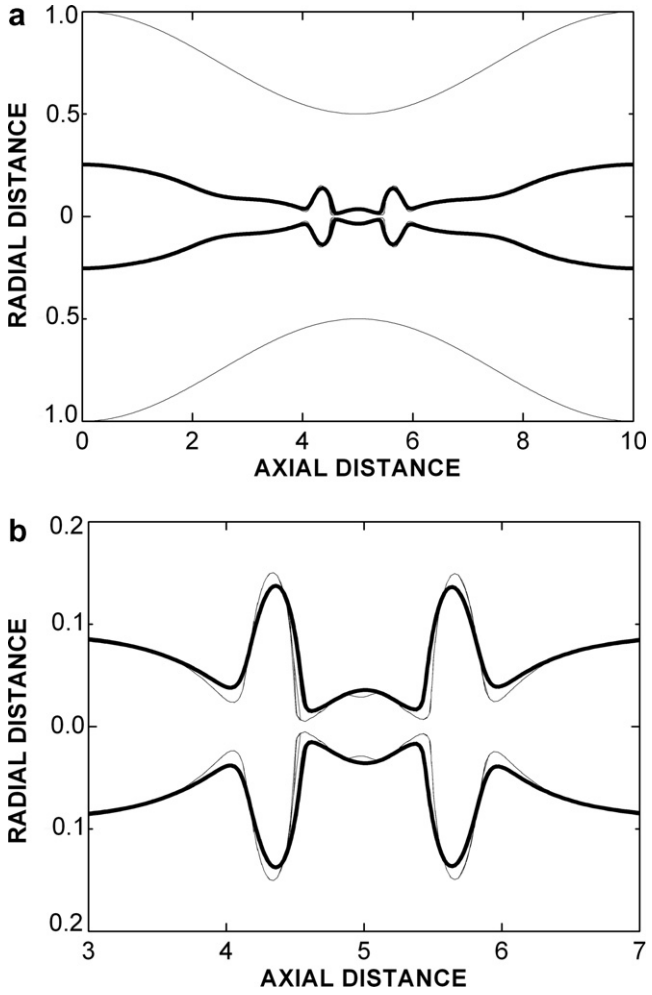

Fig. 21. Evolution of the interface for (a) $\tau = 0.163$, (b) $\tau = 0.163$, expanded view around the breaking point. The interface has been reconstructed by the VT method and the convection terms are approximated with upwinding using β according to SL. The thick line corresponds to (100×500) computational cells in radial and axial directions respectively and the thin line to (75×350) cells. The time step in both discretizations is $\Delta\tau = 10^{-5}$. $(\alpha, \Lambda, \mu^{o}, \rho^{o}, Re, We^{-1}, V) = (0.5, 10, 1, 1, 10, 100, 0.04)$.

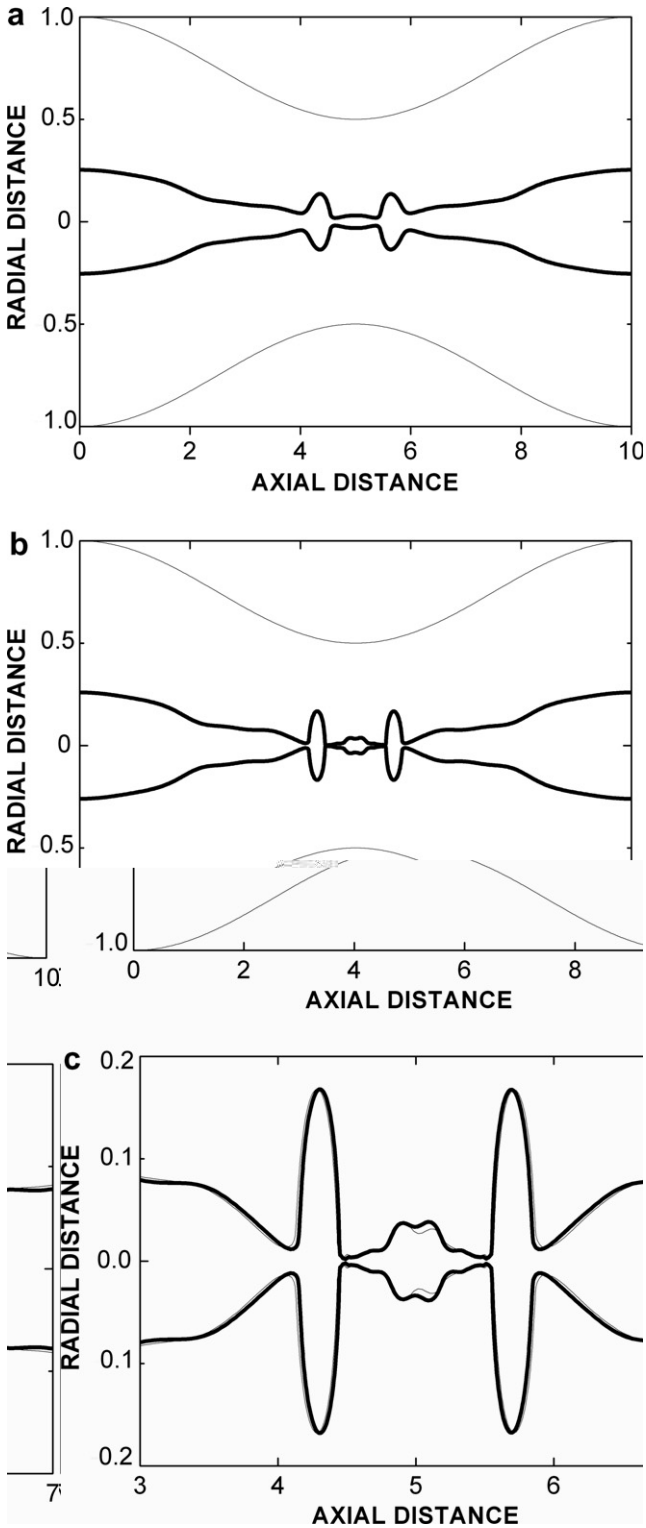

Fig. 22. Evolution of the interface for (a) $\tau = 0.163$, (b) $\tau = 0.17$ and (c) $\tau = 0.17$, expanded view around the breaking point. The interface has been reconstructed by the FT method and the convection terms are approximated with upwinding using β according to SL. The thick line corresponds to (100×500) computational cells in radial and axial directions respectively and the thin line to (75×350) cells. The time step in both discretizations is $\Delta\tau = 10^{-5}$, the parameters are identical to those in Fig. 21.

breakup point even closer, $\tau = 4.66$ in Fig. 19d. Additionally, we performed the same calculation using our FT code, but with Eq. (48) as an approximation of the delta function. Fig. 20 gives the interface with the two discretizations at the same time, but much earlier than before, $\tau = 3.5$. We clearly see that the finer mesh reduces the strong oscillations generated with the coarser mesh, where the neck is about to form, but the results are still not acceptable. This observation, when we approximate the dirac function by Eq. (48), is noticed primarily in cases where the ratio of the volume of the core fluid over the total volume of the tube is small.

We just saw that increasing the capillary force increases the difficulty in obtaining a converged solution, especially for the interface. To study this further, we kept the rest of the parameters the same as above, but increased We^{-1} by two orders of magnitude from 1 to 100. Both VT and FT calculations produced results with non physical oscillations, which did not converge with mesh refinement and are not shown for conciseness. It is anticipated that the much larger capillary force induces a faster breaking of the interface than before, which, of course, is accompanied by higher velocities. This would increase the importance of the convective terms in the momentum balances, although the Reynolds number is only 10 here. This led us to believe that our upwinding scheme should be introduced even in this case. In Fig. 21 we give the time evolution of the interface, which is reconstructed by VT coupled with upwinding, where the parameter β is calculated according to SL [38]. Fig. 21a corresponds to a time, $\tau = 0.163$, just before the first breaking of the interface, which seems to occur simultaneously in two locations. The result has converged, except near the necking. This is more clearly seen in Fig. 21b, where we give an expanded view of the previous figure near the interface for $-0.2 \leqslant x \leqslant -0.2$ and $3 \leqslant y \leqslant 7$. We observe that the results with the two discretizations do not coincide. On the other hand, when we reconstruct the interface by our FT method using the same upwinding as in Fig. 21, we obtained converged results between the two different mesh discretizations even a little later in time, but still just before breakup, Fig. 22. Comparing now the converged results for We^{-1} equal to 1 and 100, we observe that the maximum radial velocity increases by a factor of 10 and the maximum axial velocity by a factor of 35. This verifies our justification for using upwinding even here that the Reynolds number is rather small as also suggested by Eqs. (14) and (15). The convergence with mesh refinement of both VT and FT for this case is seen in Fig. 23. The very fast and simultaneous breaking of the interface in two neighboring points and the tendency to form two drops makes the VT rather inaccurate for any but the finest mesh. The FT method is always more accurate and converges faster with mesh refinement, although the two coarser meshes should be avoided for these parameter values. The CPU times required to compute these results up to $\tau = 0.17$ using our VT algorithm range from 21 sec for the 25×125 mesh to 23 min for the 125×625 mesh.

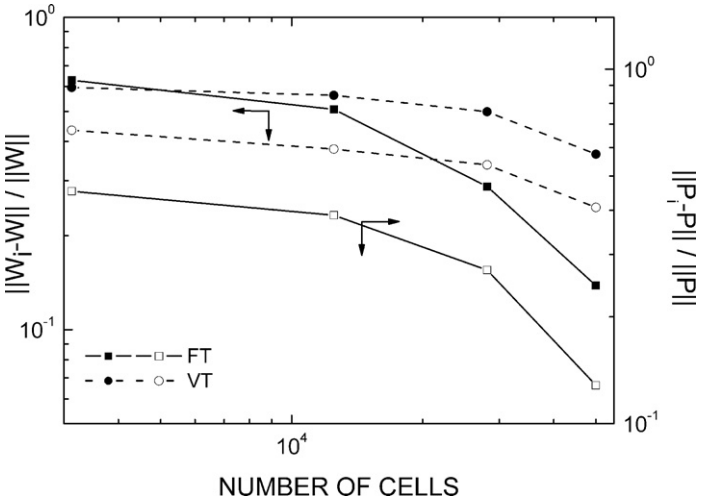

Fig. 23. The relative error of the axial velocity and the pressure, left and right vertical axis respectively, for both FT and VT methods. As a reference value, W and P, the solution for a discretization with 125×625 computational cells is used, while the vectors W_i and P_i are calculated using 25×125, 50×250, 75×375 and 100×500 cells. The parameter values are identical to those in Fig. 21.

According to the scenario we described above further increasing the We^{-1} should accelerate the breaking of the interface. Indeed, increasing We^{-1} by a factor of 2 to 200 decreases the time of the first breaking by nearly 35% from $\tau = 0.163$ to $\tau = 0.106$. Here again, combining our VT scheme with the same upwinding, gives results that are not fully converged with mesh refinement near the breakup point, Fig. 24. The coarser mesh predicts earlier breaking of the interface, but both meshes predict the formation of a long drop and two smaller ones. On the contrary, using the FT method with the same upwinding scheme gives a converged solution and finer details of the interface just before breakup, Fig. 25. Indeed, the expanded view in Fig. 25c shows that the core fluid tends to break and form a long drop spanning nearly all the length of the tube, two small drops near the two ends of the long drop and two even smaller drops in between them. The smaller drops seem to be formed first. All these events happen soon after the pressure gradient is applied and around the tube constriction. Therefore, for large values of We^{-1} and small volume ratio of the core fluid, upwinding is required in spite of the small Re and the FT method converges faster with mesh refinement than the VT method. So with FT a coarser grid can produce a converged solution resulting in savings in computational power and time. Alternatively, both methods and especially VT would yield results with increased accuracy, if the mesh is refined locally around the deforming interface as in [46], where the error is maximized due to using the

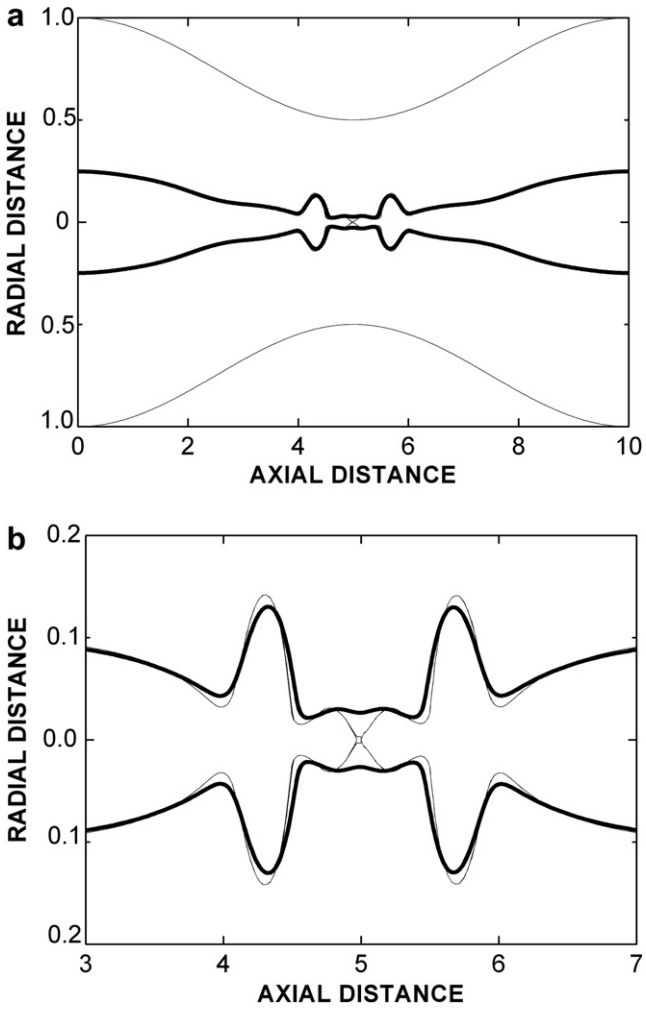

Fig. 24. Evolution of the interface for (a) $\tau = 0.106$, (b) $\tau = 0.106$, expanded view around the breaking point. The interface has been reconstructed by the VT method and the convection terms are approximated with upwinding using β according to SL. The thick line corresponds to (100×500) computational cells in radial and axial directions respectively and the thin line to (75×350) cells. The time step in both discretizations is $\Delta\tau = 10^{-5}$. $(\alpha, \Lambda, \mu^{o}, \rho^{o}, Re, We^{-1}, V) = (0.5, 10, 1, 1, 10, 200, 0.04)$.

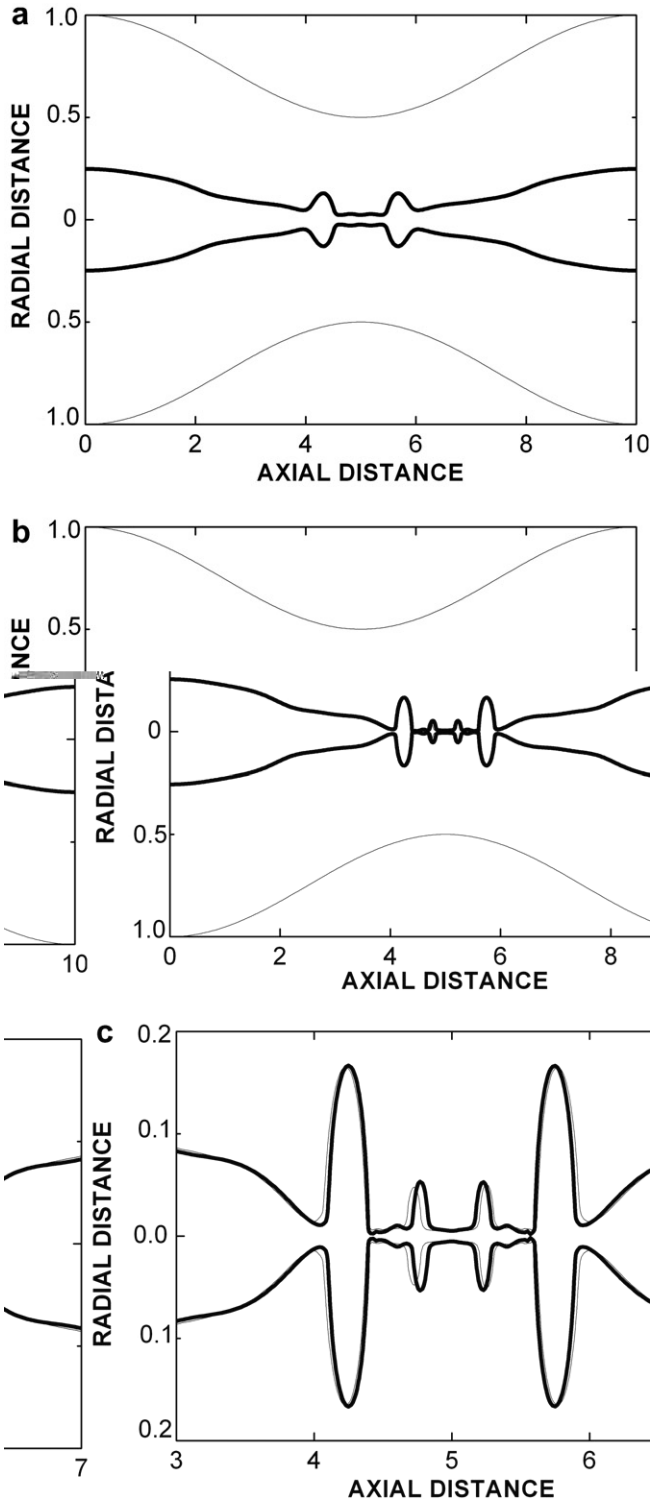

Fig. 25. Evolution of the interface for (a) $\tau = 0.106$, (b) $\tau = 0.112$ and (c) $\tau = 0.112$, expanded view around the breaking point. The interface has been reconstructed by the FT method and the convection terms are approximated with upwinding using β according to SL. The thick line corresponds to (100×500) computational cells in radial and axial directions respectively and the thin line to (75×350) cells. The time step in both discretizations is $\Delta\tau = 10^{-5}$, the parameters are the same with those in Fig. 24.

CSF method for calculating the surface tension force and even with a discontinuous representation of the interface (in VT).

Finally, we examined a case of the parameter values for which there is a single breaking of the interface leading to the creation of a large drop (slug) of the core fluid at the center of the tube. Such a case should arise for moderate values of the Re and We^{-1} numbers and rather large volume fraction of the core fluid. In Fig. 26 we draw the evolution of the total flow rate computed at the entrance of the tube. The continuous line and the dashed line are computed with VT and FT, respectively. Both lines are computed using 75 and 350 computational cells in the radial and axial directions respectively. For earlier times the two lines coincide, but later they seem to diverge somewhat. In Fig. 27 we see some snapshots of the interface evolution. In particular, in Fig. 27a we see the snapshot just before the first breaking of the interface. The continuous and the dashed interface are computed with VT and FT, respectively. We notice that the two interfaces coincide at $\tau = 3.0827$ except at the neck itself, which means that the interface breaks at almost the same time according to both methods. In Fig. 27b we focus on the neck of the interface ($-0.4 \leqslant y \leqslant 0.4$ and $8.2 \leqslant y \leqslant 8.6$) at the same time. Now it becomes clear that the result obtained using the FT method is smoother, whereas the calculation using VT exhibits some very small amplitude, but non physical oscillations very close to the breaking point. In Fig. 27c we have drawn the snapshots of the interface for a later time, $\tau = 21.3$, when the total flow rate has become periodic as seen in Fig. 26. Both methods predict the creation of a drop of core fluid which flows along with the annular fluid in the axis of symmetry of the tube. The two drops are slightly displaced with respect to each other, the one computed by FT (dashed line) follows the other one (continuous line). We also examined the convergence of both methods with mesh refinement. Indicatively, in Fig. 28, we present the locations and shapes of the drop after the computations have proceeded for a fairly long time, $\tau = 21.3$ and observe satisfactory agreement with mesh refinement for both methods.

7.4. Multigrid and LU factorization

We compared the two solvers, direct LU decomposition and iterative Multigrid, when the VT algorithm was used for the interface, in order to examine which one is provides a converged solution faster. First we examined a case of the core-annular flow where the two fluids have the same densities, $\rho^o = 1$, and moderate values for the rest of the parameters. We performed calculations with the 2 solvers in identical processors. After 24 h the code with the Multigrid solver reached $\tau = 5.35$, while the code with LU decomposition reached $\tau = 18.55$. We conclude that when the two fluids have the same densities and the matrix entries do not change

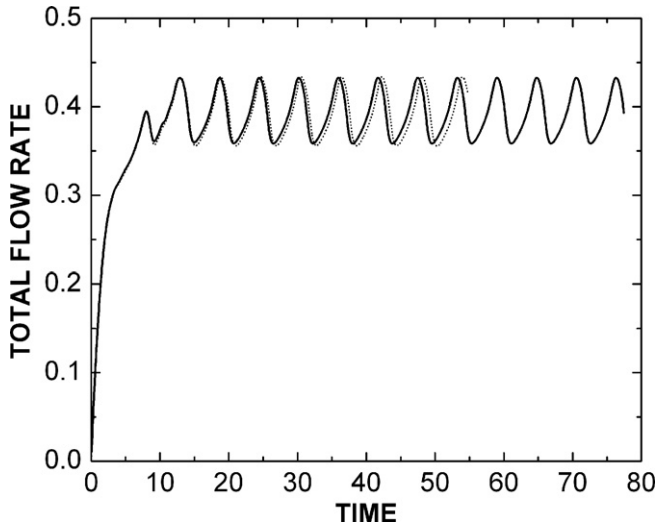

Fig. 26. Evolution of the total flow rate computed at the entrance of the tube. The continuous line and the dashed line are computed with the VT and FT methods, respectively. Both lines correspond to (75×350) computational cells in the radial and axial directions respectively. $(\alpha, \Lambda, \mu^o, \rho^o, Re, We^{-1}, V) = (0.5, 10, 1, 1, 50, 2, 0.49)$.

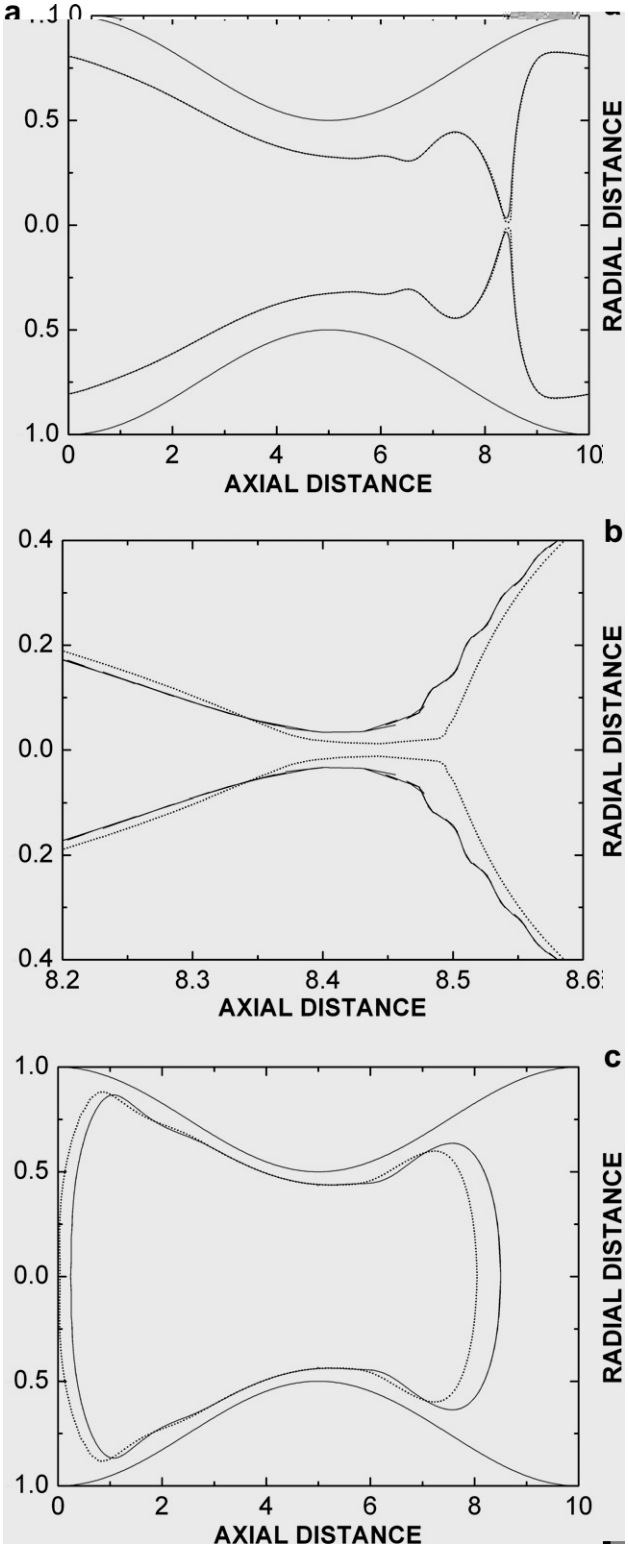

Fig. 27. Interface evolution for (a) $\tau = 3.0827$, (b) $\tau = 3.0827$, expanded view around the breaking point (c) $\tau = 21.3$. The continuous and the dashed interface are computed with VT and FT methods respectively. The results are obtained with (75×350) computational cells in the radial and axial directions respectively, the parameters are the same with those in Fig. 26.

because of that, then LU decomposition and repeated application of the backward substitution step is about 3.5 times faster than Multigrid. In Fig. 29 we present the evolution of the interface of this particular calculation. In each instant, the two interfaces computed using either Multigrid or LU decomposition, coincide, as they should. The interfaces have been computed with 128 and 512 computational cells in the radial and axial directions, respectively, which are powers of 2, as required by the Multigrid solver. In Fig. 29a taken at $\tau = 0.9$ the interface nearly follows the shape of the tube wall, while in Fig. 29b a neck tends to form on the interface at axial distance $y = 8.25$. Later on, in Fig. 29c the deformed interface moves downstream and its neck is thinner and appears at $y = 8.6$. In Fig. 29d the interface has broken up and a large drop of core fluid is formed along with a satellite drop, both of which travel along the center of the tube and eventually merge. After some time, a periodic flow has been established, seen in Fig. 29e at $\tau = 18.55$.

We also compared the Multigrid solver against LU decomposition with the VT algorithm, keeping all the parameters the same as in the previous case, except for the density ratio, which increases now to $\rho^o = 5$. Although we allowed both codes to 'run' in identical processors, now the one using Multigrid was about 10 times faster than the other one. Specifically, when the first code had reached dimensionless time $\tau = 100.0$ the second had reached only $\tau = 10.0$. In Fig. 30 we see the time evolution of the flow rate. Initially the flow

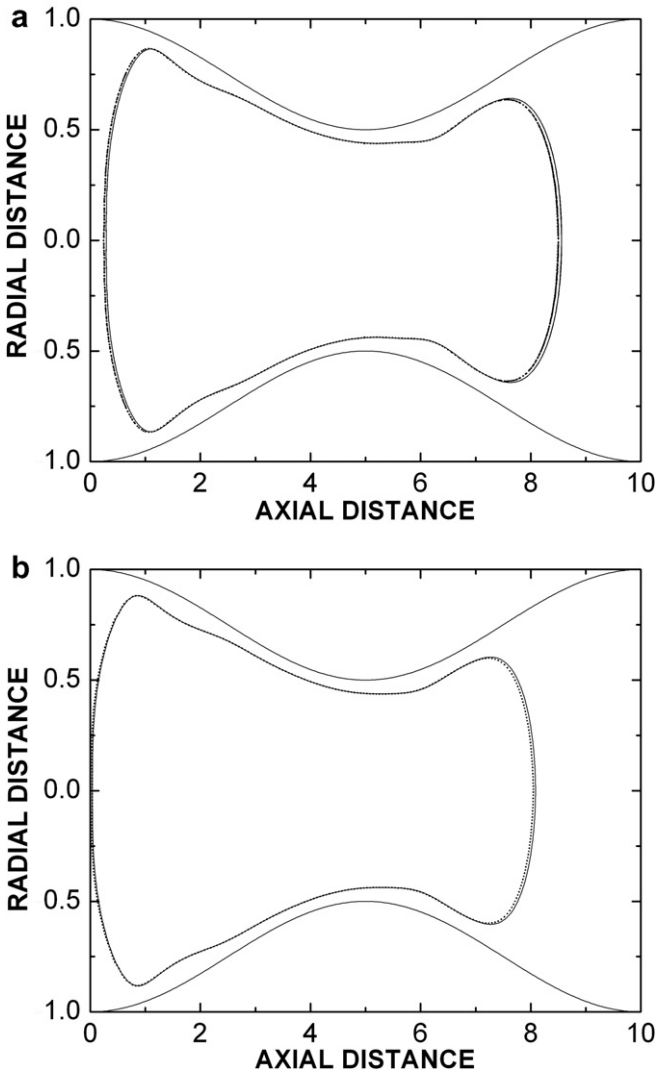

Fig. 28. Interface shapes at $\tau = 21.3$ using (a) VT and (b) FT. The continuous line corresponds to (100×500) computational cells in the radial and axial directions, respectively and the dashed line to (75×350) cells, the parameters are the same with those in Fig. 26.

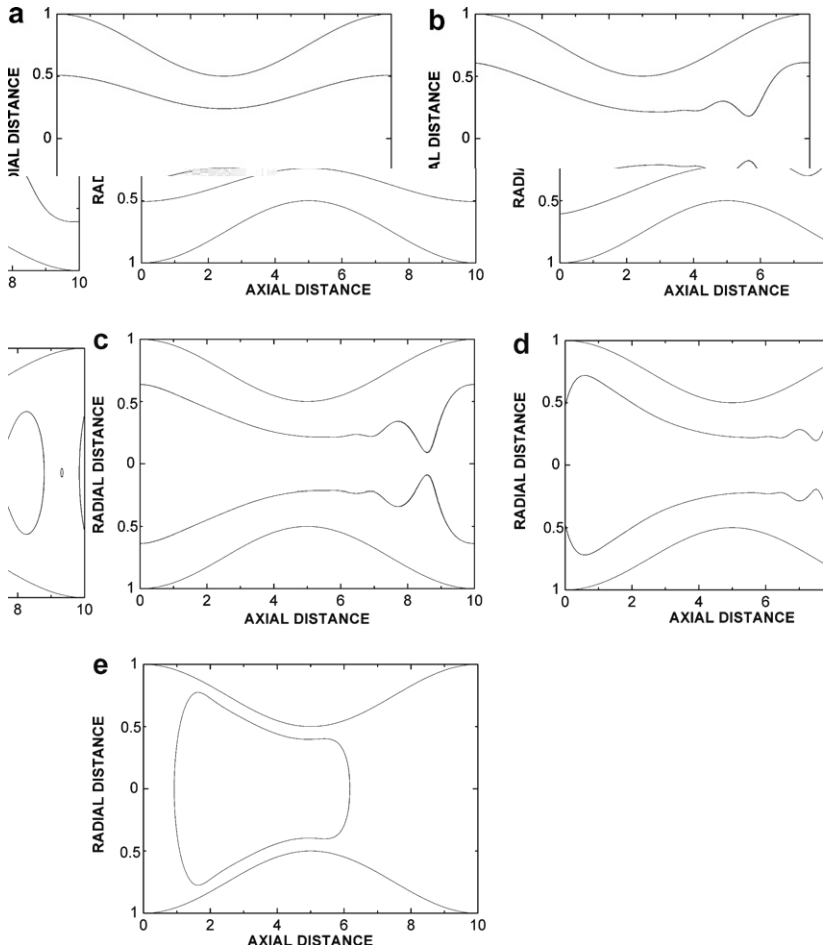

Fig. 29. Interface evolution for (a) $\tau = 0.9$, (b) $\tau = 3.55$, (c) $\tau = 4.0$, (d) $\tau = 5.35$, (e) $\tau = 18.55$. The interface has been constructed using VT with Multigrid and VT with LU decomposition. Both lines correspond to (128×512) computational cells in radial and axial directions respectively. $(\alpha, \Lambda, \mu^{o}, \rho^{o}, Re, We^{-1}, V) = (0.5, 10, 5, 1, 50, 1, 0.25)$.

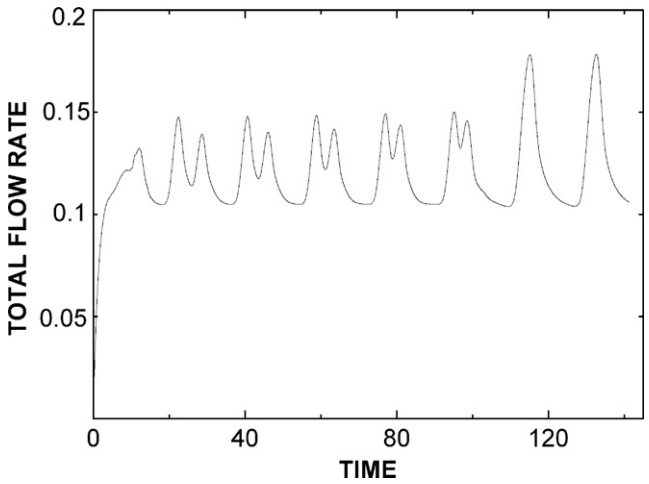

Fig. 30. Evolution of the total flow rate computed at the entrance of the tube. The line corresponds to (128×512) computational cells in radial and axial directions respectively. $(\alpha, \Lambda, \mu^{o}, \rho^{o}, Re, We^{-1}, V) = (0.5, 10, 5, 5, 50, 1, 0.25)$.

rate is zero but increases abruptly because of the externally imposed pressure gradient. From time 20 to 100, the flow rate fluctuates between the values 0.1 and 0.15, and has a double periodic variation. For time greater than 100, it oscillates between 0.1 and 0.18 and has a single period. In Fig. 31a, we observe that the interface again follows the shape of the tube, while at time $\tau = 3.95$ in Fig. 31b a narrow neck is created just beyond the axial

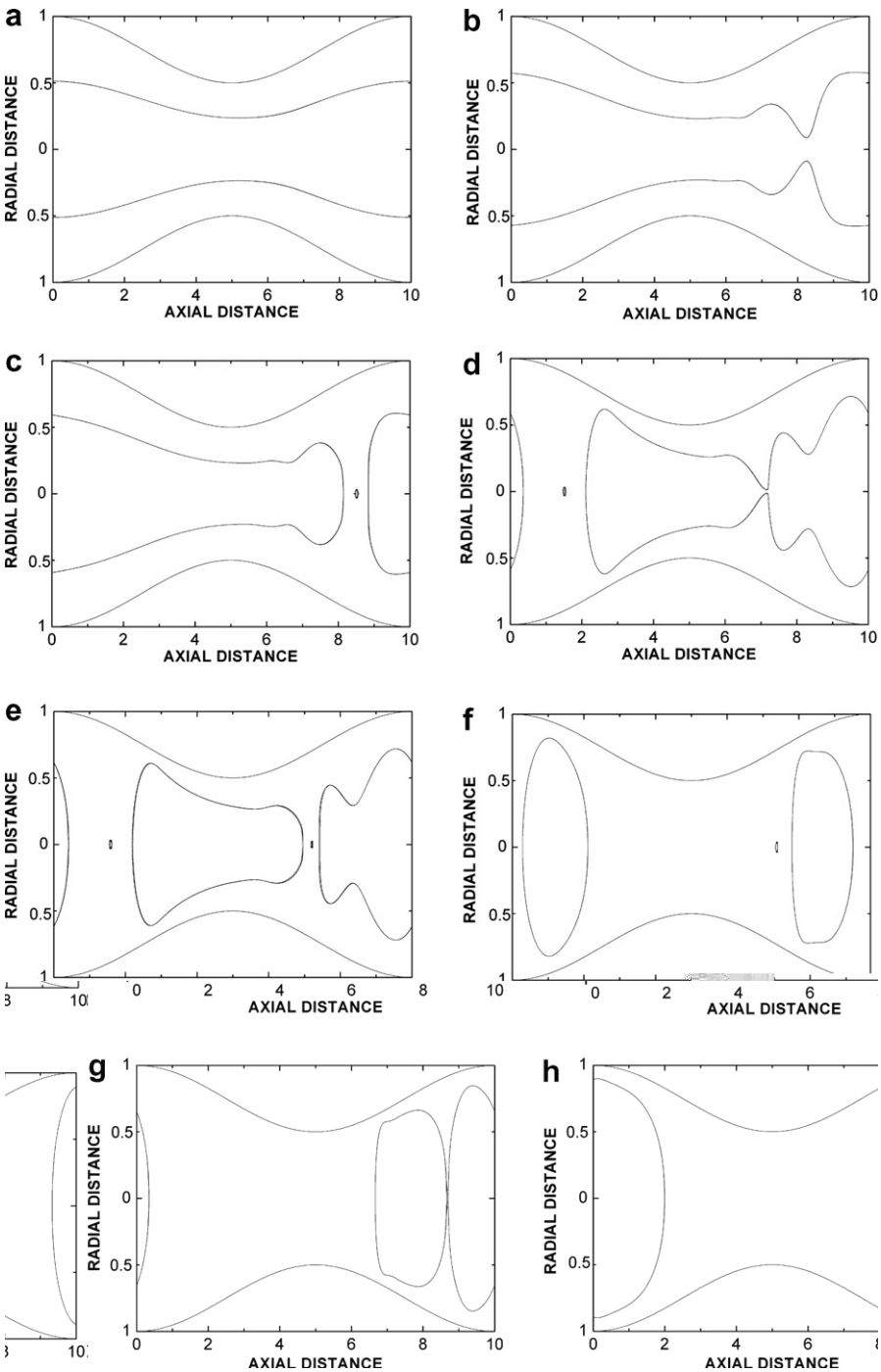

Fig. 31. Evolution of the interface for (a) $\tau = 1.55$, (b) $\tau = 3.95$, (c) $\tau = 4.3$, (d) $\tau = 10.3$, (e) $\tau = 10.45$, (f) $\tau = 15.4$, (g) $\tau = 100.84$ and (h) $\tau = 141$. The interface has been constructed using VT with Multigrid and VT with LU decomposition. Both lines correspond to (128×512) computational cells in the radial and axial directions respectively. The parameters are as in Fig. 30.

distance $y = 8.0$. After the first breaking of the interface, a large and a satellite drop of core fluid are formed, Fig. 31c. Subsequently, at time $\tau = 10.3$ in Fig. 31d we see a second breaking of the interface which leads to two large drops of core fluid and two smaller ones, Fig. 31e. It is the motion of these two large drops through the tube that gives rise to the doubly periodic variation of the flow rate. Later on and sequentially, the small drops merge with the large ones, Fig. 31f, while eventually, at $\tau = 100.84$ in Fig. 31g, the two large drops merge and lead to a single one, which travels in the center of the tube. Now the flow rate varies with a single period.

We should also make a remark related to the error due to the two methods of advecting the volume fraction with VT, given in Section 4.2. The first method was direct, but advected the volume of the rectangle adjacent to each corner of a cell in both the horizontal and the vertical directions, whereas the second method was iterative and split the rectangle into two triangles advecting the material in one of them horizontally and the material in the other one vertically. We examined the accuracy in computing the total volume fraction of the core fluid in various cases we studied and we consistently observed the following effect. In particular in the previous case, at $\tau = 10$ we found that the volume fraction of the core fluid had a relative error $O(10^{-5})$ when computed with the first method and a relative error $O(10^{-7})$, when computed with the second method. Both values are more than acceptable.

8. Conclusions

We developed a VT and an FT algorithm for calculating the two-phase flow in a constricted tube. Initially the fluids are stationary in a core-annular arrangement and they are set in motion by a constant pressure gradient. To solve the conservation equations with finite differences in this geometry requires the use of a boundary-fitted coordinate transformation. In order to accurately compute flows with relatively large Reynolds numbers and fast interface changes towards breakup, due to increased capillary forces, requires upwinding. We implemented the upwinding scheme of Ref. [37] for the approximation of the convective terms in the momentum equations and found that it eliminates the otherwise large variations of the interface and produces converged results with mesh refinement. Upwinding was more effective when its intensity is increased by using the β parameter suggested in SL [38] instead of that suggested in TF [37], as the importance of convective terms increases. When the volume of the core fluid is small compared to the total volume of the tube or the inverse *We* increases (capillary force increases), we obtain converged results when the interface is reconstructed by FT, but not always when we use the VT method. It seems that the inherent in VT surface discontinuity between cells leads to a less accurate calculation of the surface tension force. Producing a globally continuous interface upon reconstruction or refining the mesh locally around the interface could produce more accurate results at the cost of additional computational complexity and required CPU time. The latter would be necessary, if the scaling laws at interface pinching are to be extracted from our simulations Moreover, calculating more accurately the changes of the material properties as the interface is crossed or using other methods to compute the stress-jump condition could improve the accuracy of our results. Generally, our FT algorithm converges faster to the solution with mesh refinement than VT and, thus, can accommodate a less fine mesh. For density ratios different than 1, the Multigrid method is about 10 times faster than LU decomposition, while for density ratios equal to 1, LU decomposition is 3.5 times faster than Multigrid. Both the VT and FT schemes, carefully used, can generate a complete picture of the large variety of flow fields and fluid arrangements in this complex geometry that have been reported in relevant experiments.

Acknowledgment

This work was partially supported under the PENED 2001 program of the General Secretariat of Research and Technology of Greece (program number PENED 2001-01ED568).

References

[1] J.A. Deiber, W.R. Schowalter, Flow through tubes with sinusoidal axial variations in diameter, AIChE J. 25 (4) (1979) 638–645.
[2] A. Lahbabi, H.-C. Chang, Flow in periodically constricted tubes: transition to inertial and nonsteady flows, Chem. Engng Sci. 41 (10) (1986) 2487–2505.

[3] S. Pilitsis, A.N. Beris, Calculations of steady-state viscoelastic flow in an undulating tube, J. Non-Newt. Fluid Mech. 31 (1989) 231–287.
[4] B. Ramaswamy, S. Chippada, S.W. Joo, A full-scale numerical study of interfacial instabilities in thin-film flows, J. Fluid Mech. 325 (1996) 163–194.
[5] Ch. Kouris, J. Tsamopoulos, Dynamics of axisymmetric core-annular flow in a straight tube: I. The more viscous fluid in the core, bamboo waves, Phys. Fluids 13 (4) (2001) 841–858.
[6] Ch. Kouris, J. Tsamopoulos, Dynamics of axisymmetric core-annular flow: II. The less viscous fluid in the core, saw tooth waves, Phys. Fluids 14 (3) (2002) 1011–1029.
[7] Ch. Kouris, J. Tsamopoulos, Core-annular flow in a periodically constricted circular tube, I. Steady state, linear stability and energy analysis, J. Fluid Mech. 432 (2001) 31–68.
[8] Ch. Kouris, J. Tsamopoulos, Core-annular flow in a periodically constricted circular tube, II. Dynamics, J. Fluid Mech. 470 (2002) 181–222.
[9] Y. Dimakopoulos, J. Tsamopoulos, A quasi-elliptic transformation for moving boundary problems with large anisotropic deformations, J. Comput. Phys. 192 (2) (2003) 494–522.
[10] Y. Dimakopoulos, J. Tsamopoulos, Transient displacement of a Newtonian fluid by air in straight or suddenly constricted tubes, Phys. Fluids 15 (7) (2003) 1973–1991.
[11] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annu. Rev. Fluid Mech. 31 (1999) 567–603.
[12] J. Richards, A. Beris, A. Lenhoff, Steady laminar flow of liquid–liquid jets at high Reynolds numbers, Phys. Fluids A 5 (7) (1993) 1703–1716.
[13] J. Richards, A. Lenhoff, A. Beris, Dynamic breakup of liquid–liquid jets, Phys. Fluids A 6 (8) (1994) 2640–2655.
[14] J. Li, Y. Renardy, Direct simulation of unsteady axisymmetric core-annular flow with high viscosity ratio, J. Fluid Mech. 391 (1999) 123–149.
[15] A. Coward, Y. Renardy, M. Renardy, R. Richards, Temporal evolution of periodic disturbances in two-layer Couette flow, J. Comput. Phys. 132 (1997) 346–361.
[16] J. Li, Y. Renardy, M. Renardy, A numerical study of periodic disturbances of two-layer Couette flow, Phys. Fluids 10 (12) (1998) 3056–3071.
[17] J. Li, Y. Renardy, M. Renardy, Numerical simulation of breakup of a viscous drop in simple shear flow through a volume-of-fluid method, Phys. Fluids 12 (2000) 269–282.
[18] F. Mashayek, N. Ashgriz, Advection of axisymmetric interfaces by the volume-of-fluid method, Int. J. Numer. Meth. Engng. 20 (1995) 1337–1361.
[19] D.L. Youngs, Time-dependent multi-material flow with large fluid distortion, in: K.W. Morton, M.J. Baines (Eds.), Numerical Methods for Fluid Dynamics, Academic Press, New York, 1982, p. 273.
[20] W. Rider, D. Kothe, Reconstructing volume tracking, J. Comput. Phys. 141 (1998) 112–152.
[21] L. Chen, S.V. Garimella, J.A. Reizes, E. Leonardi, The development of a bubble rising in a viscous liquid, J. Fluid Mec. 387 (1999) 61–96.
[22] G. Chen, C. Kharif, S. Zaleski, J. Li, Two-dimensional Navier–Stokes simulation of breaking waves, Phys. Fluids 11 (1999) 121–133.
[23] Y. Renardy, M. Renardy, PROST: A parabolic reconstruction of surface tension for the volume-of-fluid method, J. Comput. Phys. 183 (2002) 400–421.
[24] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, St. Zaleski, Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows, J. Comput. Phys. 152 (1999) 423–456.
[25] J. Brackbill, D. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys. 100 (1992) 335–354.
[26] J. Wu, S. Yu, B. Jiang, Simulation of two-fluid flows by the least-squares finite element method using a continuum surface tension model, Int. J. Numer. Meth. Engng. 42 (1998) 583–600.
[27] Br. Lafaurie, C. Nardone, R. Scardovelli, St. Zaleski, G. Zanetti, Modeling merging and fragmentation in multiphase flows with SURFER, J. Comput. Phys. 113 (1994) 134–147.
[28] S.O. Unverdi, G. Tryggvason, A front tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys. 100 (1992) 25–37.
[29] Ch. Peskin, B. Printz, Improved volume conservation in the computation of flows with immersed elastic boundaries, J. Comput. Phys. 105 (1993) 33–46.
[30] 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 (2001) 708–759.
[31] W. Tauber, S. Unverdi, Gr. Tryggvason, The nonlinear behavior of sheared immiscible fluid interface, Phys. Fluids 14 (8) (2002) 2871–2885.
[32] M. Sussman, P. Smereka, Axisymmetric free boundary problems, J. Fluid Mech. 341 (1997) 269–294.
[33] S. Popinet, S. Zaleski, A front tracking algorithm for accurate representation of surface tension, Int. J. Numer. Meth. Fluids 30 (1999) 775–793.
[34] C.S. Peskin, The immersed boundary method, Acta Numer. (2002) 479–517.
[35] Sc. Shin, S. Abdel-Khalik, V. Daru, D. Juric, Accurate representation of surface tension using the level contour reconstruction method, J. Comput. Phys. 203 (2005) 493–516.
[36] G. Agresar, J. Linderman, G. Tryggvason, K. Powell, An adaptive, cartesian, front tracking method for the motion, deformation and adhesion of circulating cells, J. Comput. Phys. 143 (1998) 346–380.

[37] M. Tabata, S. Fujima, An upwind finite element scheme for high-Reynolds number flows, Int. J. Numer. Meth. Fluids 14 (1991) 305–322.
[38] P. Singh, L.G. Leal, Finite-element simulation of the start-up problem for a viscoelastic fluid in an eccentric rotating cylinder geometry using a third-order upwind scheme, Theor. Comput. Fluid Dyn. 5 (1993) 107–137.
[39] P. Wesseling, An Introduction to Multigrid Methods, John Wiley and Sons, New York, 1991.
[40] F.H. Harlow, J. E Welsh, Numerical calculation of time-dependent viscous incompressible flow with free surface, Phys. Fluids 8 (1965) 2182.
[41] B.P. Leonard, The QUICK algorithm: a uniformly third-order finite-difference method for highly convective flows, in: K. Morgan, C. Taylor, C.A. Brebbia (Eds.), Computer Methods in Fluids, Pentech Press, London, 1980, pp. 159–195.
[42] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes, The Art of Scientific Computing, Cambridge University Press, Cambridge, 1990.
[43] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201–225.
[44] A.J. Chorin, A numerical method for solving incompressible viscous flow problems, J. Comput. Phys. 2 (1967) 12.
[45] C. Pozrikidis, Instability of two-layer creeping flow in a channel with parallel-sided walls, J. Fluid Mech. 351 (1997) 139–165.
[46] N. Nikolopoulos, A. Theodorakakos, G. Bergeles, Three-dimensional numerical investigation of a droplet impinging normally onto a wall film, J. Comput. Phys. (2007), doi:10.1016/j.jcp.2006.12.002.