Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 192 (2003) 494–522
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A quasi-elliptic transformation for moving boundary problems with large anisotropic deformations

Yannis Dimakopoulos, John Tsamopoulos *

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

Received 31 March 2003; received in revised form 1 July 2003; accepted 25 July 2003

Abstract

We have developed a quasi-elliptic set of equations for generating a discretization mesh that optimally conforms to an entire domain that undergoes large deformations in primarily one direction. We have applied this method to the axisymmetric problem of the transient displacement of a viscous liquid by a high-pressure gas. The liquid initially fills completely a tube the diameter of which may be constant or change smoothly, suddenly or periodically along its finite length. Key ingredients for the success of the proposed transformation are limiting the orthogonality requirements on the mesh and employing an improved node distribution function along the deforming boundary. The retained orthogonal term along with the penalty method for the imposition of the boundary conditions overcome the inherent restrictions of a conformal transformation, producing meshes of high quality. This term also eliminates the discontinuous slopes of the coordinate lines that are normal to the free surface. These usually arise due to the harmonic transformation around highly deforming surfaces. The mathematical anisotropy in the mesh generating equations directly corresponds to the physical one, when the initially straight and normal to the axis of symmetry air/fluid interface, at the tube entrance, develops mostly along the axial direction, generating a long, open bubble. Moreover, the generalized node distribution imposes an optimal discretization of the bubble surface itself. Combining this scheme with the mixed finite element method produces a powerful tool, with extended robustness and accuracy. In order to reduce the computational cost, the resulting set of equations is solved with a non-linear, Gauss–Seidel technique and a variable time step. Specific applications of the proposed scheme are presented.

Keywords: Moving boundary problems; Mesh generation; Finite element methods; Liquid displacement by gas; Adaptive time-stepping

1. Introduction

We are interested in developing accurate and efficient numerical schemes for simulating the displacement of one fluid by another from a tube the diameter of which may be constant or may vary smoothly, suddenly or periodically along its finite length. The deployment of such prototype geometries is necessary in order to

* Corresponding author. Tel.: +30-2610-997-203; fax: +30-2610-998-178.
E-mail address: tsamo@chemeng.upatras.gr (J. Tsamopoulos).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2003.07.027

successfully analyze phenomena that take place in much more complicated, but often inaccessible geometries and, thus, such schemes would have numerous practical applications. For example, there is continuous interest in the displacement of oil from the ground using air or steam or various surfactants in the so-called ''enhanced oil recovery methods'' or in the way pollutants and water flow and mix in the ground. Similarly, the flow of air and vapor in the airway tubes of the lungs is modeled, aimed at exploring conditions to facilitate pulmonary airway reopening. Moreover, such studies may be useful in modeling potential accidents in nuclear reactors where light water may be displaced by gas after a steam explosion. Because of its important advantages over conventional injection molding [1], we are currently interested in a polymer processing operation, the so-called gas-assisted injection molding (GAIM) in which nitrogen or air at very high pressures are used to displace a polymer from a mold of a rather complex geometry.

Taylor [2] studied the steady development of a round-ended air column in a horizontal and extremely long tube; its length was equal to about a thousand tube radii. A systematic experimental investigation of this problem at even higher Capillary numbers (up to 10) was performed by Cox [3,4]. Motivated by the slow, two-phase flow in channels of microscopic dimensions, where the driven fluid is more viscous than the driving one and the capillary forces are important, Bretherton [5] developed an approximate solution by, in effect, the method of matched asymptotic expansions and performed experiments to validate it. Bretherton's analysis has been thoroughly clarified and extended by Park and Homsy [6], while studying fluid displacement in a Hele Shaw cell. Although fluid displacement by air in a capillary has been studied experimentally for many years, its theoretical study has been restricted to simplified or steady state models. The most common and convenient model for analyzing this phenomenon assumes steady motion of a semi-infinite bubble in a tube of infinite length. Reinelt and Saffman [7] used finite differences to solve the Stokes equations in a channel between two parallel plates or in a circular tube. To discretize the physical domain, they applied a combination of two different grids: one curvilinear conformal to the free surface and another one rectilinear parallel to the straight boundaries of the confining walls. The overlapping grids were stretched so that the number of grid points was greater in regions where they were needed the most. Interpolation was used to connect the solution between the two grids. A two-stage procedure was applied for the solution of the problem. Beginning with an initial guess for the shape of the finger, a system of equations equivalent to the biharmonic equation was solved. Having calculated the flow field, the shape of the finger was updated to satisfy the normal-stress boundary condition and so on. Even so, numerical results were reported in the range of $10^{-2} < Ca < 2$ only. Beyond this range, the applied iterative scheme failed to converge. Shen and Udell [8] presented a solution to the same problem, but instead of finite differences, they followed the finite element technique, with a rather uniform mesh of points. Finite element solutions of the momentum and continuity equation, using also a two-stage procedure, were obtained only for $5 \times 10^{-3} < Ca < 2 \times 10^{-1}$. These authors attempted to find solutions for capillary numbers greater than 0.2, but their results suffered from oscillations along the meniscus profile. They suggested that the oscillations were probably caused by insufficient resolution of the flow field near the bubble front and in particular near the flow stagnation point(s). More recently, Giovedoni and Saita [9] solved simultaneously for the flow field and the bubble shape using the spine parameterization introduced by Kistler and Scriven [10]. Inertial effects were taken for the first time into consideration. Converged solutions were obtained for Reynolds numbers between 0 and an upper limit, which depended on Ca, whereas Ca was varied between 10^{-4} and 10.

Very few researchers have examined the transient penetration of a bubble in a liquid-filled tube. The main reason is the lack of robust methods for constructing meshes that highly deform with time [11]. An attempt to solve a similar problem was made by Frederiksen and Watts [12]. They studied both the fluid entrainment by a moving plate and the cavity flow problems using a classical finite element approximation for the position of the internal nodes. Although their meshes were very coarse, their calculations approximated satisfactorily the actual phenomena. However, for small values of the capillary number saw tooth instabilities covered the whole free surface of the viscous liquid. Coating flows were also examined by

Sackinger et al. [13], in an attempt to test the efficiency of their combination of finite element and pseudo-solid methods. Despite the high resolution of the contact line region there was a large shifting of the nodes along the moving wall in dip coating, probably caused by inappropriate boundary conditions on their grid generation scheme. More recently, they have advanced their methodology by performing three-dimensional calculations [14,15].

Accurate simulation of moving boundary flows is a difficult and demanding task, because computing a succession of transient states may require reconstruction of the mesh several times as the flow domain changes in shape and size. The coupling between the instantaneous location of the interface and the field variables introduces additional non-linearities through the boundary conditions imposed there. This may be the reason that the full dynamics of the processes we are interested in have not been examined thoroughly. The available numerical approaches can be classified as Lagrangian, Eulerian, and mixed Lagrangian–Eulerian based on the method generating the computational grid. Only Eulerian techniques provide easy handling of large interfacial distortions, because the grid points remain stationary or move in a predetermined manner, whereas the fluid moves in and out the computational cells. This is achieved at the expense of an interface that is not ''sharply'' defined and, consequently, the boundary conditions on it are not as accurately approximated as the rest of the field equations. Instead, the interface is calculated based on the values of the ''color function'' by interpolating between cells, which requires very fine meshes to achieve certain accuracy and makes the computations very expensive, especially when the density ratio between the two fluids is different from unity. The surface tracking method [16] is a significant improvement for Eulerian methods because it determines the interface explicitly using a separate unstructured grid.

In the Lagrangian approach, the grid moves with the local fluid velocity (and the interface), which requires frequent remeshing, but provides a sharp definition of the interface even at large deformations and irrespective of the fluid properties. Finally, in ALE (arbitrary Lagrange–Euler) methods, kinematic descriptions combine the two previous approaches, but still need a detailed technique for the mesh movement. The Lagrangian aspect of the method involves interpolation of the solution from the old to the new mesh, which is a nontrivial and diffusive operation. This difficulty is alleviated by employing a curvilinear coordinate system that conforms to the moving boundary. The evolving physical domain is mapped onto a simple and time-independent computational one in which the moving boundary coincides with one of the coordinate surfaces and in which it is trivial to generate the mesh. It is fairly easy to perform this boundary-conforming mapping using simple algebraic relations [10,17,18]. However, this technique is restricted to relatively simple initial shapes and small deformations and requires that the mapping function is single-valued.

Alternatively, elliptic mesh generation schemes in two dimensions have been proposed [19–21], which produce the mapping between the two domains by solving two nonlinear PDEs. In spite of this additional computational cost, the technique is preferred when more involved geometries arise, because it does not have the limitations of the simpler algebraic methods and allows writing general codes. The quality of the mesh to be constructed is based on the type of the mapping equation. Christodoulou and Scriven [20] extended the earlier work by Ryskin and Leal [19] to develop a system of differential equations capable of generating quasi-orthogonal grids, by minimizing a functional which quantifies the deviation of the mesh from an orthogonal one that satisfies generalized Cauchy–Riemann equations. The main drawback of this scheme was that by applying the same PDE in both coordinate directions it did not allow independent control of mesh spacing in each direction. This was not overly restrictive, while applying this scheme for the analysis of the slide coating flow, where the physical domain did not deform too much or preferably in one of the directions. In contrast to this, Tsiveriotis and Brown [21] had to solve a free boundary problem in directional solidification, where the free boundary was quite deformed and preferentially in one direction. Typically one phase penetrated the other at distances up to three times its cross-section. They overcame this additional difficulty by relaxing the contribution from orthogonality in the normal to the interface direction. Along the interface, nodes were equally distributed, while at the other boundaries reflective boundary

conditions were used, allowing in this way the motion of the boundary nodes according to the interface deformation. To improve the accuracy of the solution on the interface, Tsiveriotis and Brown [22] proposed a two- to one-element splitting scheme, for a transition from a smaller number of elements in the bulk to a larger one close to the interface. However, this splitting resulted in elements with great aspect ratios and the generated grid lacked homogeneity; there were regions with extremely high concentration of elements and others where the grid was rather coarse. Recently, Helenbrook [11] tried to overcome the inherent restrictions of the "direct" harmonic transformation, using a biharmonic operator. This allowed him to specify two boundary conditions on each side of the computational domain, but as he mentioned, its main disadvantage was the increased computational expense (about four times the cost of employing the harmonic operator), which makes it unsuitable for large-scale calculations.

In our transient simulations we have to generate successive meshes that allow the bubble front to eventually penetrate inside the liquid at distances even larger than 12 times the tube radius. Having thoroughly examined the performance of the above-mentioned mesh generating techniques in our problem, we found out that they were not satisfactory for such large deformations. Thus we developed a new elliptic mesh generation technique, the key features of which are presented in Section 3 along with the finite element method for solving the complete problem. Before that, in Section 2, we present the governing equations, which depend on two dimensionless parameters, the modified Reynolds number and the applied by the gas pressure normalized by capillary forces, and various geometric ratios. Results are presented in Section 4, where fluid displacement in straight, constricted, complex, and undulated tubes is discussed. Finally, conclusions are drawn in Section 5.

2. Problem formulation

A cross-section of a prototype tube along its axis of symmetry is shown schematically in Fig. 1. In its most general representation, the tube can be split in its entrance and exit regions, of different diameters each, in its expansion and contraction regions, here represented by segments of sinusoidal curves and its main region, in the middle. Each region can have different lengths. The radii of the introductory, the

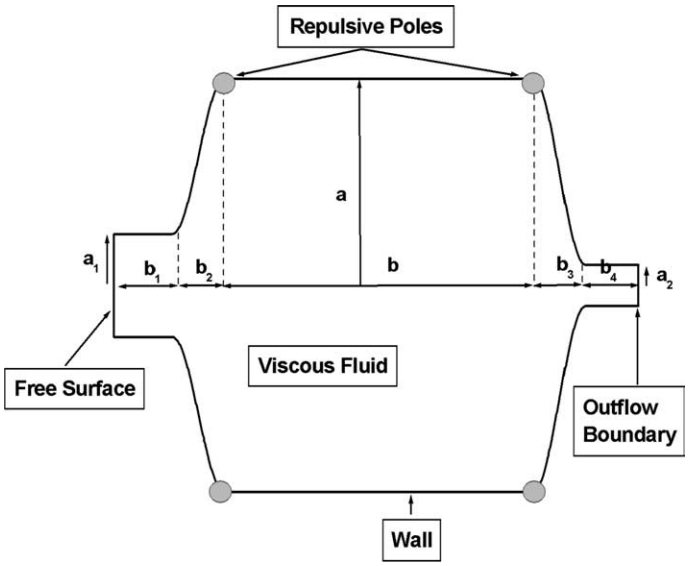

Fig. 1. Schematic of a complex tube initially filled with fluid, which is displaced by high-pressure air.

primary and the exit tubes are denoted as a_1, a, and a_2, respectively, and their corresponding lengths as b_1, b, and b_4, respectively. The lengths of the expansion and the contraction regions are equal to b_2 and b_3, respectively. Thus, the radius of this tube along its length is given by

$$S_r(Z) = \begin{cases} a_1, & 0 \leqslant \bar{Z} \leqslant \bar{Z}_1, \\ f(Z) \equiv a_1 + (a - a_1)\sin\left[\frac{\pi}{2}\frac{\bar{Z}-\bar{Z}_1}{\bar{Z}_2-\bar{Z}_1}\right], & \bar{Z}_1 \leqslant \bar{Z} \leqslant \bar{Z}_2, \\ a, & \bar{Z}_2 \leqslant \bar{Z} \leqslant \bar{Z}_3, \\ g(Z) \equiv a_2 + (a_2 - a)\sin\left[\frac{\pi}{2}\frac{\bar{Z}-\bar{Z}_3}{\bar{Z}_4-\bar{Z}_3}\right], & \bar{Z}_3 \leqslant \bar{Z} \leqslant \bar{Z}_4, \\ a_2, & \bar{Z}_4 \leqslant \bar{Z} \leqslant \bar{Z}_5, \end{cases} \tag{1}$$

where $\bar{Z}_1 = b_1$, $\bar{Z}_2 = \bar{Z}_1 + b_2$, $\bar{Z}_3 = \bar{Z}_2 + b$, $\bar{Z}_4 = \bar{Z}_3 + b_3$, $\bar{Z}_5 = \bar{Z}_4 + b_4$ and an overbar indicates a dimensional quantity. From this prototype geometry one can generate all the others that will be examined in this work. For example, by setting $b_1 = b_2 = b_3 = b_4 = 0$ we obtain a straight tube and by setting $b_1 = b_2 = b_3 = 0$ we obtain the suddenly constricted tube, etc.

Initially, the tube is filled completely with a viscous liquid, which forms a straight interface with a gas at the tube entrance. The fluid is incompressible and Newtonian, of constant viscosity μ and density ρ and sticks to the tube wall, forming a three-phase contact line at the entrance of the tube, which is located in the left hand side of the tube cross-section in Fig. 1. At start up, the pressure in the air at the tube entrance is increased abruptly from zero to $\bar{P}_{\text{ext}}$, whereas the pressure at the tube exit remains at zero and displacing of the fluid is initiated. This change causes a continuous deformation of the free surface, the position vector of which is given by

$$\underline{\bar{F}} = \bar{R}\underline{e}_r + \bar{Z}\underline{e}_Z, \tag{2}$$

in the primitive representation relative to a cylindrical coordinate system centered at the axis of symmetry of the tube, where $(\underline{e}_r, \underline{e}_\theta, \underline{e}_Z)$ are the corresponding unit vectors. The fluid pressure is denoted by $\bar{P}(\bar{R}, \bar{Z}, \bar{\tau})$ and the radial and axial component of the velocity vector in the fluid by $\bar{v}_r(\bar{R}, \bar{Z}, \bar{\tau})$ and $\bar{v}_Z(\bar{R}, \bar{Z}, \bar{\tau})$, respectively,

$$\underline{\bar{v}} = \bar{v}_r(\bar{R}, \bar{Z}, \bar{\tau})\underline{e}_r + \bar{v}_Z(\bar{R}, \bar{Z}, \bar{\tau})\underline{e}_Z. \tag{3}$$

We scale lengths with the radius of the primary tube, a, but, due to the accelerating nature of the flow, no characteristic velocity exists. Instead, we scale pressure with the externally applied, but constant air pressure, $\bar{P}_{\text{ext}}$, and derive a characteristic velocity assuming that Poiseuille flow takes place in a straight tube of radius a and length L_{eq} equal to the total length of the examined geometry, for example, $L_{\text{eq}} = b + b_4$ in a constricted tube, $v_{\text{ch}} = (a^2\bar{P}_{\text{ext}})/(\mu L_{\text{eq}})$. Thus the resulting dimensionless groups are a modified Reynolds number, which compares the applied pressure to viscous forces and the dimensionless applied pressure relative to the capillary force. They are defined as

$$Re_p = \frac{\rho a^3 \bar{P}_{\text{ext}}}{L_{\text{eq}}\mu^2}, \quad P_{\text{ext}} = \frac{a^2\bar{P}_{\text{ext}}}{\sigma L_{\text{eq}}}. \tag{4}$$

In these definitions, σ is the surface tension of the air–liquid interface. In all geometries to be examined the aspect ratio, $\varepsilon = a/L_{\text{eq}}$ will arise along with additional ratios of length scales in specific geometries. For example, in a suddenly constricted tube we will have also the contraction ratio, $r_c = a_2/a$ and the aspect ratio of the main tube, a/b.

The geometry in Fig. 1 has several points that may pose additional computational difficulties. They are the three-phase contact line at the entrance of the tube, the expansion and contraction corners, and the recirculation corners. It is well known that points where abrupt changes in geometric shape (i.e.

contraction corner) and/or boundary conditions (contact line) take place are responsible for a singular behavior of the stress tensor. Moreover, the expanding portion of the tube will provide additional area for the bubble to move to, introducing additional complications to its shape. To handle numerically the non-integrable stress singularity or the unexpectedly deforming bubble, a careful choice of boundary conditions and numerical methods is needed. Although in the present formulation the contact point is stationary due to fluid adherence at the tube entrance, fluid flow does take place in its vicinity which can be computed only when the capillary force exerted on the moving surface is below a certain limit. Beyond that value, all field variables are inaccurate or do not converge. This is first observed in the interface shape, which develops a saw tooth shape between successive surface nodes. The resolution of the flow field in such cases requires special treatment near the contact point, which is beyond the scope of the present study, since in GAIM capillary forces are clearly subdominant to the applied pressure as well as to viscous forces.

The Navier–Stokes equations subject to the constraint of divergence – free velocity field, for an incompressible fluid, govern the transient motion of a viscous fluid within the tube

$$Re_p \frac{D\underline{v}}{D\tau} - \nabla \cdot \underline{\underline{\sigma}} = 0, \tag{5}$$

$$\nabla \cdot \underline{v} = 0, \tag{6}$$

where τ is the dimensionless time, $D/D\tau = \varepsilon(\partial/\partial\tau) + \underline{v} \cdot \nabla$ is the material derivative, $\nabla = (\partial/\partial R)\underline{e}_r + (1/R)(\partial/\partial\Theta)\underline{e}_\theta + (\partial/\partial Z)\underline{e}_Z$ is the gradient operator and $\underline{\underline{\sigma}} = -\varepsilon^{-1}P\underline{\underline{I}} + (\nabla\underline{v} + (\nabla\underline{v})^{\mathrm{T}})$ is the total stress tensor. Along the free surface the velocity field should satisfy a local balance between viscous stresses, $\underline{n} \cdot \underline{\underline{\sigma}}$, surface tension, $2H\underline{n}/P_{\mathrm{ext}}$, and gaseous pressure, $\varepsilon^{-1}\underline{n}$:

$$\underline{n} \cdot \underline{\underline{\sigma}} = \frac{2H}{P_{\mathrm{ext}}}\underline{n} - \frac{1}{\varepsilon}\underline{n}, \quad 2H = -\nabla_s \cdot \underline{n}, \quad \nabla_s = (\underline{\underline{I}} - \underline{n}\underline{n}) \cdot \nabla. \tag{7}$$

In Eq. (7), $2H$ is twice the local mean curvature of the free surface; $\underline{n}$ is its inward unit normal, ∇_s is the surface gradient operator, and $\underline{\underline{I}}$ is the identity tensor. Taking the inner product between Eq. (7) and the unit tangent, $\underline{t}$, we get the tangential stress balance

$$\underline{t}\underline{n} : \underline{\underline{\sigma}} = 0, \tag{8}$$

while acting similarly with the normal vector, Eq. (7) yields

$$\underline{n}\underline{n} : \underline{\underline{\sigma}} = \frac{2H}{P_{\mathrm{ext}}} - \frac{1}{\varepsilon}. \tag{9}$$

The transient free surface shape is a material surface provided that there is no mass transfer across it

$$\frac{D\underline{\underline{F}}}{D\tau} = \underline{v}. \tag{10}$$

The velocity field must be also bounded at the centerline, and all variables must be symmetric around it

$$\underline{n} \cdot \underline{v} = 0, \quad \underline{t}\underline{n} : \underline{\underline{\sigma}} = 0, \quad F_R = 0. \tag{11}$$

Finally, the velocity field must be zero along the tube wall

$$v_r = 0, \quad v_Z = 0, \tag{12}$$

and fully developed far downstream, where we apply hard boundary conditions [23]

$$v_r = 0, \quad \frac{\partial v_Z}{\partial Z} = 0. \tag{13}$$

The mathematical statement of the problem is completed by specifying the initial conditions. Initially, fluid is assumed to occupy all the interior of the tube, to be under constant pressure $\bar{P}(\bar{R}, \bar{Z}, 0) = 0$, and to be quiescent $\underline{\bar{v}}(\bar{R}, \bar{Z}, 0) = \underline{0}$. Its free surface in contact with the air is assumed to be flat and located at the entrance of the tube.

3. Numerical solution

For reasons related to accuracy, flexibility, and robustness of the numerical algorithm, the resulting set of equations is solved by a combination of an elliptic grid generation scheme and the mixed finite element method. The use of the finite element method guarantees the accuracy and the stability of the numerical results in each time step, while the elliptic mesh generation offers a very robust way of constructing a mesh of points even in very deformed and evolving domains. The principle of this numerical approach has been used [20,21] before, albeit in much less deformed or deforming domains.

3.1. Elliptic grid generation

When the grid must evolve with the solution at each time step, an inherent requirement of moving boundary problems, a succession of mesh reconstructions, followed by interpolation of the solution vector from the old grid to the new one must be used. To overcome this restriction, a modification of Winslow's equations has been developed [20,21,24]. The new set of mapping equations is capable of dynamically adapting the grid points so as to move in response to the developing free surface. The point distribution over the domain is thus readjusted dynamically to keep a relatively uniform concentration of points in the whole physical domain, without relying on prior knowledge of the interface location or placing unphysical restrictions on its shape. The physical domain is mapped onto a fixed in time computational one, wherein a simple mesh tessellation is generated. In the present problem, we chose as computational domain the volume occupied by the viscous fluid initially. The mapping is defined as follows:

$$(R, Z, \tau) \xrightarrow{J} (\xi, \eta, \hat{\tau}). \tag{14}$$

More specifically, every fluid particle being at time τ at a position with coordinates (R, Z) is transformed in a new plane with global coordinates (ξ, η). The new independent variables vary in the range $0 \leqslant \xi \leqslant 1$, $0 \leqslant \eta \leqslant \varepsilon^{-1}$, $0 \leqslant \hat{\tau}$, for a straight tube, and appropriately modified for every other specific tube geometry.

Saltzman and Brackbill [24] with their pioneering work set the fundamental principles for the construction of a quantitative mesh generation scheme. According to them, the three necessary characteristic terms to describe quantitatively a mapping are: smoothness of the mapping, orthogonality and concentration of the coordinate curves in the domain [25]. Algebraic techniques can control the concentration of the coordinate lines, but are unable to produce meshes with the other two characteristics. Even so, no differential mapping method has been developed that accomplishes these characteristics for as large deformations as those anticipated in the present problem. A two-dimensional coordinate system becomes smoother as the magnitude of the gradients of the transformed coordinates, $|\nabla\xi|$ and $|\nabla\eta|$, decreases. A measure of the concentration of coordinate lines is the ratio of the area of a part of the physical domain to the corresponding area in the transformed domain, i.e. the Jacobian of the transformation, J. Despite its fundamental character, the analysis in [24] has limited applicability even for fixed domain problems because the selection of three optimal scaling factors is a difficult task. Kreis et al. [26] explored this problem and concluded that poorly chosen weights may lead to a mathematically ill-posed problem. The same

conclusion was drawn in moving boundary problems [23]. So in contrast to what initially was recommended, Christodoulou and Scriven [20] introduced the ratio, $S = \sqrt{(R_\xi^2 + Z_\xi^2)/(R_\eta^2 + Z_\eta^2)}$, of scale factors and its inverse, S^{-1}, for generating the η- and ξ-level curves, respectively, along with an arbitrary logarithmic term in order to regularize the orthogonality functional which is related to the Cauchy–Riemann equations. A variety of combinations and modifications of the above characteristics have been proposed [25]. In our extensive numerical experiments we found out that the logarithmic term did not offer any improvement in generating the mesh, but including S did so. Moreover, we found out that we need not include a term affecting the concentration of mesh lines, this could be accomplished through the related boundary conditions, see below. Apparently, Tsiveriotis and Brown [21] reached similar conclusions.

Furthermore, one must force the coordinate lines, which are parallel to the deforming interface (ξ-curves, having constant η-values) to follow it in its large deformations and even concentrate near it without the strong requirement that they remain orthogonal to the rest of the boundaries, [21]. On the other hand, the coordinate lines normal to the interface (η-curves) must meet them smoothly and orthogonally. Thus anisotropy of the mapping equations is generated, which proved to be valuable and directly corresponds to the physical anisotropy of the deformation: the arc length of the free surface often exceeds its original size by many times, while deforming in the direction of its original normal vector. Additional systematic tests have demonstrated that we should not include either a different scaling in the Laplacian or add a forcing term in its right hand side in the equation generating the ξ-coordinates, see also [21]. Therefore, the elliptic equation set that performed the best in our highly deforming case is

$$\nabla \cdot \left(\varepsilon_1 \sqrt{\frac{R_\xi^2 + Z_\xi^2}{R_\eta^2 + Z_\eta^2}} + (1 - \varepsilon_1) \right) \nabla \xi = 0 \quad \text{(equation generating the } \eta\text{-curves)}, \tag{15}$$

$$\nabla \cdot \nabla \eta = 0 \quad \text{(equation generating the } \xi\text{-curves)}, \tag{16}$$

where the subscripts indicate differentiation with respect to that variable and ε_1 is an empirically chosen parameter ranging between 0 and 1 and it usually equals 0.1. The gradient operator is still defined in the physical domain as in Section 2.

In the hypothetical presence of an orthogonality term in the (ξ-curves), mesh spacing becomes overly restrictive and the imposition of the boundary conditions is not possible for all the boundaries without destroying mesh smoothness. Indeed, numerical experiments have shown that when an orthogonal term is present in Eq. (16), two distinct regions of mesh distribution in the physical domain develop, which are separated by a transition zone of highly distorted elements. This is illustrated in Fig. 2 with a sequence of meshes generated according to the Christodoulou and Scriven [20] orthogonal transformation and initially using as boundary conditions equidistribution of nodes on all the boundaries with an optimally defined arc length on the free surface. More on the boundary conditions that should accompany Eqs. (15) and (16) will be presented shortly. To keep the plots clearer, we depict rectangular and not triangular elements, but this does affect our conclusions in any way. The flow parameters are $Re_p = 0$, $P_{\text{ext}} = 8333$, and $\varepsilon^{-1} = 12$, and the snapshots have been taken at $\tau = 0.555$, 1.3536, and 1.7493. Due to the additional orthogonality constraint, which tries to enforce that the angle between the ξ- and η-coordinates is nearly 90°, even the boundary constraint on the axis of symmetry had to be relaxed to simple orthogonality there. Otherwise, the mesh generation scheme would be overly restrictive, leading to failure the Newton–Raphson iterations either from the very beginning or after a few time-steps. In other words, we did not impose any condition on the distance between nodes along the axis of symmetry (i.e. equidistribution), allowing it to increase or decrease. Alternatively, one could act similarly on the free-surface or the tube wall, but such a change would have had an extremely destructive impact on the mesh. The negative effect of the orthogonal term becomes visible from the beginning of the simulation, Fig. 2(a), when the size of the bubble is small, by the nonuniformity of the mesh around the bubble. At

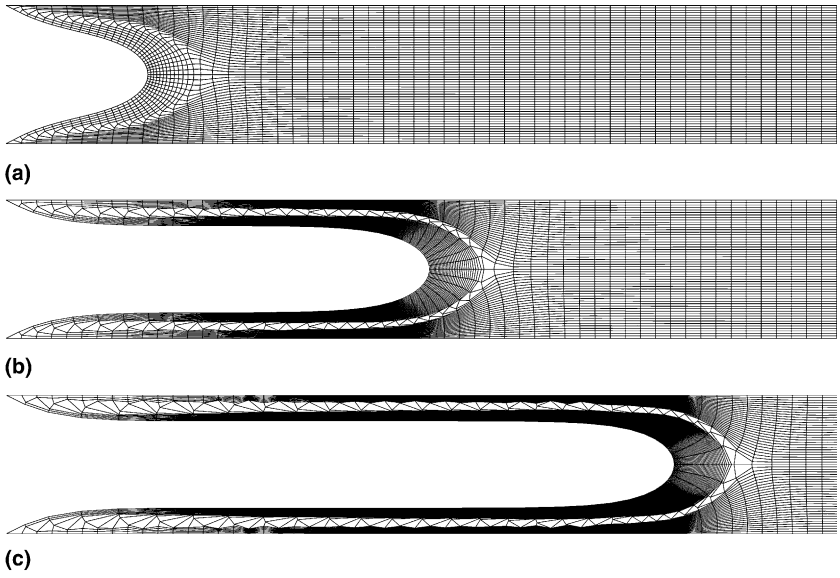

Fig. 2. Mesh deformation at $\tau = 0.555$, 1.3536, and 1.7493, for flow parameters $(Re_p, P_{\text{ext}}, \varepsilon^{-1}) = (0, 8333, 12)$. The mesh has been constructed using the geometric transformation of [20] with 40×80 radial and axial elements, respectively, and $\varepsilon_s = \varepsilon_1 = \varepsilon_2 = 0.1$. On the axis of symmetry, the equidistribution condition was dropped; its application would lead to an over specified system of equations, because of the orthogonality constraint in both equations in the bulk.

larger deformations two distinct regions are apparent, one surrounding uniformly the free surface (I) and another one with element sides nearly perpendicular to the wall (II). The zone between them is occupied by highly distorted rectangular elements with their interior angles varying from nearly 0° to 180°, making the Jacobian of the domain transformation nearly singular. The line separating the distorted from the regular elements-close to the tube wall is quite corrugated. Additionally, the aspect ratio of the elements in region I is large, while that of region II is small. All these make apparent why the orthogonality requirement in the ξ-curves is inappropriate for the deep-penetration problems studied herein.

It may seem awkward that, although the computational domain is simpler and is in fact used to readily construct a simple mesh in it, the equations generating the coordinate lines in physical space are written with the pair of (ξ, η) as unknowns! The reason for that is that the corresponding mapping $(R, Z) \rightarrow (\xi, \eta)$ is preferred because it is one-to-one provided that the curvature is non-positive and the boundary convex in the computational domain, something that can be achieved by its construction [27], whereas this is not always guaranteed if the dependent and independent variables are exchanged in Eqs. (15) and (16), see [19,25]. Moreover, Mastin and Thompson [28] have shown that the mapping generated by the Winslow system [29] which does not include any orthogonal term have a strictly positive Jacobian for a uniform mesh. When the physical domain is not convex, adding internal restrictions on the nodes solves this problem and this is something we had to employ in the constricted tube calculations, see Section 4.2. Then Eqs. (15) and (16) can be inverted to obtain $R(\xi, \eta)$ and $Z(\xi, \eta)$, i.e. the coordinate lines in the physical domain. Moreover, the governing equations, Eqs. (5)–(13), and the mesh generating equations, Eqs. (15) and (16), are written in the physical domain with independent variables (R, Z), so, in general, we need to transform them in the computational one. This procedure, although straightforward, adds to the complexity of the formulation by introducing new spatial and temporal derivatives in the governing equations according to the chain rule of differentiation

$$\frac{\partial}{\partial R}=\frac{\partial \xi}{\partial R}\frac{\partial}{\partial \xi}+\frac{\partial \eta}{\partial R}\frac{\partial}{\partial \eta},$$

$$\frac{\partial}{\partial Z}=\frac{\partial \xi}{\partial Z}\frac{\partial}{\partial \xi}+\frac{\partial \eta}{\partial Z}\frac{\partial}{\partial \eta}, \tag{17}$$

$$\frac{\partial}{\partial \tau}=\frac{\partial}{\partial \hat{\tau}}+\frac{\partial \xi}{\partial \tau}\frac{\partial}{\partial \xi}+\frac{\partial \eta}{\partial \tau}\frac{\partial}{\partial \eta}.$$

In these expressions, clearly time is independent of either the original or the mapped coordinate system, $\tau=\hat{\tau}$, so that $\partial\tau/\partial R=\partial\tau/\partial\xi=0$, etc. and the coefficient of the time derivative $\partial/\partial\hat{\tau}$ is unity. Moreover, the partial differentiations, e.g. $\partial\xi/\partial\tau$, are implied under constant location in the physical space making these derivatives nonzero. Obviously, the partial time derivatives of the coordinates in physical space are nonzero as well. In these expressions the derivatives of the mapped domain coordinates with respect to those in the physical domain must be expressed in an inverse manner. This is readily accomplished and, for example, we obtain

$$\xi_R=\frac{Z_\eta}{|J|},\quad \xi_Z=\frac{-R_\eta}{|J|},\quad \xi_\tau=\frac{R_\eta Z_\tau - Z_\eta R_\tau}{|J|},\qquad \text{where } |J|=\left|R_\xi Z_\eta - R_\eta Z_\xi\right|. \tag{18}$$

3.2. Boundary conditions

The quality of the mesh to be constructed depends not only on the bulk equations, but also on the boundary conditions that go along with them. Appropriate boundary conditions for Eqs. (15) and (16) are: (a) on fixed parts of the boundary (tube walls, axis of symmetry and the exit of the tube) the equation that defines the boundary curve replaces the mesh generation equation associated with the coordinate that is constrained on that boundary, and (b) the angle formed between the boundary and the mesh lines intersecting it. These correspond to the streamline-velocity potential couple in flow problems or to the temperature-heat flux couple in heat transfer problems. However, as we will demonstrate shortly, in order to generate an acceptable mesh one needs to control the node distribution on the boundary. This is achieved by imposing for the type (b) boundary conditions, orthogonality between the mesh lines and the boundary lines and adding to them the constraints described below, upon multiplication by a large (penalty) parameter, because these constraints are not natural for this problem. Thus, only the latter are effectively added to the bulk equations at elements on the domain boundary. Introducing them independently would lead to divergence of the iterative scheme.

These boundary constraints should be imposed on the dependent variables of Eqs. (15) and (16), i.e. on ξ or η. For example, on the axis of symmetry, $\xi=0, R=0$, and the equidistribution of nodes would require $\eta_{ZZ}=0$. However, the mesh in the computational domain is constructed algebraically (see end of this section) and we need instead to relate it to that in the physical domain. So we transform these requirements to equivalent ones in the physical domain. Using the chain rule transformations similar to those given by Eq. (17) one may show that this is equivalent to

$$Z\eta\eta=0, \tag{19}$$

where this equation requires that along the boundary increments of the arc length always vary in the physical domain as they do in the computational one. In the same boundary the second boundary condition sets the location of the boundary, $R=0$. At the free surface, the boundary condition for the η-equation, on which it is mapped ($\eta=0$), is the kinematic condition, Eq. (10). The requirement of node equidistribution

for the ξ-equation should be $\xi_{ss} = 0$, but again, because on this boundary, ξ is only a function of s and vice versa, then $\mathrm{d}/\mathrm{d}s = (\mathrm{d}\xi/\mathrm{d}s)(\mathrm{d}/\mathrm{d}\xi)$ and it is straight forward to show that this is equivalent to $s_{\xi\xi}/s_{\xi} = 0$ or

$$\frac{\mathrm{d}^2 s(\xi)}{\mathrm{d}\xi^2} = 0, \qquad \text{where } s(\xi) = \int_0^{\xi} \sqrt{w_1 R_{\xi}^2 + w_2 Z_{\xi}^2}\,\mathrm{d}\xi \quad \text{and} \quad w_1 + w_2 = 2. \tag{20}$$

$s(\xi)$ is a weighted arc length along the free surface and w_1, w_2 are weight parameters, which have to be adjusted by trial and error to optimize performance. Previous researchers [20,21] used an equidistribution condition on the free surface ($w_1 = w_2 = 1$), which is useful and accurate only when the deformation of the free surface is not very large. In our case Z_{ξ} becomes much larger than R_{ξ}, and their scheme produces unsatisfactory results. So we must use weights such that $w_1 \gg w_2$, in order to counterbalance the previously mentioned large disparity in the derivatives and prevent strong repulsion of the nodes from the tip of the bubble or from the neighborhood of the contact line, where the curvature of the surface is large. On the boundaries except for the moving surface, Tsiveriotis and Brown [21] used the condition that the respective coordinates intersect them orthogonally. This being a weak condition could smooth out irregularities in the node distribution. However, we found out that this is not a good choice for the present problem. Indeed, Fig. 3(a) and (b) show that applying this condition on all boundaries except for the free surface, where node equidistribution is applied, results in strong repulsion of nodes from the contact line, an area that needs high resolution. At later times, not shown here, there is stronger repulsion of nodes from the bubble tip as well making even the bubble shape nonphysical. As a final unsuccessful attempt we show in Fig. 3(c) the mesh generated following [20] with orthogonality boundary conditions applied on all boundaries. We observe that in addition to the strong repulsion of nodes from the contact line, the highly distorted elements near the free surface cause node-by-node oscillations on it.

For constructing the mesh in the computational domain a bottom-up algebraic technique is adopted. If the computational domain is still somewhat complicated, it is split in even simpler subdomains. Mesh tessellation takes place in these subdomains introducing equidistant nodes in both directions in the computations domain irrespective of the particular geometry that we examine herein. Then, the subdomains are patched together and double nodes are removed as described in [30].

3.3. Mixed finite element method

The coordinate lines in the computational domain generate rectangular elements, which in turn are halved to eventually create triangular elements. These, when mapped to the physical domain, were found to better conform to its large deformations. The mixed Galerkin finite element method was used following the work by Poslinski et al. [17,18] for viscous free-surface problems. With the term 'mixed' we imply using higher order polynomials to interpolate the velocity field than the pressure in order to comply with the Babuska-Brezzi condition [31]. In particular, the velocity field of the viscous fluid is represented by quadratic Lagrangian polynomials, ϕ^i, the pressure and the position vectors of the mesh points by linear Lagrangian polynomials, ψ^i and χ^i, respectively (6/3 formulation [30])

$$\begin{aligned} \underline{v} &= \sum_{i=1}^{N} \phi_i \underline{v}_i^h, \\ P &= \sum_{i=1}^{M} \psi_i P_i^h, \\ \underline{G} \equiv (R, Z) &= \sum_{i=1}^{K} \chi_i (R_i^h, Z_i^h). \end{aligned} \tag{21}$$

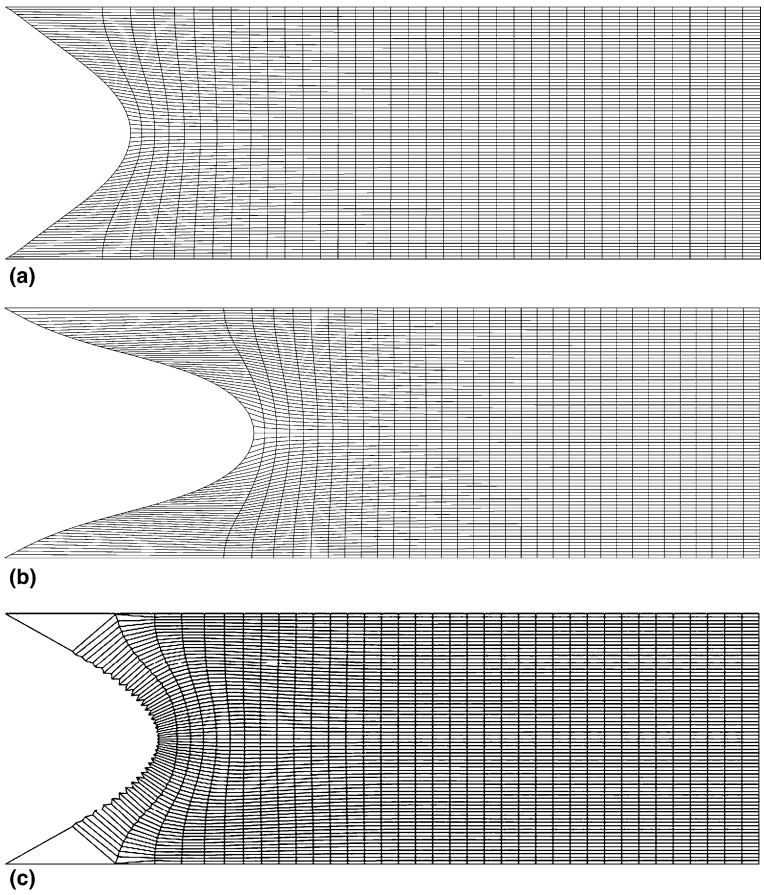

Fig. 3. Meshes generated with $(Re_p, P_{\text{ext}}, \varepsilon^{-1}) = (0, 833, 12)$ with the following methods: (a, b) using 28×60 radial and axial nodes and Eqs. (15) and (16) and orthogonality BCs except at the free interface where equidistribution is used with $w_1 = w_2 = 1$ at $\tau = 0.288$ ($Z_{\text{tip}} = 1$) and 0.772 ($Z_{\text{tip}} = 3$), respectively, and (c) using 40×80 radial and axial nodes and following the method in [20] and orthogonality BCs on all boundaries at $\tau = 0.347$ ($Z_{\text{tip}} = 1.21$). For clarity we show only half of the length of the tube and rectangular elements.

where $\underline{v}_i^h$, P_i^h, and (R_i^h, Z_i^h) are the dependent on the mesh nodal values of velocity, pressure, and position vector, and N, M, and K are the number of equations, for the above referred quantities. The mixed formulation combined with subparametric domain representation forms a stable and economical scheme that satisfies automatically the no-penetration or reflective conditions on all boundaries, except on the free surface, and conservation of mass on each element.

Galerkin's principle is invoked in order to convert the governing partial differential equations to ordinary ones; the momentum and continuity equations are weighted with the quadratic, ϕ^i, and linear, ψ^i, basis functions, respectively, and then integrated over the control volume, that is the volume of the viscous fluid.

$$R^M = \left(\phi^i, Re_p \frac{D\underline{v}}{D\tau}\right) - (\phi^i, \nabla \cdot \underline{\underline{\sigma}}) = 0, \tag{22}$$

$$R^C = (\psi^i, \nabla \cdot \underline{v}) = 0. \tag{23}$$

Eqs. (22) and (23) are the weak forms of the momentum and continuity equations, respectively. By $(\cdot,\cdot)$ is denoted the inner product defined in the whole physical domain

$$(\kappa, \lambda) = \int_V \kappa(\xi,\eta)\lambda(\xi,\eta)\mathrm{d}V, \tag{24}$$

where $\mathrm{d}V$ is given by $\mathrm{d}V = RJ\,\mathrm{d}\xi\,\mathrm{d}\eta$, and J is defined by Eq. (18). A similar inner product arises upon application of the divergence theorem on Eq. (22). Its definition involves a surface integral over part of the boundary

$$\langle \kappa, \lambda \rangle = \int_A \kappa(\xi,\eta)\lambda(\xi,\eta)\mathrm{d}A. \tag{25}$$

The locations of the mesh points continuously change in time, therefore the time derivatives in the momentum equation, Eq. (5), must be transformed according to the following relationship, which is equivalent with the last of Eq. (17):

$$\left.\frac{\partial \underline{v}}{\partial \tau}\right|_{R,Z} = \left.\dot{\underline{v}}\right|_{\xi,\eta} - \left.\dot{\underline{G}}\right|_{\xi,\eta} \cdot \nabla \underline{v}, \tag{26}$$

where the dots over the symbols indicate differentiation with respect to time. The last term in Eq. (26) is an additional convective term due to the motion of the nodes. The momentum equation involves the divergence of the stress tensor ($\nabla \cdot \underline{\underline{\sigma}}$) and requires application of the divergence theorem within each element in order to reduce the order of the velocity derivatives from two to one. The resulting line integrals cancel each other within the control volume, whereas on fluid boundaries they are either omitted in order to impose essential conditions, i.e. no slip, no penetration at the solid wall, symmetry condition at the centerline, or computed by the free surface stress balance. The mean curvature has to be specially treated as it involves products of the free surface position with its second derivative. Using the methodology proposed by Ruschak [32] and applied by Poslinski and Tsamopoulos [17], the mean curvature on the free surface is split into two parts. The first term is the derivative of the tangent vector, $\underline{t}$, of the free surface with respect to its arc length (s), while the second part is composed of the normal vector multiplied by the inverse of the second principal radius, $R_2 = R\sqrt{R_\xi^2 + Z_\xi^2}/Z_\xi$,

$$2H\underline{n} = \frac{\mathrm{d}\underline{t}}{\mathrm{d}s} + \frac{\underline{n}}{R_2}, \tag{27}$$

where $\underline{t}$ is given as $\underline{t} = (R_\xi \underline{e}_r + Z_\xi \underline{e}_Z)/\sqrt{R_\xi^2 + Z_\xi^2}$ and $\underline{n}$ is equal to $\underline{n} = (R_\xi \underline{e}_Z - Z_\xi \underline{e}_r)/\sqrt{R_\xi^2 + Z_\xi^2}$.

Introduction of Eqs. (26) and (27) into the weak form of momentum equation eventually results in

$$R^M = (\phi^i, Re_p[\dot{\underline{v}} + (\underline{v} - \dot{\underline{G}}) \cdot \nabla \underline{v}]) + (\nabla \phi^i, \underline{\underline{\sigma}}) - \left\langle \frac{\mathrm{d}\phi^i}{\mathrm{d}s}, P_{\mathrm{ext}}^{-1}\underline{t} \right\rangle - \left\langle \phi^i, \frac{\underline{n}}{R_2} \right\rangle = 0. \tag{28}$$

Similarly, the weak forms of the mapping equations are derived by multiplying them with the χ^i basis functions

$$R^{M_R} = \left(\chi^i, \nabla \cdot \left(\varepsilon_1 \sqrt{\frac{R_\xi^2 + Z_\xi^2}{R_\eta^2 + Z_\eta^2}} + (1 - \varepsilon_1)\right)\nabla \xi\right) = 0, \tag{29}$$

$$R^{M_Z} = (\chi^i, \nabla \cdot \nabla \eta) = 0. \tag{30}$$

Applying the divergence theorem, we get

$$R^{M_R} = \left(\nabla\chi^i, \left(\varepsilon_1 \sqrt{\frac{R_\xi^2 + Z_\xi^2}{R_\eta^2 + Z_\eta^2}} + (1 - \varepsilon_1) \right) \nabla\xi \right) = 0, \tag{31}$$

$$R^{M_Z} = (\nabla\chi^i, \nabla\eta) = 0, \tag{32}$$

where the integrated out terms are not included because we weakly impose orthogonality between the mesh lines and the domain boundaries on all boundaries and we add conditions on the node distribution through a penalty method [20]. For example, the node equidistribution condition on the free surface the boundary is imposed by

$$R^{M_R}\big|_{\eta=0} = \left(\nabla\chi^i, \left(\varepsilon_1 \sqrt{\frac{R_\xi^2 + Z_\xi^2}{R_\eta^2 + Z_\eta^2}} + (1 - \varepsilon_1) \right) \nabla\xi \right) + L \left\langle \frac{\mathrm{d}\chi^i}{\mathrm{d}\xi}, \frac{\mathrm{d}s}{\mathrm{d}\xi} \right\rangle = 0, \tag{33}$$

where the penalty parameter is $L = \mathrm{O}(10^3\text{–}10^5)$. Even Eqs. (19) and (20) are treated as essential conditions after multiplication by ς^i, the one-dimensional linear basis functions, and integration by parts. Finally, the equivalent algebraic form of the kinematic differential equation is

$$R^K = \left\langle \varsigma^i, \frac{D\underline{F}}{D\tau} - \underline{v} \right\rangle = 0. \tag{34}$$

Alternatively it could be stated that the free surface corresponds a surface with constant isoparametric coordinate $\eta = 0$ [19], and write

$$R^K = \left\langle \varsigma^i, \frac{D\eta}{D\tau} \right\rangle = 0, \tag{35}$$

which is equivalent to Eq. (34). Starting from either one of Eqs. (34) or (35) and applying the chain rule of differentiation, we readily deduce that

$$R^K = \langle \varsigma^i, R_\tau Z_\xi - R_\xi Z_\tau + R_\xi v_z - Z_\xi v_r \rangle = 0. \tag{36}$$

According to some investigators (e.g. [33]) a SUPG weighting of the kinematic condition must be used because of the wave nature of Eq. (36). This is indeed the case for high order polynomial approximations of the free surface, but in our case it is unnecessary because we employ approximation functions of low degree. All the above integrals are evaluated numerically by Gaussian quadrature. For triangular elements a seven-point formula was used to evaluate two-dimensional integrals and a three-point formula for one-dimensional integrals.

In order to integrate accurately the governing equations in time, the implicit Euler method, which is an A-stable approximation, with time stepping adaptation, is used. More specifically, if by $\partial\underline{\kappa}/\partial\tau = f(\underline{\kappa})$ we denote the set of equations to be integrated, its approximate form, using backward finite differences is

$$\frac{\underline{\kappa}_{n+1} - \underline{\kappa}_n}{\tau_{n+1} - \tau_n} = \underline{f}(\underline{\kappa}_{n+1}), \tag{37}$$

where the index n indicates the previous time instant. The difference $\tau_{n+1} - \tau_n$ defines the current time step $\Delta\tau_{n+1}$. The strategy for changing the time step is based on the estimation of the local truncation error, which is the difference between the accurate approximation $\underline{\kappa}_n$, and an explicitly predicted one $\underline{\kappa}_n^p$:

$$\underline{\kappa}_n^p = \underline{\kappa}_{n-1} + \Delta\tau_{n-1}\dot{\underline{\kappa}}_{n-1}, \tag{38}$$

$$\Delta\tau_{n+1} = \Delta\tau_n \left(\frac{\varepsilon}{\|\underline{d}_n\|} \right)^{1/2} \equiv \Delta\tau_e, \tag{39}$$

where ε is a user defined tolerance, $\|\cdot\|$ stands for the Euclidean norm and $\underline{d}_n = \underline{\kappa}_n - \underline{\kappa}_n^p$, is the difference between the predicted and the accurate solution at τ_n [34].

The resulting set of non-linear equations is solved by the modified Newton–Raphson iteration scheme from an initial estimate corresponding to the solution of the prediction step, Eq. (38). The modified Newton method proceeds by not updating after each iteration the Jacobian matrix and its LU decomposition, but only after a criterion of decreased convergence rate is exceeded. The entries of the Jacobian matrix are built up as sums of contributions from the element-level Jacobian, which is calculated by analytical differentiation. The agreement of successive iterates to within a prescribed numerical tolerance, typically 10^{-9}, was the criterion for convergence. The large system of algebraic equations arising at each step of the Newton iterations was solved by Gaussian elimination using a banded matrix solver.

Although the above numerical scheme guarantees the convergence to a solution at each time step, and can be easily embedded in a typical finite element code, it is extremely memory demanding. Therefore, a different scheme was adopted in order to decouple the mesh generation problem from the flow problem, [35]. In particular, the total set of equations

$$\underline{R}_{\mathrm{es}}(\underline{v}, p, \underline{G}) = \underline{0} \quad (2N + M + 2K \text{ unknowns}) \tag{40}$$

is split in two sets and each set is solved by Newton–Raphson iterations until convergence. The first set consists of the mass and momentum balances (Eqs. (23) and (28)) and their boundary conditions

$$\underline{R}^v_{\mathrm{es}}(\underline{v}^j, p^j, \underline{G}^{j-1}) = \underline{0} \quad (2N + M \text{ unknowns}), \tag{41}$$

while the second set consists of the mesh generating equations (Eqs. (31) and (32)) and their boundary conditions

$$\underline{R}^m_{\mathrm{es}}(\underline{v}^j, p^j, \underline{G}^j) = \underline{0} \quad (2K \text{ unknowns}). \tag{42}$$

Thus the complete scheme is a Gauss–Seidel (or Picard) iteration type scheme in which for a given physical domain, $\underline{G} = \underline{G}^{j-1}$, the flow problem is solved first (stage I), and once velocities and pressure have been calculated, the mesh points are moved (stage II). The basic advantage of the method is that it generates significantly smaller Jacobian matrices, which can be allocated and deallocated whenever needed. The convergence of the whole scheme is ensured by the automatic time adaptation, because the predicted solution does not differ too much from the exact one and the Newton/Kantorovich sufficient condition is satisfied, [36]. It must be mentioned that in order to improve the effectiveness of the scheme and disallow unreasonable increases of the time step, an upper bound is placed on the latter by the number of Picard iterations. So the new time step, $\Delta\tau_{n+1}$, is determined by

$$\Delta\tau_{n+1} = \min(\Delta\tau_p, \Delta\tau_e), \tag{43}$$

where $\Delta\tau_e$ stands for the predicted time step (Eq. (39)) using forward Euler technique [34], whereas $\Delta\tau_p$ is given by

$$\Delta\tau_p = \Delta\tau_n \left(\frac{\text{desired number of Picard cycles}}{\text{actual number of Picard cycles}} \right)^{1/4},$$

and $\Delta\tau_n$ is the time step in the previous instant. Usually, the desired number of Picard iterations is equal to the actual ones at the first integration step increased by 3–4.

In addition, the proposed scheme assists in readily imposing boundary conditions that depend on the location of a particular point of the physical domain (e.g. its distance from the three-phase contact point, when a Navier slip condition is employed) without increasing the bandwidth of each submatrix, reduces the calculation of matrix entries and expedites checking their correctness. Moreover, for the flow problem (and not for the mesh generation one) it is not necessary to apply the chain rule once the coordinate points of the physical domain are given. Finally, it must be mentioned that both full Newton–Raphson and Block Gauss–Seidel/Newton–Raphson methods have been numerically implemented and compared. Their execution times do not differ too much, with the NR method being typically 10% faster, since the GS/NR requires about five Picard cycles per time step. So, the GS/NR should be preferred for the reasons mentioned above.

The program was written in Fortran 90 and run on an Alpha Dec DS20E workstation at the laboratory of Computational Fluid Dynamics. Depending on the parameters and the type of the tube geometry, accuracy of the results was tested by mesh refinement and required different size of mesh, but it typically took about 1 day to complete a run.

4. Results and discussion

The transient flow generated by the penetrating bubble and the bubble shape are functions of the modified Reynolds number, the dimensionless applied pressure, and the geometric ratios characterizing the tube. The most general form of a tube section that we consider in this work is given in Fig. 1, and its various particular simplifications will be given in each subsection that follows.

4.1. Straight tube

As we mentioned in Section 2, the complex tube geometry defined by Eq. (1) reduces to a straight one by setting $b_1 = b_2 = b_3 = b_4$. The sequence of plots shown in Fig. 4 illustrates the deformation of the discretized into triangular elements physical domain, which is generated by a uniform mesh in the computational domain at $\tau = 0.65$, 1.20, and 1.80. For clarity the elements are shown as rectangular, i.e. before dividing each one of them into two triangles. We observe that, even in the cases with the largest penetration of the air bubble, the grid retains acceptable uniformity, in contrast to the one shown in Fig. 2, which was obtained under similar values of the physical parameters, but not following our new scheme. The mesh here deforms smoothly and nicely following the deformation of the free surface. Regions of particular interest, which need special attention, such as the contact line region and the front of the bubble, are handled very effectively. In this case we have $Re_p = 0$, $P_{\text{ext}} = 833$, and $\varepsilon^{-1} = 12$ and we have set the boundary control parameters, w_1 and w_2, equal to 20/11 and 2/11, respectively. Doing so, we overcame the difficulty of the large Z_ξ derivative compared with R_ξ, which is known to cause node repulsion along the ξ-direction. To simulate accurately the above-described phenomenon, the computational domain was subdivided into a uniform mesh by 37 nodes in the ξ-direction and 81 nodes in the η-direction, resulting in 5760 triangular elements and 32,497 unknowns including the coordinate locations of the mesh points. The initial time increment was set at $\Delta\tau = 0.5 \times 10^{-5}$.

From the physical point of view, we observe that the initially flat air/fluid interface deforms everywhere, but mostly around the axis of the tube initially forming a wedge-like profile. Already at $\tau \sim 0.544$ the bubble shape has turned into almost parabolic. Subsequently, most of the bubble front assumes a permanent, finger-like shape with a parabolic tip and straight sides leaving behind a layer of fluid of constant thickness, which is attached to the tube wall. Near the tube entrance the film has a small, but monotonically increasing thickness at all times. When the bubble front reaches the length of one and a half diameters, it attains a constant thickness. This thickness is equal to $h_\infty = 0.6325$ and the liquid volume fraction

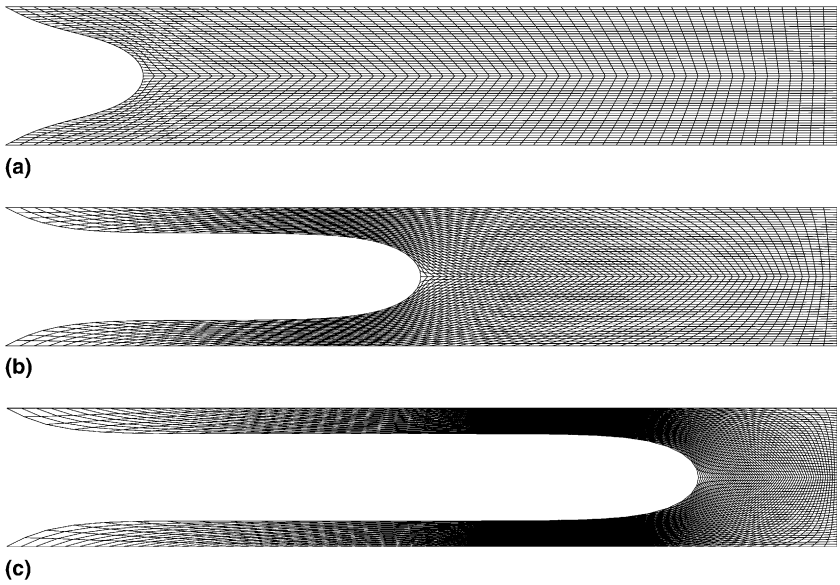

Fig. 4. The deformation of the mesh points at $\tau = 0.544$, 1.334, and 1.774. The discretization is finest at the fingertip, since both the local capillary number and the curvature are large there. $(Re_p, P_{\text{ext}}, \varepsilon^{-1}) = (0, 833, 12)$.

remaining attached to the tube wall, which is defined as $m = 1 - h_\infty^2$, is 0.60. These values have been observed experimentally by Taylor and Cox [2–4], when they studied the displacement of very viscous fluids at their higher Capillary numbers (i.e. $Ca \sim 2$–10, asymptotic region of their m vs Ca plot). Their experiments were carried out in very long tubes and they neglected the initial transients and the tube entrance effects.

4.2. Constricted tube

The methodology we presented needs only small modifications in order to extend the simulations to cover cases where the bubble enters a secondary tube of smaller diameter than that in the primary one. Now it is anticipated that the bubble radius will decrease as it tries to squeeze through the contraction and the bubble may attain a concave shape. This is not desirable, if the mapping we used is to be preserved throughout the simulations [27]. So we need to introduce an internal constraint on the nodes right at the entrance of the secondary tube. In particular, instead of the ξ-component of the mesh generation set of equations, which is dropped, the nodes are forced to obey a generalized distribution similar to that given by Eq. (20) at the entrance of the secondary tube in the physical domain. Now the nodes in both tubes translate following the deformation of the free surface. Thus, the nodes originally located at the vertical boundary between the two tubes translate and rotate with respect to the contraction corner, so that an optimal node distribution is achieved. Some indicative results of the generated mesh before, during and after the bubble penetration in the secondary tube are shown in Fig. 5. A short primary tube, $b/a = 5$, with a 4:1 contraction ratio and dimensionless length of the secondary tube $b_4/a = 4$ is at $\tau = 0$ filled with liquid. The applied pressure and the other liquid properties are such that $Re_p = 0$, $P_{\text{ext}} = 10^4$. To accurately simulate the penetration of air through the viscous fluid, a mesh of 13,160 triangular elements is used resulting in 73,757 unknowns. The sequence of finite element meshes in the displaced liquid as well as the bubble shape is shown at different time instances. At $\tau \sim 131$ the bubble nearly exceeds in length three radii of the primary tube and its front surface is nearly parabolic. At $\tau \sim 189$ the bubble tip reaches the entrance of the secondary tube and becomes very pointed as gas and liquid try to squeeze through the contraction

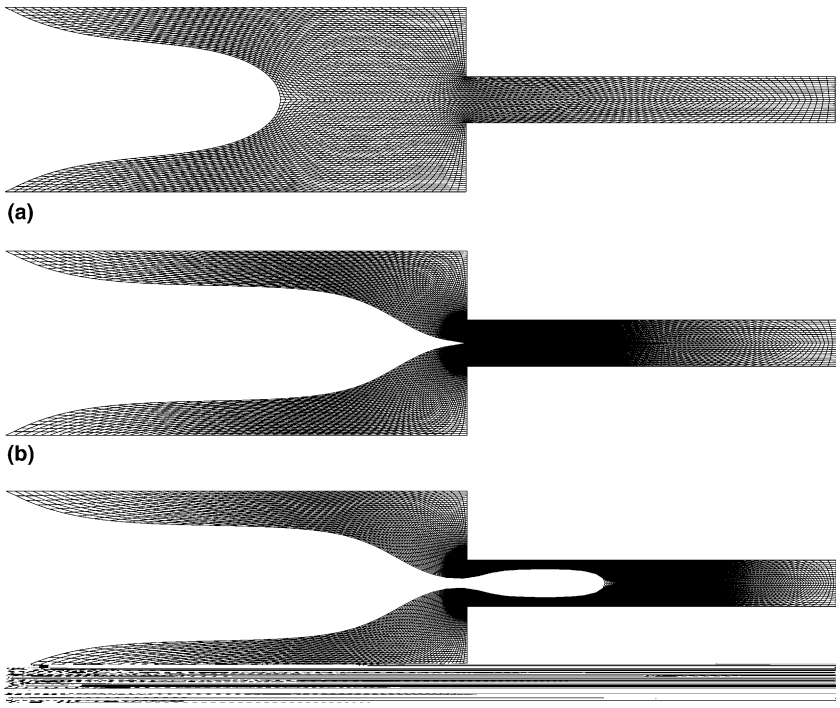

Fig. 5. Sequence of finite element meshes in the liquid that is being displaced by the gas, at $\tau = 131.32$, 189.67, and 193.14 corresponding to $Z_{\text{tip}} = 3, 5, 6.5$ with $(Re_p, P_{\text{ext}}, \varepsilon^{-1}, b_4/a, a_2/a) = (0, 10^4, 9, 4, 0.25)$. For clarity of the figure we have used only 44×53 and 11×42 radial and axial nodes in the primary and the secondary tubes, respectively. This results in 5588 triangular elements and 11,455 total nodes.

generating a locally extensional flow of the liquid. Soon after entering the narrower tube, the bubble front expands again and eventually its front becomes parabolic again. A new, narrower finger is formed that continues to travel inside the secondary tube, but with a greater speed ($\tau = 191$). In the neighborhood of the contraction corner, the bubble width remains very small, $a_b = 0.065a$. As the bubble occupies more and more space in the secondary tube, its radius seems to reach the asymptotic value of $0.156a$ or $0.624a_2$, which corresponds to a remaining liquid fraction in the secondary tube $m = 0.61$, nearly Cox's [3] value for a straight tube, although the length of the secondary tube and, thus, the fluid resistance here are not enough to give the bubble a constant velocity. Apparently the mesh has all the desired qualities: uniform distribution of nodes throughout, bound ratio of the element sides, and dynamically adjustable nodal positions as the free surface deforms greatly. Details of the mesh around the contraction corner for times corresponding in cases that the bubble has either come very close to its entrance ($\tau = 185.45$) or even penetrated in the secondary tube ($\tau = 193.94$) are shown in Fig. 6(a) and (b).The bounds of the zoomed area are $4 \leqslant Z \leqslant 6$. Using this method, we have studied the flow of Newtonian and viscoplastic fluids in straight and constricted tubes [37,38].

In order to demonstrate the superiority of the proposed quasi-elliptic grid generation methodology over algebraic ones, we applied the transfinite mapping technique of Gordon and Hall [39] with bilinear interpolant of the boundary surfaces [40] to a typical displacement simulation inside a suddenly constricted tube. According to their method, each of the primary and the secondary tubes are mapped to a unit square in the computational domain. For the primary tube, the left, right, top, and bottom sides of this square correspond to the deformed surface with coordinates $\underline{G} = (R, Z)$ given by $\underline{G}(s = 0, t)$, the common surface

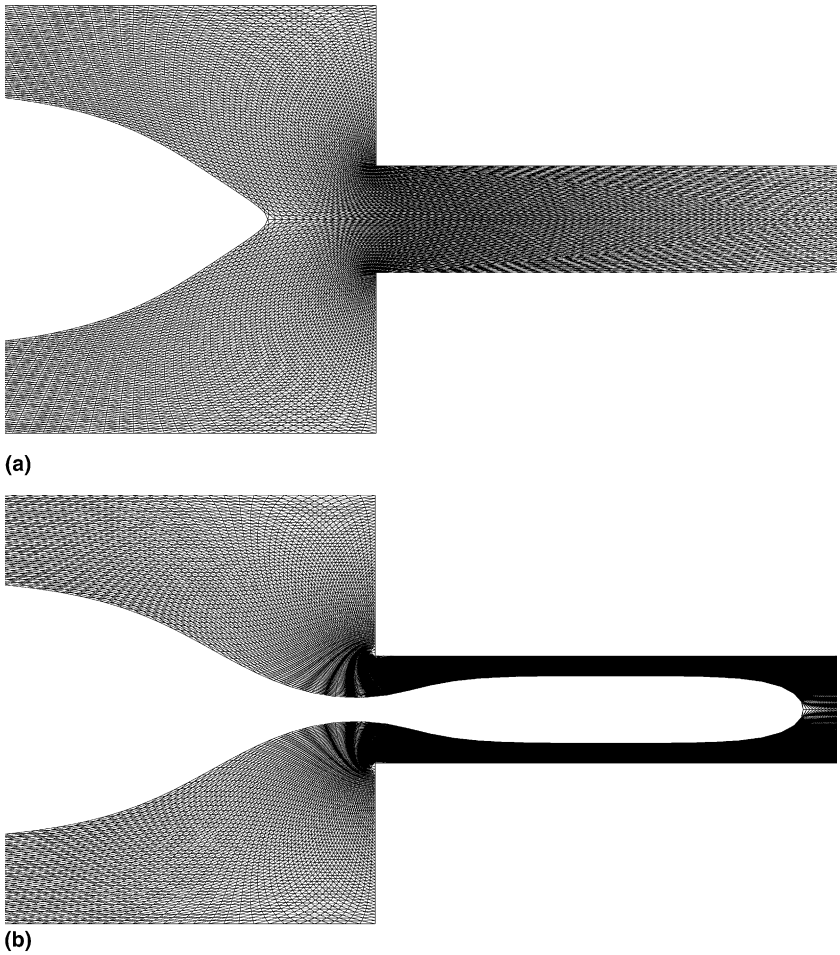

Fig. 6. Closeups of finite element meshes in the liquid that is being displaced by the gas, at $\tau = 185.45$ and 193.94. Flow parameters are identical with those of Fig. 5.

between the two tubes given by $\underline{G}(s = 1, t)$, the tube wall $\underline{G}(s, t = 1)$, and the axis of symmetry $\underline{G}(s, t = 0)$, respectively. The calculation of the coordinates of the boundary nodes is based on the simultaneous solution of a generalized node distribution condition, Eq. (20), and the kinematic condition on the free surface, or an equation defining the other three boundaries; typically via an one-dimensional finite element technique. Then we can specify the position of the interior points from the coordinates of the nodes in the four boundaries of the physical domain by applying a simple algebraic formula, without needing to use any iterative scheme (like Newton–Raphson), nor inverting any matrix. More specifically the bilinear interpolation inside the domain is

$$\begin{aligned}\underline{G}(s,t) = {} & (1-s)\underline{G}(0,t) + s\underline{G}(1,t) + (1-t)\underline{G}(s,0) + t\underline{G}(s,1) \\ & - [(1-s)(1-t)\underline{G}(0,0) + (1-s)t\underline{G}(0,1) + s(1-t)\underline{G}(1,0) + st\underline{G}(1,1)],\end{aligned} \tag{44}$$

where s, t are the normalized arclength of the vertical and horizontal boundaries of the computational domain ($[0,1] \times [0,1]$). A similar approach is followed for the secondary tube.

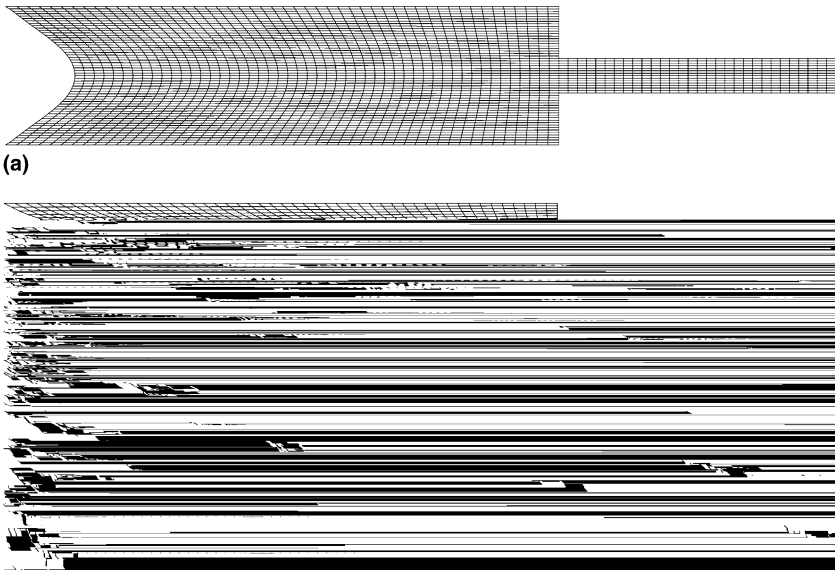

Fig. 7. Meshes generated with $(Re_p, P_{\text{ext}}, \varepsilon^{-1}, b_4/a, a_2/a) = (0, 10^4, 12, 4, 0.25)$ using the transfinite mapping technique with bilinear interpolant, at $\tau = 26.27$, 97.07, and 112.28. For clarity we show rectangular elements.

We test this methodology on a constricted tube with typical geometric parameters $b/a = 8$, $b_4/a = 4$, $a_2/a = 0.25$ (Fig. 7). Flow conditions are such that $Re_p = 0$, $P_{\text{ext}} = 10^4$. For the discretization of the computational domain, we indicatively use 50×28 and 30×7 elements in each direction inside the primary and the secondary tube, respectively. Apparently, the quality of the grid in the first stages of the simulation (Fig. 7(a): $\tau = 26.27$, $Z_{\text{tip}} = 2$) is high. The coordinate lines inside the primary tube follow closely the geometry of the four boundaries, remaining straight between the lower and the upper horizontal lines, or progressively changing from parabolic to straight between the vertical boundaries. Later on, although the mesh retains its high quality along the whole boundary, the algebraic transformation produces elements with large skewness in about the middle of the radial distance (Fig. 7(b): $\tau = 97.07$, $Z_{\text{tip}} = 4$). The bubble has almost run one half of the primary tube when the simulation stops (Fig. 7(c): $\tau = 112.28$, $Z_{\text{tip}} = 4.657$), because of the failure of the iterative solution scheme to converge, as a consequence of the large deformation of the elements in the above-referred region. Up to this instance the accuracy of the calculations is as good as that from the elliptic grid generation method. Therefore, we may remark that despite their low computational cost and their effectiveness in simply connected geometries, algebraic techniques do not have the robustness of the proposed elliptic one.

4.3. Complex tube

A very often encountered geometry in industrial applications such as GAIM is the one having both an expanding and a contracting region as given in Fig. 1, because here the evolution of the bubble is controlled more readily. Now the mapping equations from the physical domain (R, Z) to the computational one (ξ, η) must be modified as follows [25]:

$$\nabla \cdot \left(\varepsilon_1 \sqrt{\frac{R_\xi^2 + Z_\xi^2}{R_\eta^2 + Z_\eta^2}} + (1 - \varepsilon_1) \nabla \xi \right) = Q, \tag{45}$$

$$\Delta\eta = 0, \tag{46}$$

where Q is a forcing term, added to the RHS of the ξ-component of the mesh generating equations turning it into a modified Poisson equation. This additional term reduces the repulsion, which is caused by the concave nature of the complex geometry at the boundary around the corners of the primary tube and it has the form

$$Q = -\sum_{i=1}^{2} b_i \exp\{-d_i[(\xi - \xi_i)^2 + (\eta - \eta_i)^2]\}. \tag{47}$$

b_i and d_i are appropriate coefficients ($d_1 = d_2 = 3$) and (ξ_i, η_i) stand for the coordinates of the two recirculation corners. The final shape of initial physical domain is generated after a sequence of gradual deformations of the straight computational one. The procedure is completed with the gradual increase of b_i's from 0 to 200. It is impossible to apply their final values without any continuation procedure, because such an attempt makes the Newton–Raphson iterations to fail. Moreover, the distribution of the nodes along the varying tube wall follows Eq. (20), albeit with w_1 slightly smaller than w_2; their typical values are 0.8 and 1.2, respectively.

The complex tube of Fig. 8, which has been discretized in triangular elements, has an expansion ratio equal to $r_e = a_1/a = 0.25$ and a contraction ratio $r_c = a_2/a = 0.1$. Its total dimensionless length is $\varepsilon^{-1} = \bar{Z}_5/a = 3$, and the particular lengths of the introductory and the exit tubes are $\varepsilon_i^{-1} = \bar{Z}_1/a = 0.33$ and $\varepsilon_e^{-1} = b_4/a = 0.33$, respectively. To capture the large variation of the tube geometry an increased number of elements in the axial direction is needed. Characteristically, we use 120 elements in the axial direction and 46 in the radial direction, which is the crucial one, since it controls the resolution of the free surface. Unfortunately, the number of elements in the radial direction governs the band-width of the Jacobian matrix and, thus, it cannot be increased much without significantly increasing the computational time. The resulting number of triangular elements is 11,040 and the total number of unknowns is equal to 61,887. In the upper part of Fig. 8 we depict the mesh generated when attractive term is absent from Eq. (45). We then observe that the ξ-lines of the grid are mostly concentrated around the axis of symmetry, while the regions

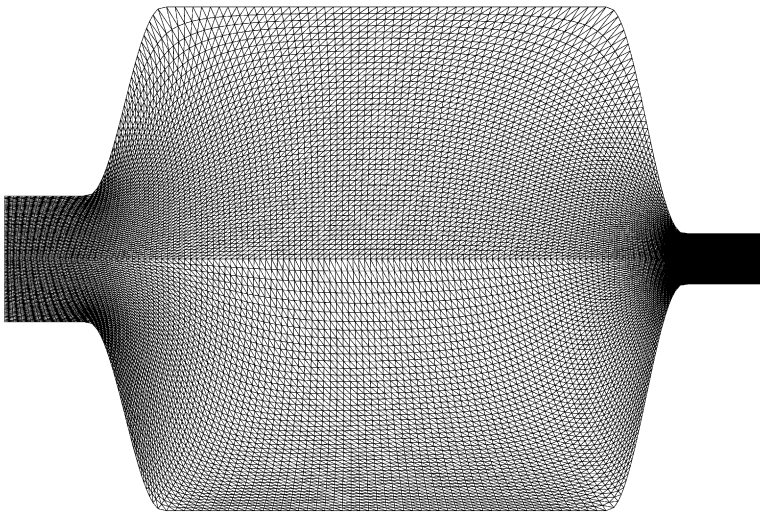

Fig. 8. Mesh generated without and with attractive terms in the RHS side of Eq. (45), on the upper and the lower part of the figure, respectively.

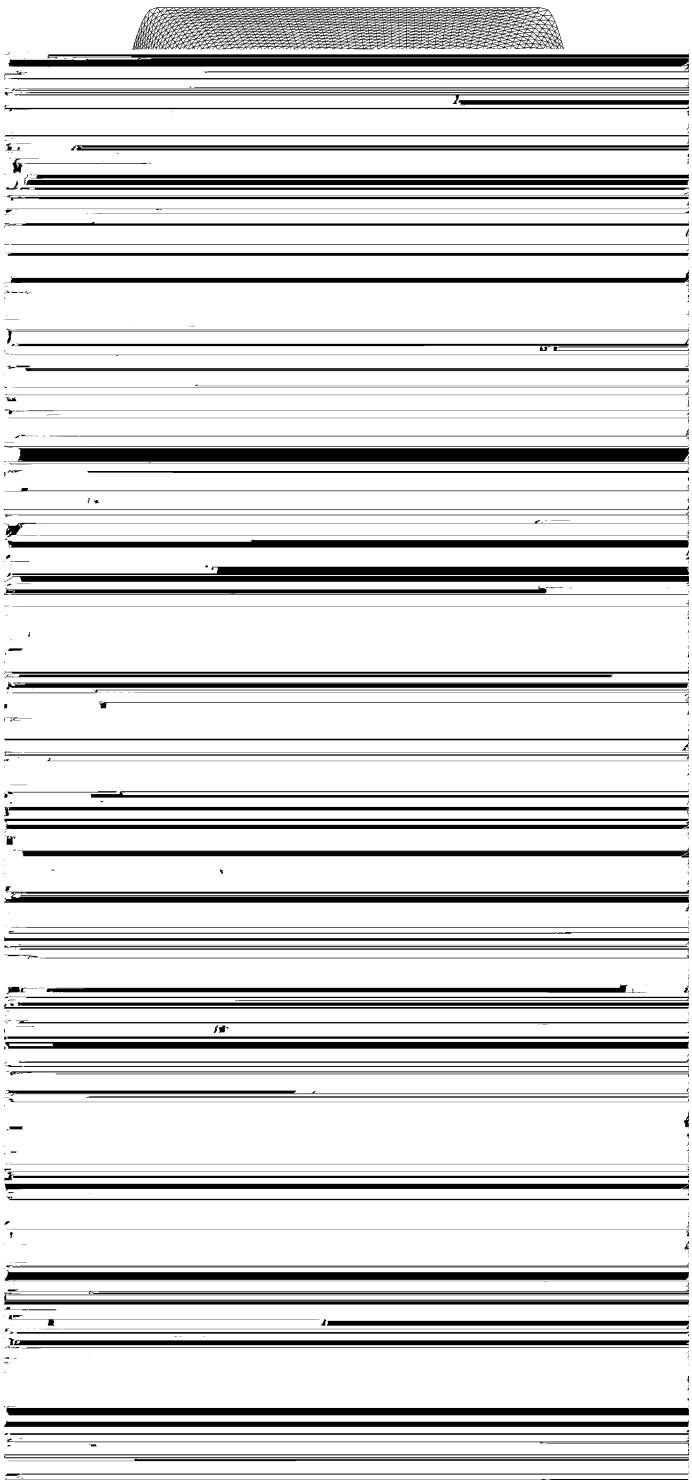

Fig. 9. The deformation of the mesh points at $\tau = 37.5$, 785.5, and 1387 in a case of a complex tube. $(Re_p, P_{\text{ext}}, \varepsilon^{-1}, a_1/a, a_2/a) = (0, 10^5, 3, 0.25, 0.1)$.

around recirculation corners have been covered by elements with large aspect ratios. However, in the lower part of Fig. 8 which has been generated by adding this attractive term at both corners, we observe a much improved node distribution throughout the tube.

Fig. 9 depict three different stages of a bubble evolution and the corresponding meshes in a complex tube for a liquid with $P_{\text{ext}} = 10^5$ and $Re_p = 0$; (a) when the bubble is traveling in the introductory tube ($\tau = 37.5$), (b) when it is expanding nearly uniformly in all directions with a constant velocity in the main tube ($\tau = 787.5$), and (c) when it is elongating moving fast towards the exit of the main tube ($\tau = 1387$). As one can observe in Fig. 9, the bubble surface is moving both horizontally and vertically. Therefore, the nodes of the free boundary must be allowed to move not only in the axial direction as before, but also along the radial direction. This is achieved by forcing the weighting factor w_2 to now have a greater value than that of the w_2 ($w_2 = 1.2 > w_2 = 0.8$). In this way, a better distribution of the nodes is achieved along the free surface and the simulations can follow the bubble while it travels longer distances ($\tau = 1387$). Despite the large deformation of the bubble, nodal points adjust their position smoothly and nicely, constructing an optimal set of finite elements. We must pay special attention on the two corners of the main tube and see the effect of the two attractive terms, Q, on the mesh. Now, the η-lines follow more closely the wall and the maximum distance between them has decreased significantly. In Fig. 9(b), we can also see the effect of the orthogonal term on the free surface: ξ-lines try to intersect the bubble vertically, so they bend themselves in a way that is dictated by the geometry. The concentration of the nodes retains its initial uniformity independently from the depth of penetration.

4.4. Undulating tube

A final case that we have examined is often used in simulations of enhanced oil recovery and it is a harmonically undulating tube with straight entrance and exit supplements. The tube surface (S_r) is function of the axial distance

$$S_r(\bar{Z}) = \begin{cases} R_{\max}, & 0 \leqslant \bar{Z} \leqslant b_1, \\ \frac{R_{\max}}{2}\left[\left(1+\frac{R_{\min}}{R_{\max}}\right)+\left(1-\frac{R_{\min}}{R_{\max}}\right)\cos\left(\pi\left(2\frac{\bar{Z}-b_1}{\lambda}\right)\right)\right], & b_1 \leqslant \bar{Z} \leqslant b_1+b, \\ R_{\max}, & b_1+b \leqslant \bar{Z} \leqslant b_1+b+_2, \end{cases} \tag{48}$$

where $R_{\min}$ and $R_{\max}$ are the minimum and the maximum radii of the undulated part of the tube, b_1, b, and b_2 are the lengths of the entrance, primary and exit tubes, and λ stands for the wavelength of the undulating

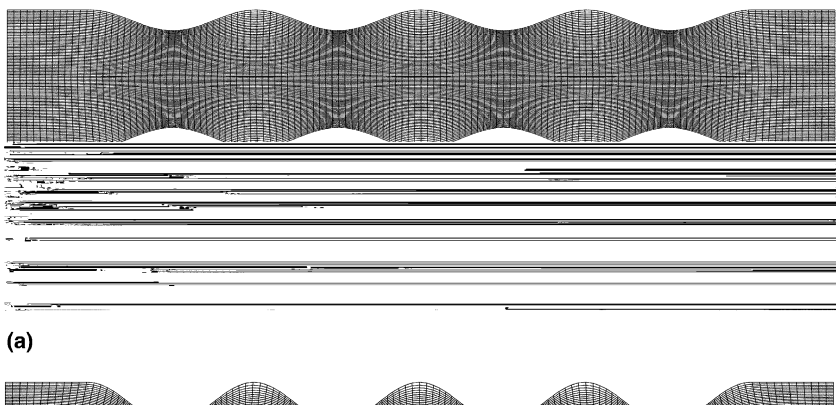

Fig. 10. Typical physical domains of undulated tubes generated by deforming straight ones: (a) $R_{\min}/R_{\max} = 0.7$ and (b) $R_{\min}/R_{\max} = 0.4$. The rest of the ratios are $b_1/R_{\max} = 1.2$, $b_2/R_{\max} = 2.4$, $n = 4$, and $\lambda = 2.1$.

part. In particular, λ is defined as $\lambda = b/n$ where n stands for the number of the geometric undulations. In order to construct the mesh in an undulated tube with a given undulation ratio $R_{\min}/R_{\max}$, we use a continuation technique beginning from a straight tube, where $R_{\min}/R_{\max} = 1$ of equal total length and progressively we decrease the radii ratio until reaching the desired value. The continuation step is usually equal to 0.05. Fig. 10 depict two indicative cases with $R_{\min}/R_{\max}$ ratios equal to 0.7 (Fig. 10(a)) and 0.4 (Fig. 10(b)), respectively. The number of elements for the tube with the smaller amplitude of undulations is 46 and 130 in the radial and the axial direction, respectively, resulting in 24,273 nodes. On the top boundary of the rectangular domain we apply the generalized distribution condition, Eq. (20), in order to achieve an optimal concentration of the nodes in the physical domain. Typical values of the domain parameters are $b_1/R_{\max} = 1.2$, $b_2/R_{\max} = 2.4$, and $\lambda = 2.1$.

An extremely computationally demanding situation is shown in Fig. 11. The undulating ratio of the tube surface is equal to 0.40 and consequently the total arc length of the developing bubble is even larger now. So accuracy requires an increased number of elements in the radial direction must be used – namely 60. Flow conditions remain creeping ($Re_p = 0$) and the effect of capillary forces is insignificant when compared to the applied gaseous pressure ($P_{\text{ext}} = 8.33 \times 10^3$). Fig. 11(a) correspond to $\tau = 4.841$ and Fig. 11(b) to

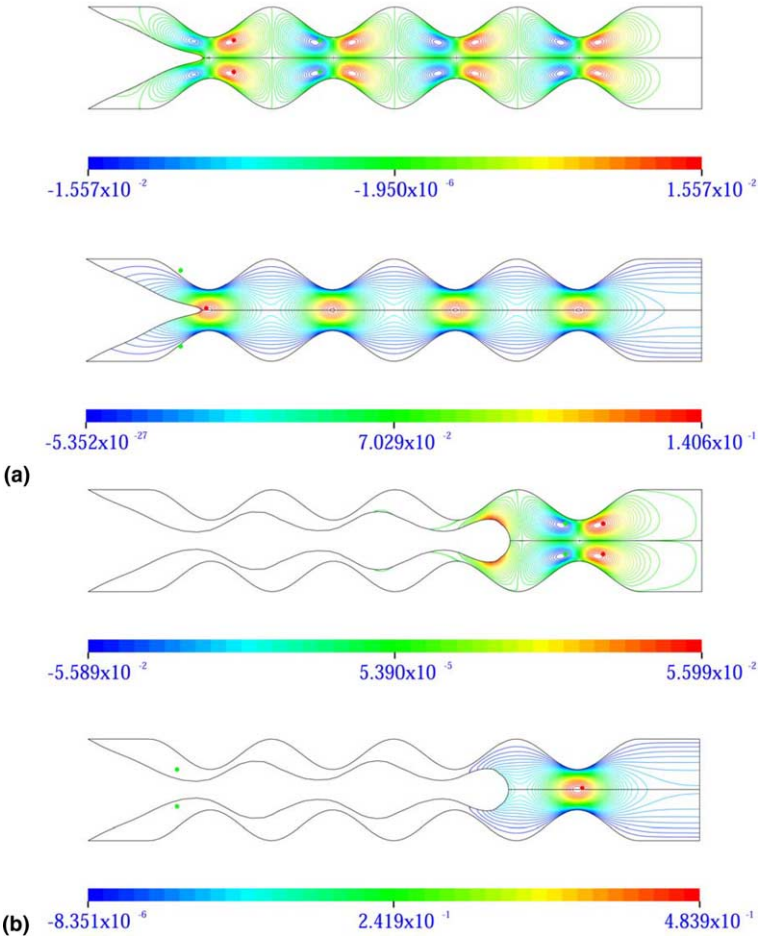

Fig. 11. Contour lines of the axial and radial velocity component, upper and lower part of each figure, in an undulated tube with $R_{\min}/R_{\max} = 0.4$ at $\tau = 4.841$ and 8.769. $(Re_p, P_{\text{ext}}, \varepsilon^{-1}, n) = (0, 8333, 12, 4)$.

$\tau = 8.769$, or equivalently to axial penetrations equal to 2.251 and 8.251, respectively. In the upper part of each figure we can observe contour plots of the radial velocity component, while in the lower part contours of the axial velocity are shown. Apart from the clarity of the results, which is a demonstration of the quality of the generated mesh, we observe the symmetry of the radial velocity field that exists inside each undulation of the tube. In particular the left side of each undulation is colored red (positive values), due to the expanding motion of the fluid particles, while the right side is colored blue (negative values), because of the fluid's tendency to enter the next undulation. The front of the bubble is extremely smooth, another result of the high discretization of the free surface and the negligible effect of the surface tension. At greater times (Fig. 11(b)), the bubble surface is seen to follow the tube geometry, although it is permanently shifted somewhat from it. The thickness of the remaining film is larger on the right side of each undulation, and smaller on the left side.

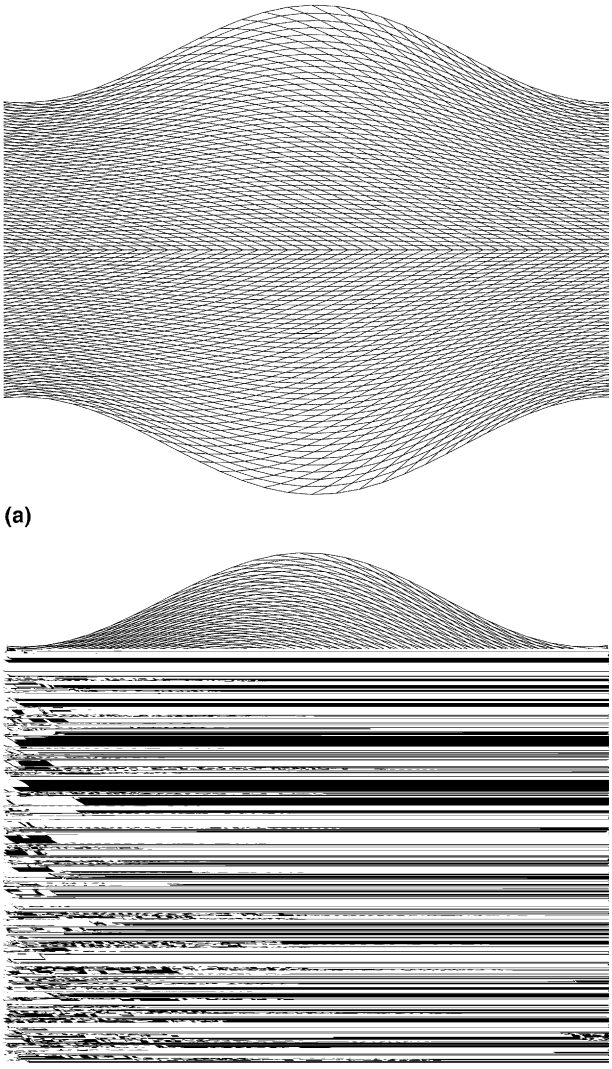

Fig. 12. Closeups of the finite element mesh in the region of the third undulation, at $\tau = 8.38$ and 9.68. $(Re_p, P_{\text{ext}}, R_{\min}/R_{\max}, \varepsilon^{-1}, n) = (10^3, 8333, 0.6, 12, 4)$.

Fig. 12 show the deformation of the mesh in the third spatial undulation after flow has been initiated. Fluid properties and flow conditions are now such that $Re_p = 10^3$, $P_{ext} = 8.33 \times 10^3$, the geometric ratios are identical with those of Fig. 11. Time instants correspond to occasions before (Fig. 12(a): $\tau = 8.38$) and during the penetration (Fig. 12(b): $\tau = 9.68$) of the bubble in that area. Despite the higher concentration of nodes in the entrance of this undulation, the quality of the mesh is high, which is confirmed further by the quality of the generated flow field. Fig. 13(a) and (b) show contour plots of the axial velocity at the same time instances. Here even before the bubble enters this undulation, at $\tau = 8.38$, the flow field is not symmetric due to the finite inertia in the flow. The axial velocity is maximized on the axis of symmetry just after the tube cross-section is minimized. Characteristically, the contours of the axial velocity form homocentric circles around this maximum. Further downstream, the axial velocity is locally minimized, just after the

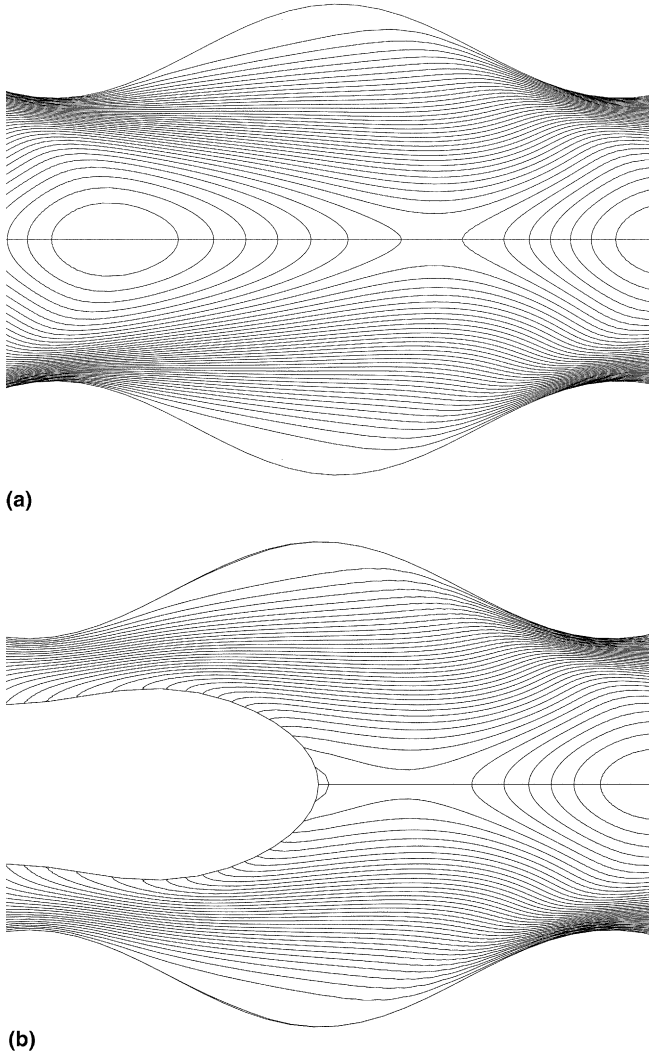

Fig. 13. Closeups of the axial velocity in the region of the third undulation, at $\tau = 8.38$ and 9.68. $(Re_p, P_{ext}, R_{min}/R_{max}, \varepsilon^{-1}, n) = (10^3, 8333, 0.6, 12, 4)$.

tube cross-section is maximized. This pattern repeats itself in all undulations before the bubble front reaches them. Away from the core area the contour lines of the axial velocity tend to follow the tube geometry. When the bubble front enters this undulation, it disrupts the previously described flow field, placing, for example, the location of the axial velocity maximum to its tip.

5. Concluding remarks

We presented a new numerical scheme in order to investigate large displacements of a Newtonian fluid by pressurized air inside tubes of various cross-sections. Our numerical algorithm is a combination of the finite element technique and a new two-dimensional quasi-elliptic transformation with appropriate boundary conditions for generating the time dependent coordinates of the nodes of the mesh. Key elements for the success of this algorithm are the anisotropy of the transformation in the liquid volume and the generalized node distribution along the free surface. The anisotropy is required because the initially vertical to the flow direction free surface is deforming primarily in the axial direction. The operator of the transformation is a modified Laplacian, because the ξ-component of the mesh generation set, where the free surface is mapped, includes an orthogonal contribution in order to avoid discontinuous slopes [25]. Due to the large deformations, the spacing between the R-coordinates corresponding to the ζ-curves near the front of the bubble is very small, forcing the Z-coordinates corresponding to the η-curves to cross them with discontinuous slopes. The orthogonal term makes the Z-curves straighter and assists in keeping the boundary nodes on the front. On the other hand, adjusting the weighting factors so that $w_1 \gg w_2$ in the definition of the arc length along the surface effectively controls the node spacing there and prevents their motion in the radial direction. The imposition of the boundary constraints takes place using the penalty method. For the solution of the resulting set of equations a block Gauss–Seidel method is adopted in order to reduce the memory requirements, the number of Jacobian matrix entries, and generally to decouple the mesh and flow problems. The convergence of the method was guaranteed by using a time-step adaptation scheme, which also controls the stability of the calculations.

The proposed mesh generating scheme was applied in four different geometries: a straight tube, a suddenly constricted one, a complex tube, and an undulating tube. Numerical simulations demonstrated the improved characteristics of our scheme against earlier differential transformations or any algebraic (transfinite) ones. The particularly difficult simulations inside the secondary part of a constricted tube or in complex ones, were done by just changing a few lines of the code to either allow the motion of the nodes on the internal tube/tube boundary, or to add a forcing term in the right hand side of the mesh generating set of equations to keep enough nodes near the tube corners. It should be mentioned that we have applied this method to a wide range of parameter values, $0 \leqslant Re_p \leqslant 5 \times 10^6$, $150 \leqslant P_{\text{ext}} \leqslant 10^4$, and $0.5 \leqslant \varepsilon^{-1} \leqslant 30$, e.g. [37,38]. However the resulting fluid transient velocities of the bubble give rise to a velocity-based Reynolds number that is about 2 orders of magnitude smaller than Re_p and a capillary number that is 2–4 orders of magnitude smaller than P_{ext} depending on the geometry.

Acknowledgements

This work was partially supported under the EPETII program (Grant No. 89, starting year 1998) of the General Secretariat of Research and Technology of Greece and under the EPEAEK program of the Ministry of Education of Greece. We would like to thank Dr. Costas Christodoulou for discussions related to his grid generating method, and to Prof. George Georgiou for offering his plotting subroutines.

References

[1] Z. Tadmor, C. Gogos, Principles of Polymer Processing, Wiley, New York, 1979.
[2] G.I. Taylor, Deposition of a viscous fluid on the wall of a tube, J. Fluid Mech. 10 (1961) 161–165.
[3] B.G. Cox, On driving a viscous fluid out of a tube, J. Fluid Mech. 14 (1962) 81–96.
[4] B.G. Cox, An experimental investigation of the streamlines in viscous fluid expelled from a tube, J. Fluid Mech. 20 (2) (1964) 193–200.
[5] F.P. Bretherton, The motion of long bubbles in tubes, J. Fluid Mech. 10 (1961) 166–188.
[6] C.-W. Park, G. Homsy, Two-phase displacement in Hele Shaw cells: theory, J. Fluid Mech. 139 (1984) 291–308.
[7] D.A. Reinelt, P.G. Saffman, The penetration of a finger into a viscous fluid in a channel and tube, SIAM J. Sci. Statist. Comput. 6 (1985) 542–561.
[8] E.I. Shen, K.S. Udell, A finite element study of low Reynolds number two-phase flow in cylindrical tubes, J. Appl. Mech. 52 (1985) 253–256.
[9] M.D. Giovedoni, F.A. Saita, The axisymmetric and plane cases of a gas phase steadily displacing a Newtonian liquid-A simultaneous solution of the governing equations, Phys. Fluids 9 (8) (1997) 2420–2428.
[10] S.F. Kistler, L.E. Scriven, Coating flows, in: J.R.A. Pearson, S.M. Richardson (Eds.), Computational Analysis of Polymer Processing, Applied Science, London, 1984, pp. 243–299.
[11] B.T. Helenbrook, Mesh deformation using the biharmonic operator, Int. J. Numer. Meth. Eng. 56 (2003) 1007–1021.
[12] C.S. Frederiksen, A.M. Watts, Finite-element method for time-dependent incompressible free surface flow, J. Comput. Phys. 39 (1981) 282–304.
[13] P.A. Sackinger, P.R. Schunk, R.R. Rao, A Newton–Raphson pseudo-solid domain mapping technique for free and moving boundary problems: a finite element implementation, J. Comput. Phys. 125 (1996) 83–103.
[14] R.A. Cairncross, P.R. Schunk, T.A. Baer, R.R. Rao, P.A. Sackinger, A finite element method for free surface flows of incompressible fluids in three dimensions. Part I. Boundary fitted mesh motion, Int. J. Numer. Methods Fluids 33 (2000) 375–403.
[15] R.A. Cairncross, P.R. Schunk, T.A. Baer, R.R. Rao, P.A. Sackinger, A finite element method for free surface flows of incompressible fluids in three dimensions. Part II. Dynamic wetting lines, Int. J. Numer. Methods Fluids 33 (2000) 405–427.
[16] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys. 100 (1992) 25.
[17] A.J. Poslinski, J.A. Tsamopoulos, Inflation dynamics of fluid annular menisci inside a mold cavity-I. Deformation driven by small gas pressure, Chem. Eng. Sci. 46 (1) (1991) 215–232.
[18] A.J. Poslinski, D. Orlicki, J.A. Tsamopoulos, Inflation dynamics of fluid annular menisci inside a mold cavity-II. Deformation driven by large gas pressure, Chem. Eng. Sci. 46 (2) (1991) 597–608.
[19] G. Ryskin, L.G. Leal, Orthogonal mapping, J. Comput. Phys. 50 (1980) 71–100.
[20] K.N. Christodoulou, L.E. Scriven, Discretization of free surface flows and other moving boundary problems, J. Comput. Phys. 99 (1992) 39–55.
[21] K. Tsiveriotis, R.A. Brown, Boundary-conforming mapping applied to computations of highly deformed solidification interface, Int. J. Numer. Methods Fluids 14 (1992) 981–1003.
[22] K. Tsiveriotis, R.A. Brown, Solution of free-boundary problems using fnite-element/Newton methods and locally refined grids: application to analysis of solidification microstructure, Int. J. Numer. Methods Fluids 16 (1993) 827–843.
[23] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 32 (1982) 199–259.
[24] J. Saltzman, J. Brackbill, Applications and generalizations of variational methods for generating adaptive meshes, in: Numerical Grid Generation, Elsevier, Amsterdam, 1982, pp. 865–884.
[25] J.F. Thompson, Z.U.A. Warsi, C.W. Mastin, Numerical Grid Generation, North-Holland, Amsterdam, 1985.
[26] R.I. Kreis, F.C. Thames, H.A. Hassan, Application of a variational method for generating adaptive grids, AIAA J. 24 (3) (1986) 404–410.
[27] A.S. Dvinsky, Adaptive grid generation from harmonic maps on Riemannian manifolds, J. Comput. Phys. 95 (1991) 450–476.
[28] C.W. Mastin, J.F. Thompson, Elliptic systems and numerical transformations, J. Math. Anal. Appl. 62 (1978) 52.
[29] A.M. Winslow, Numerical solution of quasilinear Poisson equation in nonuniform triangle mesh, J. Comput. Phys. 1 (1967) 149–172.
[30] O.C. Zienkiewicz, R.L. Taylor, The finite element method, McGraw-Hill, New York, 1994.
[31] M. Fortin, Old and new finite elements for incompressible flows, Int. J. Numer. Methods Fluids 1 (1981) 347–364.
[32] K.J. Ruschak, A method for incorporating free boundaries with surface tension in finite element fluid-flow simulators, Int. J. Numer. Methods Eng. 15 (1980) 639–648.
[33] B.T. Helenbrook, A two-fluid spectral-element method, Comput. Methods Appl. Mech. Eng. 191 (2001) 273–294.

[34] P.M. Gresho, R.L. Lee, R.L. Sani, On the time-dependent solution of the incompressible Navier–Stokes equations in two and three dimensions, in: C. Taylor, K. Morgan (Eds.), Recent Advances in Numerical Methods in Fluids, Pineridge Press, Swansea, UK, 1980, pp. 27–81.
[35] R. Keunings, An algorithm for the simulation of transient viscoelastic flows with free surfaces, J. Comput. Phys. 62 (1986) 199–220.
[36] B. Cockburn, Devising discontinuous Galerkin methods for non-linear hyperbolic conservation laws, J. Comput. Appl. Math. 128 (2001) 187–204.
[37] Y. Dimakopoulos, J. Tsamopoulos, Transient displacement of a viscoplastic material by air in straight and suddenly constricted tubes, J. Non-Newt. Fluid Mech. 112/1 (2003) 43–75.
[38] 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.
[39] W.J. Gordon, C.A. Hall, Construction of curvilinear coordinate systems and applications to mesh generation, Int. J. Numer. Methods Eng. 7 (1973) 461–477.
[40] S.A. Coons, Surfaces for computer-aided design of space forms, Projects MAC, Design Div., Dept. Mech. Engng., Mass. Inst. Available from: Clearing house for Federal Scientific-Technical Information, NBS, Springfield, VA, USA, 1964.