Journal of Computational Physics 228 (2009) 1980–2011

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

On the elliptic mesh generation in domains containing multiple inclusions and undergoing large deformations

N. Chatzidai, A. Giannousakis, Y. Dimakopoulos, J. Tsamopoulos *

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

ARTICLE INFO

Article history:
Received 26 April 2008
Received in revised form 22 September 2008
Accepted 14 November 2008
Available online 3 December 2008

Keywords:
Moving boundary problems
Elliptic mesh generation
Structured meshes

ABSTRACT

We present an improved method to generate a sequence of structured meshes even when the physical domain contains deforming inclusions. This method belongs to the class of Arbitrary Lagrangian–Eulerian (ALE) methods for solving moving boundary problems. Its tools are either (a) separate mappings of the domain boundaries and enforcing the node distribution on lines emanating from singular points or (b) domain decomposition and separate mappings of each subdomain using suitable coordinate systems. The latter is shown to be more versatile and general. In both cases a set of elliptic equations is used to generate the grid extending in this way the method advanced by Dimakopoulos and Tsamopoulos [Y. Dimakopoulos, J.A. Tsamopoulos, A quasi-elliptic transformation for moving boundary problems with large anisotropic deformations, J. Comput. Phys. 192 (2003) 494–522]. We shall present examples where this earlier method and all other mesh generating methods which are based on a conformal mapping or solving a quasi-elliptic set of PDEs fail to produce an acceptable mesh and accurate solutions in such geometries. Furthermore, in contrast to other methods, appropriate boundary conditions and constraints such as, orthogonality of specific mesh lines and prespecified node distributions on them, can be easily implemented along a specific part of the domain or its boundary. Hence, no attractive terms at specific corners or singular points are needed. To increase the mesh resolution around the moving interfaces while keeping low the memory requirements and the computational time, a local mesh refinement technique has been incorporated as well. The method is demonstrated in two challenging examples where no remeshing is required in spite of the large domain deformations. In the first one, the transient growth of two bubbles embedded in a viscoelastic filament undergoing stretching in the axial direction is examined, while in the second one the linear and non-linear dynamics of two bubbles in a viscous medium are determined in an acoustic field. The large elasticity of the filament in the first case or the large inertia in the second case coupled with the externally induced large deformations of the liquid domain requires the accurate calculation which is achieved by the method we propose herein. The governing equations are solved using the finite element/Galerkin method with appropriate modifications to solve the hyperbolic constitutive equation of a viscoelastic fluid. These are coupled with an implicit Euler method for time integration or with Arnoldi's algorithm for normal mode analysis.

© 2008 Elsevier Inc. All rights reserved.

1. Introduction

The accurate numerical simulation of many scientifically and technologically important processes with free or moving boundaries has been a challenging research area for many decades. Computational methods for numerical simulation of such

* Corresponding author. Tel.: +30 2610 997 203; fax: +30 2610 993 255.
E-mail address: tsamo@chemeng.upatras.gr (J. Tsamopoulos).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.11.020

problems require powerful discretization techniques based on the use of appropriate grids. These are discrete sets of points along lines well covering the physical domain, which must satisfy some specific properties like smoothness and orthogonality along certain boundaries so that the accuracy and the stability of the calculation is guaranteed [1]. In particular, orthogonality only near the moving boundary is often necessary to achieve accurate results, for example, in simulations of flows where boundary layers arise.

Depending on the method generating the computational grid, the available numerical approaches can be classified as Lagrangian, Eulerian and mixed Lagrangian–Eulerian. In the Lagrangian formulation, the coordinate system is moving with the fluid following its local velocity. This method has several useful properties since the interfaces can be specifically delineated and precisely followed, the free-surface boundary conditions are easily applied and curved grid boundaries of any arbitrary shape can be treated. However, it has some significant disadvantages since the grid becomes very often severely distorted and consequently demands multiple reconstructions. Indicatively, Saksono and Peric [2] have used a Lagrangian method for simulating the oscillations of droplets and the stretching of a liquid bridge, where a remeshing procedure proved to be an essential feature for improving the overly distorted finite elements. On the other hand in the Eulerian formulation, the grid points remain fixed relative to the observer, while the fluid moves through the cells. This method has the advantage of being able to handle extreme interface distortions, but the interface is not as sharp and requires a special procedure to locate it with some accuracy. Hirt and Nichols [3] implemented the Volume of Fluid (VOF) method in an Eulerian hydrodynamics code in order to study a variety of highly complicated free surface flows. A significant improvement for Eulerian methods is the front tracking method [4,5] because it determines the interface explicitly using a separate grid for it in addition to that used for the fluid volume.

An intermediate approach is the arbitrary Lagrangian–Eulerian (ALE) method, in which each node of the mesh moves independently of the local fluid velocity. Its main advantage is that it reduces the number of remeshing procedures, and consequently it reduces the projection errors that are introduced after such a remeshing cycle, especially in the continuity equation. However, remeshing techniques still may be needed even in the ALE method. This can be minimized by employing a curvilinear coordinate system that conforms to the moving boundary. The irregular physical domain is mapped onto a simple and time-independent computational one in which the moving boundary coincides with one of the coordinate surfaces (or part of it) and in which it is trivial to generate the mesh. There are two ways to do this by using either algebraic expressions or partial differential equations. The algebraic grid generation methods are based on simple interpolation functions [6]. However, with this procedure problems arise related to smoothness, node overlapping and grid-line folding [7]. Hence heuristic rules must be introduced for controlling the mesh development. Moreover, this technique is restricted to simple initial shapes and small deformations.

The elliptic mesh generation schemes are based on the solution of a partial differential equation (PDE) for each computational coordinate; see an extensive review by Thompson et al. [8]. The simplest of the mesh generation algorithms is conformal mapping, where the Cauchy–Riemann equations are satisfied. Conformal meshes are smooth and orthogonal and when the boundary shape is the only constraint on the mesh generated, they are usually the most efficient ones [8]. However, conformal mapping does not allow control of mesh spacing. Orthogonal meshes [9,10] become then the simplest choice, since they are less restricted than conformal meshes. Christodoulou and Scriven [10] in particular have extended the earlier work in [9] incorporating new features in the system of the differential equations, by minimizing a functional which quantifies the deviation of the mesh from an orthogonal one that satisfies generalized Cauchy–Riemann equations and by including forcing terms. This work was further improved by Tsiveriotis and Brown [11] who proposed a 'mixed mapping method'. It seemed to allow independent control of mesh spacing in each direction which is important when deformations of the free surface preferentially arise in one direction. To improve the accuracy of the solution vector on the interface, they also proposed a non-conforming two-to-one element splitting scheme [12], for the transition from a coarser mesh in the bulk to a finer one close to the interface.

All these ideas have formed the basis for the elliptic grid generation scheme introduced by Dimakopoulos and Tsamopoulos [13]. It is a very robust and flexible method which takes into consideration all the intrinsic features of the developing surface and the deforming control volume. It has been tested in a wide range of numerical simulations [13–22] and proved very satisfactory even at very large deformations. The main advantages of this elliptic grid generation scheme are

- It can be used to solve a variety of problems such as: transient problems [13–20], steady state problems [21,22], and determine their stability [23] requiring minimum changes in the numerical algorithm and so a minimum input from the user.
- Large deformations can be simulated with minimum or even without remeshing cycles. Thus, it reduces errors due to frequent variable interpolations between meshes.
- Structured meshes can be adopted for minimizing the error dispersion/amplification [24] and improve numerical stability and accuracy.

A similar method is the pseudo-solid mesh generation technique (Sackinger et al. [25]), which treats the material as a compressible, elastic solid and solves Cauchy's equilibrium equation to determine the location of the nodal positions. However, with this method remeshing is often necessary since it has not been adopted to allow large distortions of the mesh [25,26]. Actually this issue is closely related with the boundary conditions that should be used. These must be physically consistent with the governing equations, which is not always possible. Such natural conditions for the pseudo-solid method do not include important constraints such as prespecified node distribution or line orthogonality along specific parts either of the boundary or in the interior of the control volume leading to severe distortion of the mesh near a static contact point (see Fig. 14 in [26]).

All these ALE-type methods for generating a structured grid fail when the domain contains inclusions and especially if their shapes undergo large deformations or the simulations are carried out for long times. This is expectable because they inevitably smooth out the mesh near any singular point, such as a corner (slope discontinuity of any variable on any boundary) or a point of large curvature or the poles of the bubbles included in the physical domain in our specific applications. This smoothing appears as a repulsion of the mesh lines away from such singular points. On the contrary, this is exactly where a higher discretization is needed for accurate computations. This makes imperative the special and careful treatment of the inclusions.

Body-fitted curvilinear coordinates for domains containing 2D bodies with fixed boundaries have been proposed earlier. The simpler such method maintains the connectivity of the domain and represents the inclusion(s) by empty slab(s) or slit(s) [27]. Then, both coordinate lines in computational space experience a discontinuity at certain points on its boundary. As explained above, this leads to failure of computations, and more so in moving boundary problems. The second method [27–29] transforms the domain into a simply connected one by introducing as many branch-cuts as needed at arbitrary positions to connect once the inclusions with each other and another branch-cut to connect one of the internal bodies with the external boundary of the domain. Typically, one of the coordinates is assigned the same constant value on all the internal interfaces and all the branch-cuts between the bodies and another value at the external boundary, whereas the other coordinate takes two different constant values on the cut between one of the bodies and the outer interface. This is an O-type opening of the domain. A different positioning of the cuts leads to a C-type opening, etc. In each one of the numerous computational domains that may be created [27], this procedure makes even the interfaces of the inclusions in physical space external boundaries transforming the multiply connected domain to a simply connected one. Each cut appears as two segments on the transformed boundary, each segment corresponding to the two branches of the cut in physical space. However, there at least two disadvantages in applying this method: (i) the two branches of each cut comprise re-entrant boundaries and one of them has a different orientation in computational space requiring special treatment to impose continuity of the dependent variables and their derivatives there and (ii) the resulting mesh may not have the desired distribution of nodes and may require a lot of effort to adjust them as needed. Moreover, in deforming inclusions this adjustment may require modification in time, thus increasing the computational effort and time. The last available method [27,30,31] breaks up the physical domain into several smaller blocks or subdomains and then generates separate meshes in each individual block. Hence, it has been called multiblock or block-structured or domain decomposition method. It can be used in either simply or multiply connected domains. In the second case it may be coupled with the previous method of introducing branch-cuts. Each subdomain is chosen to be geometrically much simpler than the entire configuration and, thus, to be more easily discretizable by a quasi-elliptic method, for example. On the boundaries of each subdomain mesh points are introduced just as in actual domain boundaries. Across the boundaries of the subdomains the meshes maybe unmatched or patched without enforcing continuity of mesh lines or with enforcing continuity of the mesh lines or even of their slopes. In the above order, these three possibilities offer more advantages to the numerical solutions, but also become more difficult in generating the mesh. For example, when mesh line continuity is enforced, the location of the interface nodes requires an additional data indexing procedure to link the subdomains across the interfaces. In spite of generating meshes of higher quality, the difficulty in automating the multiblock method with mesh continuity has inhibited its application to moving boundary problems.

The main contribution of this paper is the presentation of a new methodology for generating accurate meshes in domains with deforming inclusions with or without domain decomposition. The presentation is confined to problems with axial symmetry in the interest of reducing the storage requirements and the computing time. Clearly these problems have their 3D counterparts and, because of their technological importance, we are currently extending the present numerical method to deal with them. We test our scheme and we present results for two problems of scientific and technological interest: (i) bubble growth in viscoelastic filaments undergoing stretching and (ii) bubble interactions in an acoustic field fully accounting for viscous effects. In addition, we apply our numerical technique along with an appropriate mesh refinement methodology in order to limit the large number of grid points only in regions where they are needed the most.

The physics and motivation to study the chosen two problems are given in the following two subsections. The governing equations are given in Section 2 and the general solution methods in Section 3. Section 4 is dedicated to the specific procedures and methods for mesh generation in domains with deforming inclusions. There we will show that the first example can be solved accurately using a single domain with different mappings of segments of the physical boundaries to segments of the computational boundaries and with imposing the node distribution along lines that emanate from the poles of the bubbles. These lines play the role of pseudo-boundaries. In essence, this method corresponds to introducing branch-cuts along the axis of symmetry and using the second of the three methods we discussed above for solving problems with inclusions. This idea, when applied to the second example, fails completely as we show with the two most promising mappings among a variety of mappings we have tried. Indeed, the resulting grids are too skewed and irregular and cannot follow closely the deforming boundaries of the inclusions. Now splitting the domain to subdomains, each one having its own curvilinear coordinate system, becomes imperative. The entire region is treated as a single one when writing and solving the governing equations. The curved interfaces bounding the subregions form internal interfaces across which information must be transferred. Thus, this problem can be solved accurately for long times by efficiently advancing some of the ideas from the third of the methods mentioned above. In Section 5 we give numerical results using the optimum methods for each problem.

1.1. Bubble growth in Newtonian and viscoelastic filaments undergoing stretching

Bubbles or cavities develop in thin films of materials made by block copolymers (such as those based on styrene–isoprene triblocks or acrylates) that are extensively used nowadays as self-adhesives or pressure sensitive adhesives (PSAs). PSAs, in particular, have the ability to average stresses over large volumes of material, thus avoiding the sharp stress concentrations responsible for the failure of structural glassy adhesives. Controlling their adhesive properties is important, since it governs their suitability in non-structural bonding applications. Additionally, understanding the role of the small cavities that propagate along the material and cause its fracture at high deformation levels and how they are affected by the viscoelastic properties of the filament is critical in our ability to design new polymeric materials with optimal adhesive properties. In the literature, theoretical or numerical studies of bubble dynamics have been restricted either to spherically symmetric bubbles growing in an infinite non-Newtonian liquid [32,33] or to the transient deformation of a single bubble or droplet in a uniaxial extensional flow of Newtonian or viscoelastic liquids for which the flow field far away from the bubble or droplet is known [34,35]. More recently, Foteinopoulou et al. [19] have studied the deformation of a single bubble in a Newtonian or viscoelastic filament undergoing stretching. Later they extended their work to multiple bubbles that grow and deform simultaneously in a Newtonian liquid [18], but the material deformations remained smaller than in testing experiments because of the need for higher mesh refinement close to the moving interfaces.

1.2. Bubble interactions in an acoustic field fully accounting for viscous effects

Bubble dynamics in acoustic fields has been studied extensively. In particular, the interaction of pulsating bodies in a fluid was first studied by Bjerknes [36,37]. Pulsating bubbles interact with nearby solid surfaces and also, interact with each other with a force that can be attractive or repulsive depending on whether they oscillate in or out of phase, respectively. In the linear limit and for inviscid fluids, it can be shown that its magnitude is proportional to the bubble volumes and inversely proportional to the square of their distance. This type of force is known as the secondary (or mutual) Bjerknes force and is responsible for several interesting dynamic phenomena. The mutual Bjerknes force plays an important role in many acoustic phenomena or engineering applications such as the formation of bubble grapes [38], the purification of liquid melts, the separation of a gas from its solution in a liquid, etc. Over the past decades a significant amount of research has been devoted to the study of single bubble dynamics (see [39] for a review on earlier work on the subject), as well as in the problem of bubble–bubble interaction [40]. The theoretical approach to the secondary Bjerknes force assuming inviscid fluids by Pelekasis and Tsamopoulos [41,42] has produced interesting results in a wide range of forcing frequencies, pressure amplitudes and bubble sizes. The surface of the bubbles has been allowed to deform from spherical retaining its axial symmetry. However, in those studies neglecting fluid viscosity led to very large bubble deformations that frequently ended up in bubble breakup. This makes necessary the examination of the effect of fluid viscosity on the bubble dynamics not only to determine if it reduces these large deformations but also to examine how it modifies the secondary Bjerknes force.

2. Governing equations

In both problems, we consider two initially spherical gas bubbles which are at rest in a stationary fluid. We assume axial symmetry around the line connecting their centers of mass and that the surrounding fluid is incompressible with density ρ^*, whereas the density and viscosity of the gas in the bubbles are much smaller than those of the liquid. Hence in general, the pressure inside the bubbles varies with time only and according to a polytropic law. In what follows an asterisk indicates a dimensional quantity.

The flow in the liquid is governed by the momentum and mass conservation equations, which in their dimensional form are

$$\rho^* \frac{D\mathbf{u}^*}{Dt^*} - \nabla^* \cdot (-P^*\mathbf{I} + \boldsymbol{\tau}^*) = 0, \tag{1}$$

$$\nabla^* \cdot \mathbf{u}^* = 0, \tag{2}$$

where $\frac{D}{Dt^*}$ stands for the convective derivative, ∇^* for the gradient operator, $\mathbf{u}^*$ and P^* are the velocity vector and pressure in the liquid, respectively, and $\boldsymbol{\tau}^*$ is the extra stress tensor, which is generally split into a purely viscous part, $2\mu_s^*\dot{\boldsymbol{\gamma}}^*$ and a polymeric contribution $\boldsymbol{\tau}_p^*$

$$\boldsymbol{\tau}^* = 2\mu_s^*\dot{\boldsymbol{\gamma}}^* + \boldsymbol{\tau}_p^*, \tag{3}$$

where μ_s^* is the Newtonian (solvent) viscosity and $\dot{\boldsymbol{\gamma}}^*$ is the rate of strain tensor defined as $\dot{\boldsymbol{\gamma}}^* = \frac{1}{2}(\nabla^*\mathbf{u}^* + \nabla^*\mathbf{u}^{*T})$. For a Newtonian fluid, $\boldsymbol{\tau}_p^* = 0$, whereas for a polymeric material the viscoelastic part of the extra stress tensor is related to the rate of strain tensor through a constitutive equation that describes the rheology of the polymer. As such we use the differential model that has been proposed by Phan-Thien and Tanner (PTT) [43] assuming affine motion of the polymer chains

$$Y(\boldsymbol{\tau}_p^*)\boldsymbol{\tau}_p^* + \lambda^* \overset{\diamond}{\boldsymbol{\tau}_p^*} - 2\mu_p^*\dot{\boldsymbol{\gamma}}^* = 0, \tag{4}$$

where λ^* is the material relaxation time which is related to its elastic behavior, μ_p^* the polymer viscosity and the symbol $\diamond$ over the viscoelastic stress denotes the upper convected Maxwell derivative defined as

$$\overset{\diamond}{\tau}{}_p^* = \frac{D\tau_p^*}{Dt^*} - (\nabla^*\mathbf{u}^*)^T \cdot \tau_p^* - \tau_p^* \cdot \nabla^*\mathbf{u}^*, \tag{5}$$

where the superscript T stands for the transpose of a tensor and the exponential form [44] of the PPT model is used

$$Y(\tau_p^*) = \exp\left[\frac{\varepsilon_{\rm PTT}}{\mu_p^*}\lambda^* tr\tau_p^*\right]. \tag{6}$$

The parameter $\varepsilon_{\rm PTT}$ in the PTT model imposes an upper limit to the elongational viscosity, which is inversely proportional to this parameter. Moreover $\varepsilon_{\rm PTT}$ is related to the shear and extensional thinning of the polymeric material.

We note that for a Newtonian liquid, the extra stress tensor, τ^*, is not an additional unknown, but it is calculated directly from the velocity field. However, for a viscoelastic model, such as the PTT model chosen here, the stress components are unknown quantities which should be calculated through the constitutive equation. The hyperbolic nature of the latter necessitates for its accurate solution in various viscoelastic flows at high material elasticities, the Elastic-Viscous Split Stress (EVSS-G) method (Brown et al. [45]). This method splits the polymeric part of the extra stress tensor into a purely elastic, $\mathbf{\Sigma}^*$, and a viscous part

$$\tau_p^* = \mathbf{\Sigma}^* + 2\mu_p^*\dot{\gamma}^*. \tag{7}$$

Moreover, an independent and continuous interpolation, $\mathbf{G}^*$, of the components of the velocity gradient tensor, $\nabla^* u^*$, is introduced wherever the latter arises in the constitutive equation:

$$\mathbf{G}^* = \nabla^*\mathbf{u}^*. \tag{8}$$

For more details see [16,19,45]. In summary, for a Newtonian fluid Eqs. (1)–(3) need to be solved with $\tau_p^* = \mathbf{0}$, whereas for a viscoelastic fluid the entire set of Eqs. (1)–(8) must be solved simultaneously.

Along the free surface of the bubbles, the velocity field should satisfy a local force balance between the capillary forces, viscous and elastic stresses in the liquid and pressure inside each bubble i:

$$\mathbf{n} \cdot (-P^*\mathbf{I} + \tau^*) = -P_{gi}^*\mathbf{n} + 2H^*\sigma^*\mathbf{n}, \tag{9}$$

where P_{gi}^* is the pressure inside bubble i, σ^* is the surface tension, assumed to be the same in all liquid/air interfaces, $\mathbf{n}$ is the outward (for the liquid domain) unit normal to each free surface and $2H^*$ is its mean curvature which is defined as

$$2H^* = -\nabla_s^* \cdot \mathbf{n}, \quad \nabla_s^* = (I - \mathbf{nn}) \cdot \nabla^*. \tag{10}$$

The pressure inside each bubble varies following the instantaneous changes in the bubble volume according to

$$P_{gi}^* = P_{gio}^*\left(\frac{V_{io}^*}{V_i^*}\right)^\gamma, \quad i = 1, 2, \tag{11}$$

where P_{gio}^* and V_{io}^* are the initial pressure and the volume of bubble i, V_i^* its instantaneous volume and γ is taken to be equal to 1.4. The volume of each bubble is calculated after each time step through

$$V_i^* = \int\int\int dV_i^*, \quad i = 1, 2, \tag{12}$$

where $dV_i^* = r^* dr^* dz^* d\theta$ when the bubble interface is described in cylindrical coordinates, and $dV_i^* = r_s^{*2} \sin\theta\, dr_s^*\, d\theta\, d\phi$ when spherical coordinates are used. In general, the subscript s indicates spherical coordinates. Additionally, the shapes of the free surfaces are determined by invoking the kinematic condition

$$\frac{D\mathbf{F}^*}{Dt^*} = \mathbf{u}^* \tag{13}$$

where $\mathbf{F}^*$ denotes the position vector of the interface, which is given by the following expressions $\mathbf{F}^* = r^*\mathbf{e}_r + z^*\mathbf{e}_z$ and $\mathbf{F}^* = r_s^*\mathbf{e}_r$, when cylindrical or spherical coordinates are used, respectively. Along the axis of symmetry the usual symmetry conditions are imposed: (a) the normal to the axis component of velocity vector is set to zero and (b) the tangential to the axis component is symmetric.

2.1. Cavities inside a filament undergoing stretching

The liquid filament is assumed to have initially a cylindrical outer surface with a uniform radius R_{co}^* and to be confined between two solid and coaxial disks each of radius R_{co}^* also, located at initial distance H_o^*. The filament is permanently bonded on both disks and is being stretched by pulling the upper disk with a constant velocity U_o^*, while the lower disk remains stationary. Due to the assumption of axial symmetry, both bubbles inside the filament are assumed to lie along its axis of symmetry. Their radii are initially $R_{b1,o}^*$ and $R_{b2,o}^*$, respectively, and their centers are located at distances $h_{1,o}^*$ and $h_{2,o}^*$ above the lower disk. The initial distance of the centers of the bubbles is denoted as $L_o^* \equiv h_{2,o}^* - h_{1,o}^*$. A schematic of the system considered is shown in Fig. 1. As the upper disk is being pulled, the height $H^*(t^*) = H_o^* + U_o^* t^*$ of the filament increases, both bubbles deform and translate along the filament axis, and all liquid/gas interfaces get distorted.

In order to readily describe the filament interface, it is convenient to formulate this problem in cylindrical coordinates, although this choice complicates the description of the bubble surfaces. The center of the coordinate system is located at the center O of the lower disk. The axial symmetry reduces this problem to a two-dimensional one, defined in the region enclosed by the outer surface of the liquid and the axis of symmetry of the cylinder or the bubble surfaces in the radial direction and by the two disks in the axial direction. We scale all lengths with the radius of the first bubble $R^*_{b1,o}$, velocities with the pulling velocity U^*_o, time with $R^*_{b1,o}/U^*_o$ and stresses and pressure with a viscous scale $\mu^* U^*_o/R^*_{b1,o}$ where the total dynamic viscosity is $\mu^* = \mu^*_s + \mu^*_p$. Hereafter, a variable without an asterisk indicates the corresponding dimensionless variable. Similarly, the equations governing this problem, when they are rendered dimensionless, retain their form, except for the momentum balance, the constitutive law and the interfacial force balance which become

$$Re\frac{D\mathbf{u}}{Dt} + \nabla P - \nabla\cdot\mathbf{\Sigma} - 2\nabla\cdot\dot{\gamma} = 0, \tag{14}$$

$$Y(\tau_p)\mathbf{\Sigma} + De\overset{\diamond}{\mathbf{\Sigma}} + 2De(1-\beta)\overset{\diamond}{\mathbf{D}} - 2(1-\beta)(1-Y(\tau_p))\mathbf{D} = 0, \quad Y(\tau_p) = \exp\left[\varepsilon_{\mathrm{PTT}}\frac{De}{1-\beta}tr\tau_p\right], \tag{15}$$

$$\mathbf{n}\cdot(-P\mathbf{I}+\tau) = -P_{gi}\mathbf{n} + \frac{2H}{Ca}\mathbf{n}, \quad i = 1,2,3, \tag{16}$$

where $\mathbf{D} = \frac{1}{2}(\mathbf{G}+\mathbf{G}^T)$ and the four dimensionless numbers that arise are the Reynolds number, $Re = \rho U^*_0 R^*_{b1,0}/\mu^*$, the Deborah number, $De = \lambda U^*_0/R^*_{b1,0}$, which is a measure of fluid elasticity, the solvent viscosity ratio $\beta = \mu^*_s/\mu^*$ and the capillary number $Ca = \mu^* U^*_0/\sigma^*$, expressing the ratio of viscous to capillary forces. In this problem Eq. (16) is applied not only along the bubble–liquid interfaces (then, i = 1, 2 and P_{gi} stands for the time-varying, gas pressure inside bubble i), but also along the liquid–air interface, (then i = 3 and $P_{gi} = 0$ sets the surrounding air pressure to zero). The pressure inside each bubble and the corresponding volume of the bubbles is calculated via Eqs. (11) and (12), respectively. Along the axis of symmetry the usual symmetry conditions are used, written in cylindrical coordinates

$$u_r = 0, \tag{17}$$

$$\frac{\partial u_z}{\partial r} = 0. \tag{18}$$

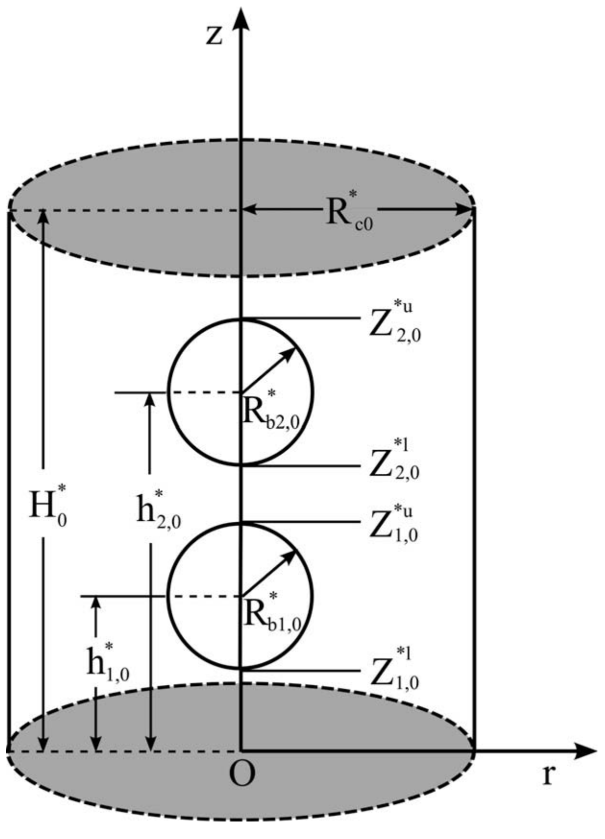

Fig. 1. Schematic of two spherical cavities inside an initially cylindrical filament.

2.2. Interacting bubbles in an acoustic field

Initially, the two bubbles are spherical, surrounded by a viscous liquid and at equilibrium. Hence the initial pressure inside them is set solely by capillarity. Subsequently, they are set in motion by a step change in the far-field pressure. This abrupt change in pressure causes volume oscillations in the bubbles, which, in turn, generate a disturbance in the pressure around them. The later induces a force between them which is always attractive, for a step change in pressure (see Bjerknes [36,37]) and a deformation of their interfaces, depending on the distance between them and the fluid properties. Fig. 2 illustrates a schematic of this flow geometry. The two bubbles are initially spherical with radii R^*_{bA}, R^*_{bB} and distance D^* between their two centers of mass. A convenient way to describe the far-field spherical symmetry of the pressure and velocity fields is to introduce a spherical coordinate system centered at the middle of the distance between the bubble centers. This choice complicates the description of the bubble surfaces, as in the previous problem, while the axial symmetry reduces it to a two-dimensional one defined in the region enclosed by the outer spherical surface and the axis of symmetry or the bubble surfaces.

We scale all lengths with the radius R^*_{bA} of the left bubble. Due to the absence of a characteristic velocity, surface tension is used for making velocity, time and pressure dimensionless. So, we scale them with $(\sigma^*/R^*_{bA}\rho^*)^{1/2}$, $(R^{*3}_{bA}\rho^*/\sigma^*)^{1/2}$ and (σ^*/R^*_{bA}), respectively. The flow is governed by the momentum and mass conservation equations, Eqs. (1) and (2), and only the momentum balance is modified when written in dimensionless form

$$\frac{D\mathbf{u}}{Dt} + \nabla P - Oh\nabla \cdot (\nabla \mathbf{u} + \nabla \mathbf{u}^T) = 0. \tag{19}$$

The dimensionless number that arises in it is the Ohnesorge number, $Oh = (\mu^{*2}_s/\rho^* R^*_{bA}\sigma^*)^{1/2}$, a measure of viscous over inertia and capillary forces. For time greater than zero, the pressure at infinity increases over its static value P_s as

$$P_\infty = P_s(1+\varepsilon), \tag{20}$$

where ε is a measure of the applied disturbance. Along the free surface of the bubbles, we impose Eq. (9) which in dimensionless form here becomes

$$\mathbf{n} \cdot (-P\mathbf{I} + \boldsymbol{\tau}) = P_{gi} + 2H\mathbf{n}, \quad i = A, B, \tag{21}$$

where $i = A, B$ stands for the interface of either bubble. The pressure inside each bubble is calculated via Eq. (11), while their volume is calculated in spherical coordinates via Eq. (12). Along the axis of symmetry, if spherical coordinates are used for the flow problem the symmetry conditions become

$$u_\theta = 0, \tag{22}$$

$$\frac{\partial u_r}{\partial \theta} = 0. \tag{23}$$

The infinite domain of the fluid is truncated to a finite spherical domain around the two bubbles with a radius much larger than the bubble radius so this boundary does not affect the flow around the two bubbles. This is facilitated using the open boundary condition suggested by Papanastasiou et al. [46].

3. Numerical implementation

3.1. Elliptic grid generation

In order to accurately and effectively simulate moving boundary flows in domains with deforming inclusions we have chosen the mixed finite element method to discretize the velocity, pressure and extra stress (for viscoelastic materials only)

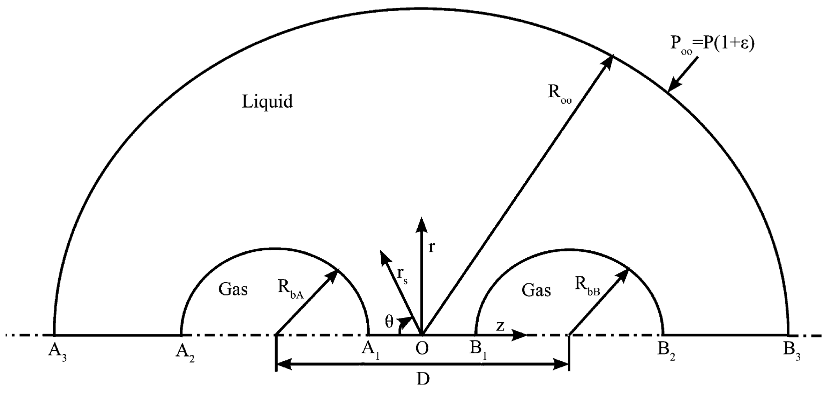

Fig. 2. Schematic of the geometry and the coordinate system of the two interacting bubbles in an acoustic field.

fields, combined with an advanced elliptic grid generator for the initial construction and subsequent motion of the mesh nodes in the liquid domain. The set of equations generating the location of the grid points in the entire domain or in each subdomain is capable to relocate them in response to the evolution of the free surface. Their distribution over the domain is readjusted dynamically to keep a relatively uniform or smoothly varying concentration of nodes in the entire physical domain, without relying on prior knowledge of the interface location and deformation or intervention during the calculations or placing non-physical restrictions on its shape.

In particular, a mapping of the physical (sub)domain is used onto a fixed with time computational domain. This is schematically represented as $(X, Y, t) \xrightarrow{J} (\xi, \eta, \hat{t})$, where J is the Jacobian of the transformation. According to this mapping, every point particle being at time t at a position with coordinates (X,Y) in physical space is mapped to a point particle in a new plane with global coordinates (ξ,η). The (X,Y) coordinates stand for either cylindrical (r,z) or spherical coordinates (r_s, θ). Subsequently, a mesh is generated in computational space with the desired properties. Then the positions of the nodes in physical space are computed by solving a set of partial differential equations. The three essential properties of a mapping are: smoothness, orthogonality and concentration of the (coordinate) curves along which the mesh points lie in physical space [27,47]. The coordinate lines that are parallel to the deforming interface must follow closely its large distortions and even concentrate near it in order to better resolve the boundary conditions there, without the strong requirement that they remain orthogonal to the rest of the boundaries [11]. On the other hand, the coordinate lines normal to the interface must intersect them nearly orthogonally. Keeping these in mind, Dimakopoulos and Tsamopoulos [13], concluded that the following system of quasi-elliptic partial differential equations combines these features

$$\underline{\nabla} \cdot \left\{ \left(\varepsilon_1 \sqrt{\frac{F(\eta)}{F(\xi)}} + (1 - \varepsilon_1) \right) \underline{\nabla} \eta \right\} = 0, \tag{24}$$

$$\underline{\nabla} \cdot \underline{\nabla} \xi = 0. \tag{25}$$

Eq. (24) generates the curves on which η is a constant. The introduction of the term with the square root here, allows them to be nearly normal to the highly deforming interface. Eq. (25) generates the curves on which ξ is a constant. These are nearly parallel to the interface and must follow it while it deforms. Its large deformations may lead to highly obtuse or highly acute angles with its neighboring boundaries. This necessitates relaxation of orthogonality between these other boundaries and the ξ-coordinate lines making the square root term inappropriate in Eq. (25) as clearly demonstrated in [13]. In the expressions above $F(q)$, $q = \xi$, η, is given by

- $F(q) = r_q^2 + z_q^2$ (when cylindrical coordinates are used) (26a)

- $F(q) = r_{s,q}^2 + r_s^2 \theta_q^2$ (when spherical coordinates are used) (26b)

in which when the variable q appears as a subscript it indicates a partial derivative with respect to it. Also, ε_1 is an empirically chosen parameter, ranging between 0 and 1, that controls the smoothness of the mapping relative to the degree of orthogonality of the mesh lines. This parameter is adjusted by trial and error and in the following simulations is set equal to 0.1.

The way the mesh generating equations, Eqs. (24) and (25), are written implies that (ξ,η) are unknown functions, while (X, Y) are the independent variables. This is preferred over its inverse, which is actually the case, because it makes the transformation one-to-one provided that its curvature is non-positive and the boundary of the computational domain is convex. These can be achieved by construction [48]. This property of the differential operator decreases the tendency of grid overlapping. Moreover, the flow equations are originally written in the physical domain with independent variables (X, Y), so we need to transform the entire set so that (ξ,η) are the independent variables. This is achieved through the chain rule differentiation; for more details see [13].

Equally important for the quality of the constructed mesh are the boundary conditions on Eqs. (24) and (25). So, on the fixed part of the boundary we replace the mesh generating equations with the equation that defines the boundary curve, while the remaining degree of freedom is used for controlling the node distribution there

$$\frac{d^2 \widehat{F}(q)}{dq^2} = 0, \quad q = \xi, \eta, \tag{27a}$$

where

- $\widehat{F}(q) = \int_0^q \sqrt{w_1 r_q^2 + w_2 z_q^2} dq$ (when cylindrical coordinates are used) (27b)

- $\widehat{F}(q) = \int_0^q \sqrt{w_1 r_{s,q}^2 + w_2 r_s^2 \theta_q^2} dq$ (when spherical coordinates are used) (27c) and $w_1 + w_2 = 2$. In the above equation, $\widehat{F}(q)$ is a weighted arc-length along the free surface and w_1, w_2 are two weights, which have to be adjusted by trial and error to optimize performance. Setting $w_1 = w_2 = 1$, distributes the nodes equally on the free surface, which is useful only when the deformation of the free surface is not very large in one of the directions. This constraint is imposed via a penalty formulation. More details and examples on the effect of w_1, w_2 are given in [13]. It is noteworthy that for the quality of the constructed mesh, spacing of the nodes can be more important than orthogonality of the coordinate lines. Finally, on the free surfaces, the kinematic condition, Eq. (13), is imposed on them, while the remaining degree of freedom is controlled again by Eq. (27).

When accumulation of the coordinate lines towards certain boundaries is necessary, specific stretching functions are introduced in the computational domain [49]. For example, when an increased concentration of mesh lines is required close to the bottom of the computational domain at the boundary located at $\xi = L$ the following expressions are used

$$Y = \eta, \quad X = K\frac{(\zeta+1) - (\zeta-1)\left(\frac{\zeta+1}{\zeta-1}\right)^{1-\frac{(\xi-L)}{K}}}{\left(\frac{\zeta+1}{\zeta-1}\right)^{1-\frac{(\xi-L)}{K}} + 1}, \tag{28a, b}$$

where K is the length of the domain along the X-direction and ζ is a parameter that controls the density of the coordinate lines and is chosen empirically, $1 < \zeta < \infty$. More specifically the closer to unity ζ is, the denser the coordinate lines are near $\xi = L$.

The discretization of the computational domain is based on triangular elements after the splitting of each rectangular element generated by the above procedure into two triangular ones in a way that will preserve the local symmetries. Triangular elements are preferred because they conform better to large deformations of the physical domain and can sustain larger distortions than the rectangular ones without making the Jacobian of the local transformation to the parent element singular. Further details about the construction of the mesh and the boundary conditions will be discussed for each case separately.

3.2. Local mesh refinement

To increase the local accuracy in regions of particular interest or in regions with sharp variations of the solution vector such as those along the bubble surface or other highly deforming interfaces, the element refinement (h-) method has been applied. The h-method has been suggested by Szabo and Babuska [50] who subdivided the elements on which the measure of the error was larger than a prescribed tolerance. Tsiveriotis and Brown [12], also applied it in a free boundary problem by introducing a transition layer of non-conforming quadrilateral elements, and found that the local refinement technique is essential in cases where elliptic grid generators are used because it relaxes the requirements on the mapping equations and adds more flexibility to the handling of the three important characteristics of the grid mentioned above.

In Fig. 3, the four stages of mesh refinement methodology in two directions around a corner are illustrated. A coarse mesh, initially tessellated in rectangular elements, is refined by adding nodes located in the centroid of each element and in the middle of each element's side. To accommodate the two regions with different number of elements, an intermediate zone of triangular elements is created. In the final stage all the rectangular elements are split into two triangular ones as already described. As we will demonstrate with specific examples this local refinement not only reduces the computational cost, but also increases the accuracy of our calculations considerably. For retaining the bandwidth of the Jacobian matrix as small as possible, the numbering of the nodes and consequently of the unknowns is based on a hierarchical method where nodes with smaller ξ or η (depending on which direction we count first) values are accounted first.

3.3. Mixed finite element method

The velocity vector as well as the position vector of the nodes are approximated with 6-node Lagrangian basis functions, ϕ^i, and the pressure, the stress tensor and the rate of strain tensor for the polymer as well as the velocity gradients are approximated with 3-node Lagrangian basis functions, ψ^i. The finite element/Galerkin method is employed, which after applying the divergence theorem results into the following weak forms of the momentum balance:

$$\int_\Omega \left[Re\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right)\phi^i + \nabla\phi^i\cdot(-P\mathbf{I} + \mathbf{\Sigma} + 2\dot{\gamma})\right]d\Omega - \int_\Gamma [\mathbf{n}\cdot(-P\mathbf{I} + \mathbf{\Sigma} + 2\dot{\gamma})]\phi^i\, d\Gamma = 0, \tag{29a}$$

for the filament undergoing stretching and

$$\int_\Omega \left[\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right)\phi^i + \nabla\phi^i\cdot(-P\mathbf{I} + Oh\boldsymbol{\tau})\right]d\Omega - \int_\Gamma [\mathbf{n}\cdot(-P\mathbf{I} + Oh\boldsymbol{\tau})]\phi^i\, d\Gamma = 0, \tag{29b}$$

for the bubbles in the acoustic pressure field, while the mass balance becomes

$$\int_\Omega \psi^i \nabla\cdot\mathbf{u}\; d\Omega = 0, \tag{30}$$

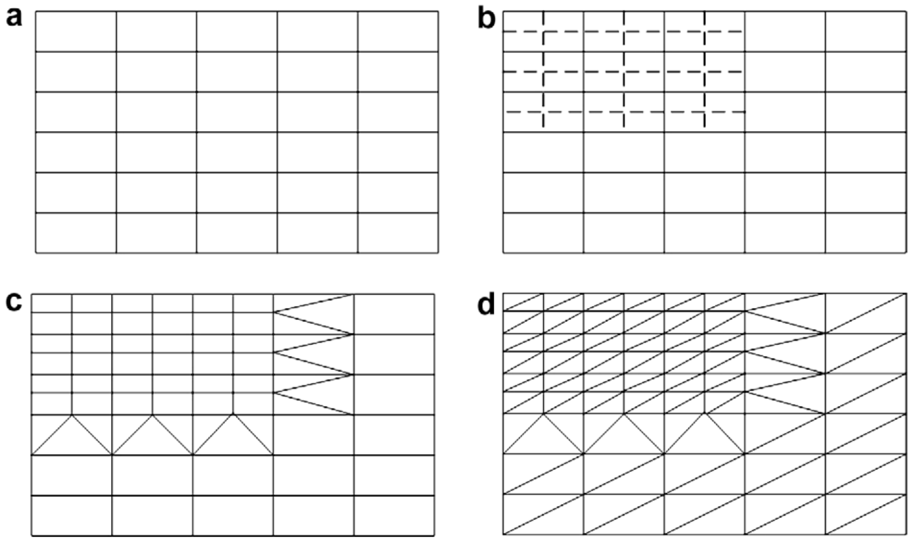

Fig. 3. Stages of mesh refinement: (a) initial mesh with rectangular elements, (b) certain rectangular elements are split in four smaller ones, (c) insertion of transition triangular elements and (d) each rectangular element is split in two triangular ones.

where $d\Omega$ and $d\Gamma$ are the differential volume and surface area, respectively. The surface integral that appears in the momentum equation is split into as many parts as the boundaries of the physical domain and the relevant boundary condition is applied therein. In order to avoid dealing with the second order derivatives that arise in the boundary integral of the interface, through the definition of the mean curvature, H, we use the following equivalent form:

$$2H\mathbf{n} = \frac{d\mathbf{t}}{ds} - \frac{\mathbf{n}}{R_2}, \tag{31}$$

where the first term describes the change of the tangential vector $\mathbf{t}$ along the free surface, R_2 is the second principal radius of curvature, and $\mathbf{n}$ is the unit normal vector on the free surface.

The weak form of the mesh generating equations in general form is derived similarly by applying the divergence theorem

$$\int_\Omega \left(\varepsilon_1\sqrt{\frac{F(\eta)}{F(\xi)}} + (1-\varepsilon_1)\right)\underline{\nabla}\eta \cdot \underline{\nabla}\phi^i \, d\Omega + L\int_\Gamma \frac{\partial \phi^i}{\partial \eta}\sqrt{F(\eta)}d\eta = 0, \tag{32}$$

$$\int_\Omega \underline{\nabla}\xi \cdot \underline{\nabla}\phi^i \, d\Omega = 0, \tag{33}$$

where $F(\eta)$ and $F(\xi)$ are given by Eq. (26), the penalty parameter is $L = O(10^3–10^5)$ and the line integral is along the free surface.

The weak form of Eq. (8) for the velocity gradient tensor is given by

$$\int_\Omega (\mathbf{G} - \nabla\mathbf{u})\psi^i \, d\Omega = 0. \tag{34}$$

Finally, the constitutive equation is a hyperbolic equation for the elastic part of the stress tensor and is solved by the Streamline Upwind Petrov–Galerkin (SUPG) method [51]

$$\int_\Omega \left[Y(\tau_p)\mathbf{\Sigma} + De\overset{\diamond}{\mathbf{\Sigma}} + 2De(1-\beta)\overset{\diamond}{\mathbf{D}} - 2(1-\beta)(1-Y(\tau_p))\mathbf{D}\right]\Psi^i \, d\Omega = 0, \tag{35}$$

where the weighting function Ψ^i is formed from the finite element basis function for the elastic stress tensor according to [52]

$$\Psi^i = \psi^i + \frac{\Delta t_{n+1}}{2}\mathbf{u} \cdot \nabla\psi^i, \tag{36}$$

where Δt_{n+1} is the current time step. The derivation of Eq. (35) is based on the characteristic Galerkin method [53] and guarantees that the upwind term in the modified basis function vanishes as $\Delta t \to 0$.

3.4. Time integration and solution procedure

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. This method has been suggested as a very efficient and robust alternative to the more expensive finite element methods. More specifically, if by $\partial \boldsymbol{\kappa}/\partial t = f(\boldsymbol{\kappa})$ we denote the set of equations to be integrated, its approximate form, using backward finite differences is

$$\frac{\boldsymbol{\kappa}_{n+1} - \boldsymbol{\kappa}_n}{t_{n+1} - t_n} = f(\boldsymbol{\kappa}_{n+1}), \tag{37}$$

where the subscript n stands for the previous time instant and κ is the entire unknown vector which includes velocities, pressure, projected rate of strain and stresses. The difference $t_{n+1} - t_n$ defines the current time step Δt_{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 $\boldsymbol{\kappa}_n$, and an explicitly predicted one $\boldsymbol{\kappa}_n^p$

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

$$\Delta t_{n+1} = \Delta t_n \left(\frac{\varepsilon}{\|\mathbf{d}_n\|} \right)^{1/2} \equiv \Delta t_e, \tag{39}$$

where ε is a user defined tolerance, $\|\cdot\|$ stands for the Euclidean norm and $\mathbf{d}_n = \boldsymbol{\kappa}_n - \boldsymbol{\kappa}_n^p$ is the difference between the predicted and the accurate solution at t_n [54].

The resulting set of algebraic equations is solved by the modified Newton–Raphson iteration scheme. This method proceeds by not updating after each cycle the Jacobian matrix and its factorized form, unless a criterion of decreased convergence rate is violated. Moreover, in the case of transient calculations a Picard iteration scheme is used for solving the set of non-linear equations at each time step and simultaneously decreasing the memory requirements. The total set of equations is split in two sub-sets: the first one consists of the mass and momentum balances and the other one consists of the mesh equations. These are solved independently from each other, using only the necessary information from the other subproblem. When the liquid exhibits viscoelastic behavior, then the extra stress unknowns are also solved separately according to the following algorithm [55]:

For each time step, solve iteratively until convergence *Problems 1–4*:

- *Problem 1:* The mass and momentum balances for $\mathbf{u}_{n+1}^{i+1}$ and P_{n+1}^{i+1} keeping $(X,Y)_{n+1}^{i}$, $\mathbf{G}_{n+1}^{i}$ and $\boldsymbol{\Sigma}_{n+1}^{i}$ fixed.
- *Problem 2:* The mesh equations for $(X,Y)_{n+1}^{i+1}$ keeping $\mathbf{u}_{n+1}^{i+1}$ P_{n+1}^{i+1}, $\mathbf{G}_{n+1}^{i}$ and $\boldsymbol{\Sigma}_{n+1}^{i}$ fixed.
- *Problem 3:* The continuous approximation of the rate of strain tensor $\mathbf{G}_{n+1}^{i+1}$ keeping $\mathbf{u}_{n+1}^{i+1}$ and P_{n+1}^{i+1}, $(X,Y)_{n+1}^{i+1}$, and $\boldsymbol{\Sigma}_{n+1}^{i}$ fixed.
- *Problem 4:* The viscoelastic constitutive equation for $\boldsymbol{\Sigma}_{n+1}^{i+1}$ keeping $\mathbf{u}_{n+1}^{i+1}$ and P_{n+1}^{i+1}, $(X,Y)_{n+1}^{i+1}$ and $\mathbf{G}_{n+1}^{i+1}$ fixed.

This is an efficient method for decoupling the non-linear equations because it results in considerably smaller Jacobian matrices, which are easier to handle. The convergence of the entire 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 [56]. In order to improve the effectiveness of the scheme and to avoid undesirable increases of the time step, an upper bound is placed on the latter by the number of Picard iterations. So the new time step, Δt_{n+1}, is determined by

$$\Delta t_{n+1} = \min(\Delta t_p, \Delta t_e), \tag{40}$$

where Δt_e is given from Eq. (39) and Δt_p is given by

$$\Delta t_p = \Delta t_n \left(\frac{\textit{desired number of Picard cycles}}{\textit{actual number of Picard cycles}} \right)^{1/4}.$$

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 Jacobian matrix of each sub-problem that results after the application of the Newton–Raphson solution technique, is stored in Compressed Sparse Row (CSR) format and the linearized system is solved by using PARDISO, a robust direct sparse matrix solver [57,58]. A Fortran 90 code was written for this purpose and was run on a workstation with dual Xeon CPU at 2.8 GHz.

4. Generation of initial meshes

4.1. Cavities inside a filament undergoing stretching

Although the domain to be discretized contains inclusions because of the presence of the bubbles in the filament, the assumption of axial symmetry makes it simply connected. Actually, the segments of the axis of symmetry connecting each disk to one of the bubbles and the bubbles with each other play the role of branch-cuts in the corresponding 3D geometry. Then as discussed in the introduction, each segment of the axis of symmetry can retain the same position, while each bubble

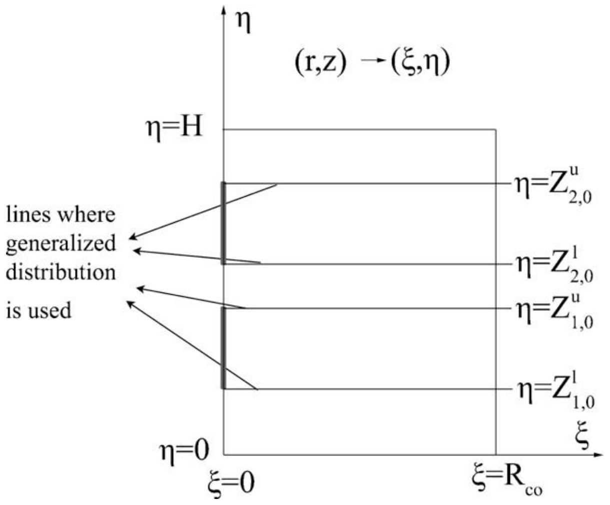

Fig. 4. Computational (ξ,η) domain of the bubbles in a filament problem. The bubble surfaces are mapped onto the thicker line segments.

surface is mapped onto two different segments of the axis of symmetry in the computational domain. In particular, each bubble surface is mapped onto segments with $\xi = 0$ and $Z_i^l \leqslant \eta \leqslant Z_i^u$, where $Z_i^{l,u}$ are the coordinates of the south (l) and north (u) pole of the lower (i=1) or the upper (i=2) bubble. The resulting computational domain is shown in Fig. 4 to be confined between the initial filament surface, which is a perfect cylinder, $\xi = R_{co}$ and its axis of symmetry, $\xi = 0$. Clearly, this shape of the computational domain is different from even the initial shape of the filament containing bubbles of a specific shape and size and, hence, it is necessary to construct the corresponding initial mesh in the physical domain. To do so, we developed a continuation procedure in which the mesh equations, Eqs. (24) and (25), are solved together with the appropriate boundary conditions. As boundary conditions, we impose the locations of all the boundaries of the liquid except for the surfaces of the bubbles and the node distribution equations therein, which are generated by Eq. (27). On the boundaries corresponding to the bubble surfaces we impose the following equations that define the two ellipsoids with the same poles as each bubble

$$\frac{r^2}{b^2} + \frac{(z - z_{i,0})^2}{R_{bi,0}^2} = 1, \tag{41}$$

where $z_{i,0}$ is the initial axial position of the center of bubble i. Initially, the boundary is a straight line coinciding with the axis of symmetry, but subsequently it takes the shape of an ellipsoid when the parameter b in Eq. (41) increases from zero and finally becomes a hemisphere when b is equal to the initial radius of the i bubble, $R_{bi,0}$. These changes in the domain shape and the discretizing mesh must be introduced gradually, so, we increase the value of the parameter b gradually from $0.01R_{bi,0}$ to $R_{bi,0}$ by zeroth order continuation. When the transient problem is solved, Eq. (41) is replaced with the kinematic condition,

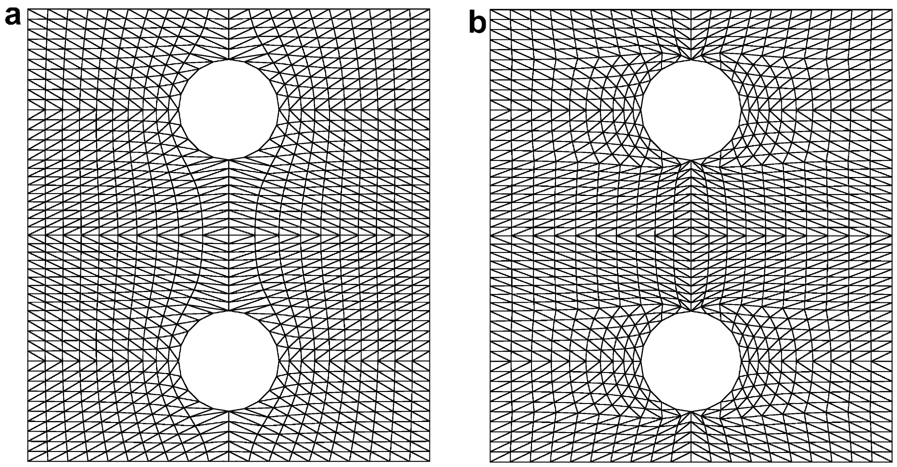

Fig. 5. Initial mesh for the bubble-filament problem (a) using the equidistribution condition only at the boundaries of the mesh problem and (b) using additional generalized distributions along the η-lines that begin from the poles of each bubble and reach the air–liquid interface, with $w_1 = 0.1$ and $w_2 = 1.9$.

Eq. (13), written in cylindrical coordinates. A similar kinematic condition determines the time-dependent location of the outer filament surface.

The mesh that is generated in this manner is shown in Fig. 5(a). For clearer viewing, the mesh we show here is quite coarse, but a much finer mesh was used in order to produce the results in this paper. We observe that the mesh is not acceptable near the poles of the bubbles, where the repulsion of the mesh lines is clearly seen, as discussed in the introduction. An easy remedy of this is to impose the generalized distributions, Eq. (27) along the η-lines, that begin from the poles of each bubble and reach the air–liquid interface (Fig. 4). Since a higher concentration of nodes is needed in the regions closer to the bubbles, along these lines we use $w_1 = 0.1$ and $w_2 = 1.9$ in Eq. (27). The mesh generated after the application of Eq. (27), is shown in Fig. 5(b). It consists of more uniform elements all around the two bubbles. With the introduction of these "internal" lines with preset node distributions, in essence, we introduce internal boundaries in the computational domain which somewhat resembles domain decomposition. Thus, we eliminate the need for adding attractive terms on corners or on the bubble poles in these elliptic equations, something we have used elsewhere successfully [14]. When we introduced such attractive terms on the bubble poles here, the resulting mesh was not as uniform as the one shown in Fig. 5(b). Additionally, the mesh refinement technique described in Section 3.1 is used here due to the large mesh deformations expected in the region near the bubbles and the filament-air free surface and the need to resolve them very accurately, especially because of the hyperbolic nature of the constitutive law. Typical meshes with and without local mesh refinement will be shown in the results section.

4.2. Interacting bubbles in an acoustic field

Given the success and simplicity of the previous method in dealing with the bubbles included in the filament, it was first attempted for this problem also to construct the mesh following a similar method. More specifically, given that here very far from both bubbles the pressure and velocity fields have spherical symmetry, a spherical coordinate system is more appropriate. Also, it seems reasonable to position its center at the middle of the distance between the centers of the two bubbles. The physical domain (r_s, θ) (see Fig. 2) is mapped on the computational domain (ξ, η) (see Fig. 6), as follows: the first and third segments, $0 \leqslant \xi \leqslant \xi_{A1}$ and $\xi_{A2} \leqslant \xi \leqslant R_\infty$, of the line $\eta = 0$ (left boundary) correspond to the line segments OA_1 and A_2A_3 in the physical domain, while its second segment $\xi_{A1} \leqslant \xi \leqslant \xi_{A2}$ corresponds to the surface of the left bubble. Similarly, the first and the third segments, $0 \leqslant \xi \leqslant \xi_{B1}$ and $\xi_{B2} \leqslant \xi \leqslant R_\infty$, of the line at $\eta = \pi$ (right boundary) correspond to the line segments OB_1 and B_2B_3, while its second segment $\xi_{B1} \leqslant \xi \leqslant \xi_{B2}$ corresponds to the surface of the right bubble, which is generally of different size. The line segments $0 \leqslant \eta \leqslant \pi$ at $\xi = 0$ and at $\xi = R_\infty$ correspond to the center of the coordinate system and to the far-field boundary in physical space, respectively. In other words the four segments on the axis of symmetry corresponding to branch-cuts are mapped in pairs on opposite boundaries of the computational domain, while each bubble surface is also mapped on the same opposite boundaries. The center of the coordinate system and the outer circular boundary are mapped on the bottom and top boundaries, respectively, in the computational space. To avoid the repulsion of the mesh lines by the bubble poles, the node locations on the lines of constant ξ emanating from both poles of each bubble are controlled. As in the previous case, this choice of computational domain necessitates the construction of an initial physical domain (containing bubbles of the desired shape and size). This is achieved via the same continuation procedure in which the mesh equations, Eqs. (24) and (25), written in spherical coordinates, are solved together with the boundary conditions which

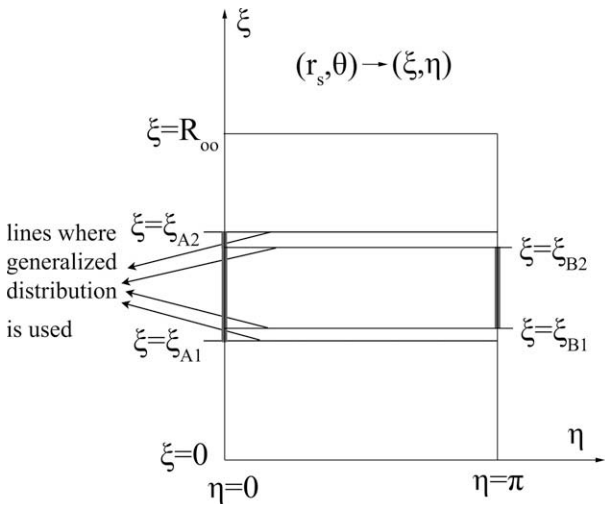

Fig. 6. Computational (ξ, η) domain of the bubble–bubble interaction problem, when the center of the spherical coordinate system is positioned at the middle of the distance between the centers of the two bubbles. The bubble surfaces are mapped onto the thicker line segments.

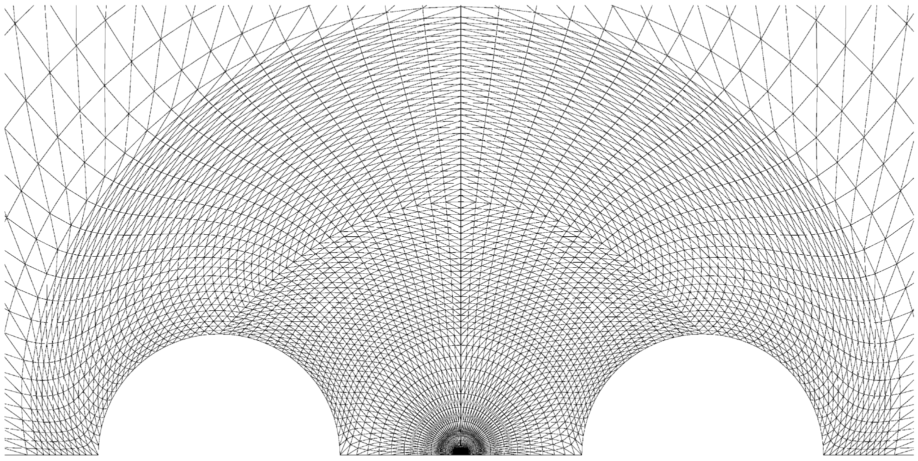

Fig. 7. Part of the initial mesh for the bubble–bubble interaction problem when the mapping of Fig. 6 is used.

set the locations of all the boundaries of the liquid except for the surfaces of the bubbles and the node distribution equations therein, which are generated by Eq. (27). On the bubble surfaces, a condition similar to Eq. (41) is imposed and meshes of gradually increasing values of the parameter b are generated, until both bubbles attain a spherical shape. The resulting initial mesh is shown in Fig. 7, after splitting each rectangle into two triangles in a way that preserves the local symmetries. This mesh seems to be quite satisfactory up for short times only. Unfortunately, as the bubbles get distorted with time the computations fail to converge. This is attributed to the asymmetry in discretizing the two poles of each bubble. This is observed first by the increasing error in pressure at the poles of the bubbles, especially when the bubbles are relatively small and eventually leads to unacceptable errors.

In a second attempt to generate a mesh for accurate calculations, it was decided to remedy this problem by placing the center of the coordinate system at the center of one of the two bubbles, e.g. the right bubble, see Fig. 8(a). Then, the surface of this bubble would be described simply and exactly by these spherical coordinates and, hence, it would be symmetrically discretized. In order to improve the discretization of the left bubble also, we impose the node locations on the ξ-lines starting from the poles of the left bubble at $\eta = 0$ all the way to $\eta = \pi$ on which the generalized node distributions, Eq. (27), are imposed. Finally, the small asymmetry in the far-field boundary condition which is introduced by the presence of the second bubble should not affect the solution since this boundary is not only located very far away from both bubbles but also the open boundary condition is applied there. Hence and although the "branch-cuts" remain the same, the physical domain (r_s, θ) (Fig. 8(a)) is mapped onto a different computational one (ξ, η) (Fig. 8(b)). In particular, the surface of the right bubble is mapped onto the entire bottom boundary $0 \leqslant \eta \leqslant \pi$ at $\xi = R_{bB}$, and the left bubble onto the line segment $\xi_1 \leqslant R_{bA} \leqslant \xi_2$ at $\eta = 0$, the left boundary. This is also a very promising arrangement out of a host of others. The locations ξ_1 and ξ_2 correspond

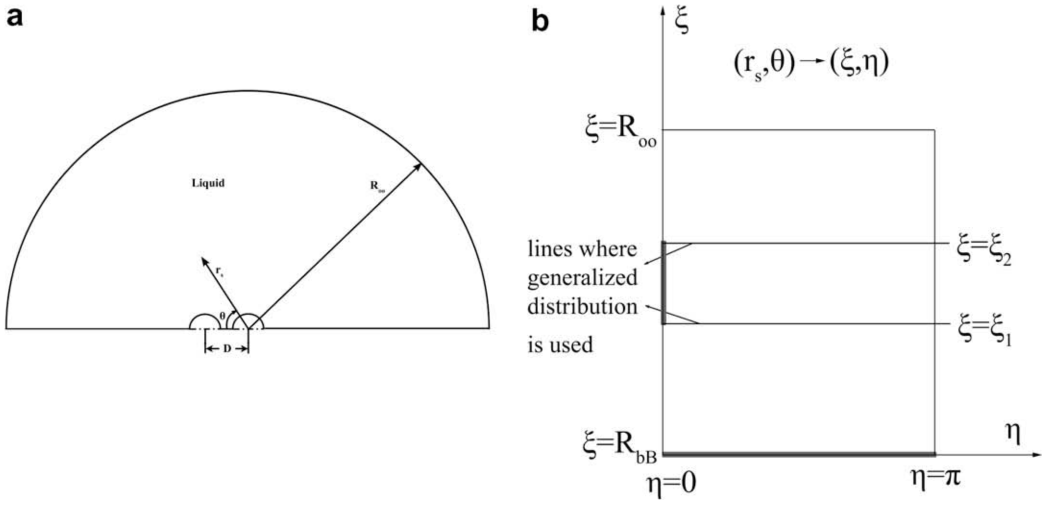
a
Liquid
R_∞
r_s
θ
D
b
ξ
$(r_s,\theta) \rightarrow (\xi,\eta)$
$\xi=R_{oo}$
lines where generalized distribution is used
$\xi=\xi_2$
$\xi=\xi_1$
$\xi=R_{bB}$
η
η=0
η=π

Fig. 8. Illustration of: (a) the coordinate system in physical space and (b) the computational domain of the bubble–bubble interaction problem, when the center of the coordinate system is placed at the center of the right bubble. The bubble surfaces are mapped onto the thicker line segments.

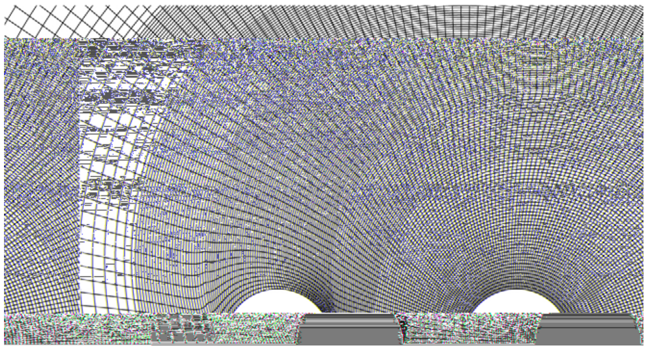

Fig. 9. Part of the initial mesh near the bubbles for the bubble–bubble interaction problem when the mapping of Fig. 8 is used. For clarity we show rectangular elements only.

to the right and left poles, respectively, of the left bubble from where the ξ-lines emanate. In order to construct the two bubbles initially the equation of a circle with radius equal to the radius of the bubble is used directly for the right bubble, while for the left bubble the continuation procedure and Eq. (41) in spherical coordinates is used as described before. The fixed parts of the domain are imposed as boundary conditions, while the node distribution equations, Eq. (27), are imposed at the line of symmetry ($\theta = 0, \theta = \pi$) and at $r_s = R_\infty$. Fig. 9 depicts a close up of the initial mesh around the two spherical bubbles with rectangular elements for reasons of clarity. This initial mesh seems satisfactory. However, drawing the pressure profile around the surface of two equal bubbles at $t = 10^{-4}$ and for $Oh^{-1} = 20$ and $P_s = 28$, Fig. 10 we observe two things: (i) the pressure in both bubble surfaces deviates from its expected value of $\sim$28, something we had noted with the previous mesh to lead to failure of computations and (ii) this deviation is about two orders of magnitude larger in the left bubble the center of mass of which does not coincide with the center of the coordinate system. Moreover, when the transient problem is solved and the bubbles undergo large deformations, the mesh around the left bubble is highly distorted and several elements are quite skewed. This leads to inaccuracies in the solution and to the breaking down of the computations after a while. On the other hand, the mesh around the right bubble remains much less distorted and follows the bubble deformations nicely. An example of the mesh around quite distorted bubbles of equal radius with $Oh^{-1} = 6.5$ and $P_s = 1000$ is shown in Fig. 11.

All the above lead us to the conclusion that in order to obtain optimal results for this geometry, the mesh around the two bubbles must be particularly fine, the elements must remain close to orthogonal and equilateral triangles and, more importantly, both bubble surfaces must be individually discretized, so that the error on their poles is decreased as much as possible. This naturally leads to the requirement that the physical domain (see Fig. 12) is decomposed into three subdomains: (a) one part around each bubble described by a separate spherical coordinate system with a center that coincides with the initial centroid of each bubble (parts A and B) and (b) a third part that will cover the rest of the domain with a center that coincides with the middle of the segment connecting the initial bubble centroids (part C), see Fig. 12(a). All parts are constructed separately using the elliptic grid generation equations (Eqs. (14) and (15)) defined on local spherical coordinate systems. At the final stage, the three sub-domains are reunited to form the original one.

For constructing the meshes around the two bubbles (parts A and B), each physical sub-domain with local coordinates (r_s, θ) is mapped separately to a computational domain with (ξ, η) coordinates. For both subdomains, the two segments from

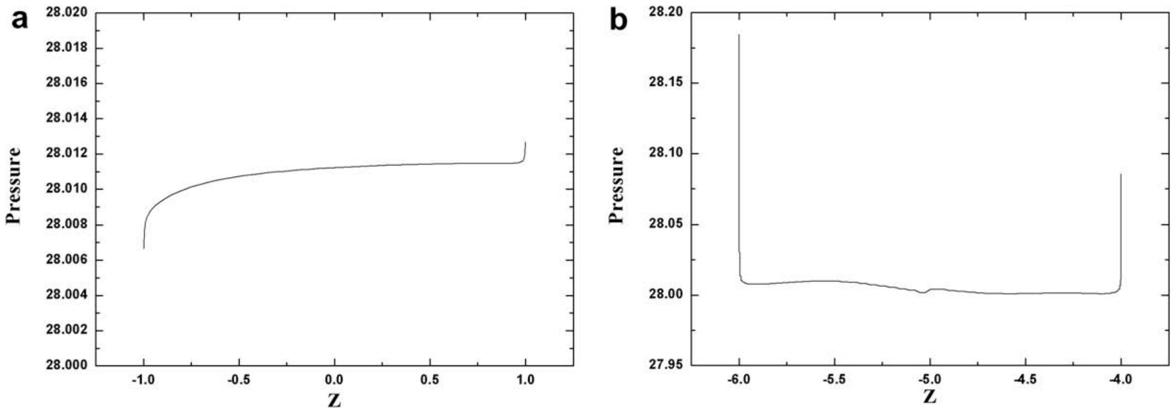

Fig. 10. Pressure variation along the moving surface of (a) the right bubble and (b) the left bubble for the mesh of Fig. 9, at $t = 10^{-5}$, for $R_{bB} = 1$, $Oh^{-1} = 20$, $P_s = 28$, $\varepsilon = 1$, $D = 5$ and $R_\infty = 30$.

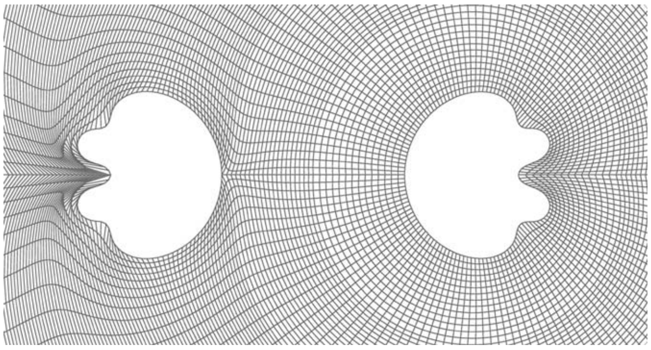

Fig. 11. Deformation of the mesh near the bubbles at $t = 0.3$ when the mapping of Fig. 8 is used for $R_{bB} = 1$, $Oh^{-1} = 6.5$, $P_s = 1000$, $\varepsilon = 1$, $D = 5$ and $R_\infty = 30$. For clarity we show rectangular elements only.

the line of symmetry are mapped to the left and right boundary $R_{bi} \leqslant \xi \leqslant L_i$, $i = A, B$ with $\eta = 0$ or $\eta = \pi$ corresponding to $\theta = 0$ or $\theta = \pi$, see Fig. 12(b) and (c). Each bubble surface is mapped to the bottom boundary $0 \leqslant \eta \leqslant \pi$ with $\xi = R_{bi}$ $(i = A, B)$ so that the center of mass of each bubble coincides with the origin of the local coordinate system. The extent of the subdomain around each bubble, which is also related to the initial distance between the two bubbles, is mapped to the top $\xi = L_i$, where L_i is the radius of the outer spherical boundary of the part i and we take $L_A = L_B$ always. Moreover, the location of the boundaries $A_4C_2E_1$ and $E_1D_2B_4$ in physical space is determined by two sets of equations: (a) up to a position that corresponds to the local angle $0 \leqslant \theta \leqslant 3\pi/4$ for part A and to the local angle $\pi/4 \leqslant \theta \leqslant \pi$ for part B we impose the equation of a circle with a radius equal to the length L_i of each part, (b) for the rest of each boundary, OE_1, we impose the equation of a straight line, see Fig. 12(a). The remaining degrees of freedom at all boundaries are used to distribute the nodes at equal distances, hence $w_1 = w_2 = 1$, along these boundaries through Eq. (27) in a spherical coordinate system. Having constructed the mesh for parts A and B the location of the boundaries $A_4C_2E_1$ and $E_1D_2B_4$ and the nodes on them are used as input data for the boundary $A_4C_2E_1D_2B_4$ of part C. More specifically, first we define the third physical domain, C, to be a spherical shell with center at the middle of the distance between the centroids of the two bubbles, inner radius L_c $(L_c = OA_4 = OB_4)$ and outer radius equal to the radius where the far-field boundary condition is applied, R_∞. Then this physical domain is mapped to a computational domain with (ξ, η) coordinates that is bounded by the left and right boundaries $L_c \leqslant \xi \leqslant R_\infty$ at $\eta = 0$ or $\eta = \pi$ corresponding to line segments on the axis of symmetry, A_4A_3 and B_4B_3, respectively, while its inner and outer spherical surface correspond to

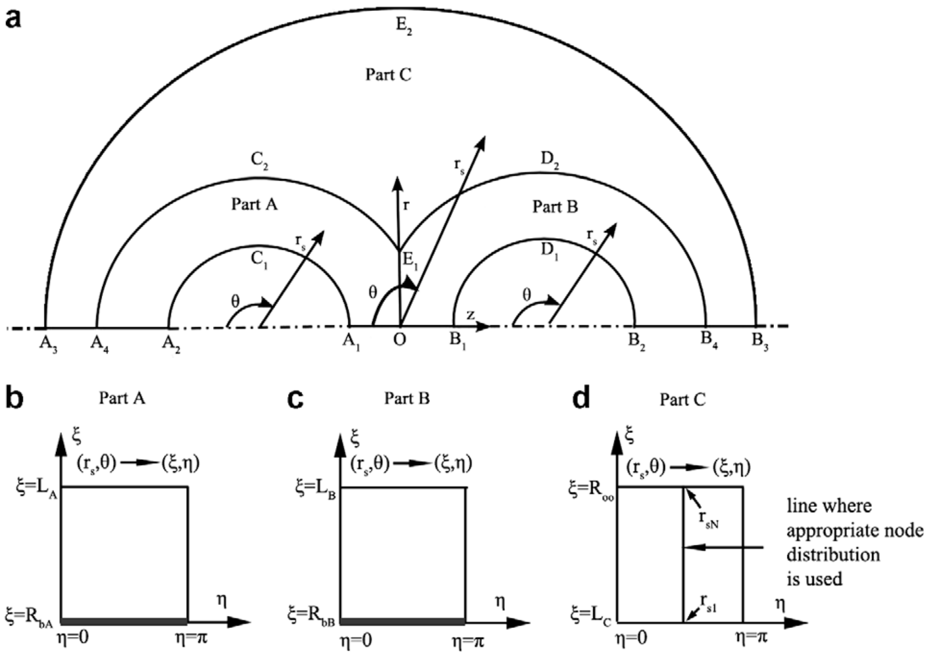

Fig. 12. (a) Schematic of the decomposition of the physical domain into three parts, and the mapping to a computational (ξ, η) domain of (b) part A, (c) part B and (d) part C. The bubble surfaces are mapped onto the thicker line segments.

$0 \leqslant \eta \leqslant \pi$ at $\xi = L_C$ or at $\xi = R_\infty$, see Fig. 12(d). Subsequently, the initial physical domain is constructed by the previously described continuation procedure. In particular, here the inner spherical surface is pulled towards the $A_4C_2E_1$ D_2B_4 boundary in physical space. Simultaneously, the locations of the nodes, along the $\eta = \pi/2$-line that corresponds to the line E_1E_2 are relocated according to

$$\frac{r_{si} - R_\infty}{\xi_i - R_\infty} = \frac{r_{s1} - R_\infty}{L_C - R_\infty}, \quad i = 1, 2, \ldots, N, \tag{42}$$

where N is the total number of nodes along this η-line, r_{si}, ξ_i are the coordinates of node i on it in physical and computational space, respectively and r_{s1} is the radial coordinate of the first node on it that changes at each continuation step. For the degrees of freedom setting the locations of the nodes on all boundaries, Eq. (27) with $w_1 = w_2 = 1$ is used. Moreover, for each computational domain Eq. (28) is used as needed for concentrating the coordinate lines along the corresponding inner boundary $\xi = R_{bA}, R_{bB}$, or L_c, with an appropriate value of ζ: usually $1.05 \leqslant \zeta \leqslant 1.2$. After the construction of the three individual domains, we construct the entire mesh by merging all three domains. This is also used as an initial state for the transient problem.

The discretization of the entire physical domain is presented in Fig. 13(a), while a magnified view close to the bubbles is presented in Fig. 13(b). Clearly, the mesh nodes are evenly distributed around both bubbles. Despite this fine discretization, it was soon realized that an even finer mesh is required close to the bubble surfaces, in order to compute the eigenvalues of bubble shape deformation to at least three significant digits. This highly sensitive computation is accomplished accurately without unnecessarily increasing the mesh where this is not needed, i.e. away from the bubbles, by following the mesh refinement technique described in Section 3.1. In Fig. 13(c) a magnified view of the mesh in the vicinity of the two bubbles is given. Three refinement levels are used for higher local resolution of the flow. Details of the mesh around the pole of the right bubble are presented in Fig. 13(d) and verify the capability of the proposed method to generate nearly optimal grids. In Fig. 14(a) and (b) we present two unequal bubbles after they have undergone several volume oscillations and have translated towards each other, while in Fig. 15 we present only the right of two equal interacting bubbles. In both cases the liquid is taken to be water and the dimensionless parameters change in response to changes in the size of the bubbles. The combined volume and shape oscillation with bubble translation has resulted in large deformation especially of the right bubble in

Fig. 13. Typical initial mesh for the bubble–bubble interaction problem (a) the entire domain, (b) a region around the bubbles with uniform mesh for the entire domain, (c) a region around the bubbles with a three-level refinement around each bubble and (d) a region around the pole of the right bubble with three-level refinement. In figures (a) and (b) only rectangular elements are shown for clarity.

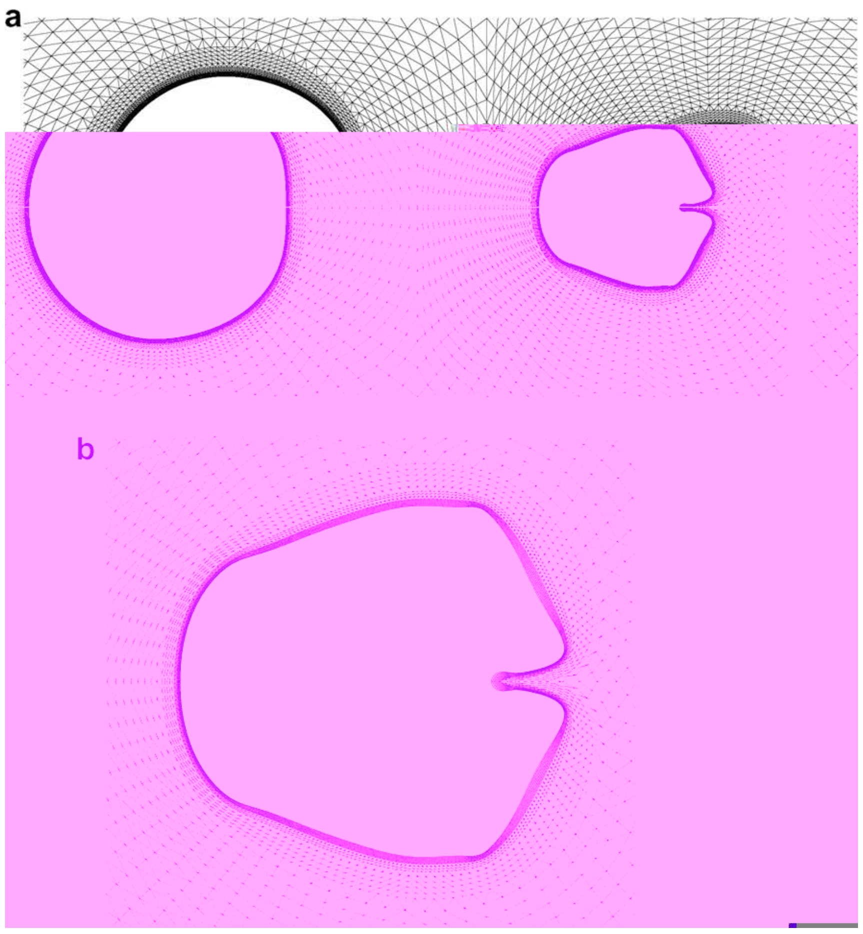

Fig. 14. Deformation of the mesh when the mapping of Fig. 12 is used. (a) a region around both bubbles, (b) a magnification around the right bubble at $t = 0.39$ for $R_{bB} = 0.5$, $Oh^{-1} = 85$, $P_s = 137$, $\varepsilon = 1$, $D = 3$ and $R_\infty = 30$.

Fig. 14 and of both bubbles in the second case (Fig. 15). In fact in the case of Fig. 15, where the inverse Ohnesorge number is large, the bubbles become highly deformed and we are able to capture accurately a very fast moving jet that emanates from the bubble side which is away from the other bubble. Eventually this will pierce the bubble and will cause its collapse, a well known phenomenon. The dynamically adjusting mesh follows very nicely the jet and the swelling that appears in its head along with every other detail on the bubble surface. It is quite challenging to capture this interface with multiple folds of varying amplitude, even if it appeared in a single bubble. This example clearly shows the advantages of splitting the domain into subdomains which closely follow the local interfaces and the quasi-elliptic method for generating the mesh. Characteristics of the specific meshes will be described in the next section.

There are several advantages of the above methodology for generating a mesh. More specifically, (i) the grid is structured with mesh lines remaining nearly orthogonal to the highly deforming boundaries which is essential to very accurately resolving the boundary conditions on these boundaries even at high values of fluid inertia or fluid elasticity; (ii) the grid is structured not only around the above regions where the solution exhibits rapid variation, but also in the entire domain, when it is well known that structured grids have better properties related with error distribution and propagation compared to unstructured ones [59,60]; and (iii) during the transient simulations resulting in large domain deformations grid restructuring is not needed at all and variable interpolations are not required or kept to a minimum. All these benefits are essential in the case of a moving boundary problem because of the hyperbolic character of the kinematic condition, not to mention the hyperbolic character of the viscoelastic constitutive law. For this reason, a moving boundary problem is susceptible to artificial short wave instabilities, unless an optimal discretization technique is adopted. Therefore, using in this case a local orthogonal spherical coordinate system around each bubble, we can decrease the local errors and perform simulations for long times. The implementation is simple, and its specific steps for creating the initial mesh are described above. Subsequently, in order to perform the transient calculations we just replace the essential conditions on the moving boundaries ($r_s = R_{bA}$ & $r_s = R_{bB}$), with the kinematic ones (Eq. (13)) and follow the solution algorithm described in Section 3.2.

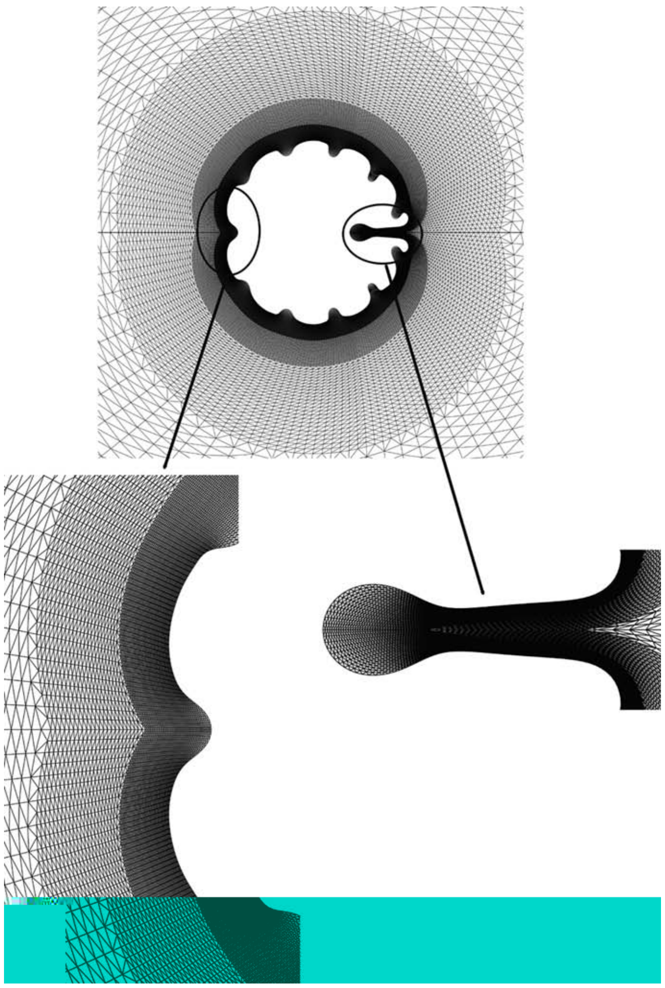

Fig. 15. Deformation of the mesh when the mapping of Fig. 12 is used. The mesh is magnified around the right bubble and the left and right pole of the bubble at $t = 0.651$ for $R_{bB} = 1$, $Oh^{-1} = 270$, $P_s = 1375.51$, $\varepsilon = 0.3$, $D = 9.9$ and $R_\infty = 30$.

It must be mentioned that our numerical experimentation showed that for the solution of the flow problem a global coordinate system can be safely used without introducing any numerical instability. This is also very convenient from a programming point of view, since no extra geometric transformations are necessary.

5. Numerical examples

5.1. Cavities inside a filament undergoing stretching

For simplicity, we assume that the two spherical bubbles are of equal radius ($R_{b1} = R_{b2} = 1$), with their centers located symmetrically with respect to the middle plane of the filament at distances $h_1 = 2$ and $h_2 = 7$ from the lower disk, respectively. The ratio of the initial filament height to the bubble radius is $H_o = 9$, while the initial aspect ratio of the filament is $\Lambda = H_o/R_{co} = 2.25$. The liquid that surrounds the two bubbles is viscoelastic with $De = 1$, $\varepsilon_{PTT} = 0.03$ and $\beta = 0.01$. Since PSAs are materials with very high dynamic viscosities (close to 10^7 Pa s), the resulting Reynolds number is practically zero.

The meshes used to discretize the liquid domain are constructed as discussed in Section 4.1 and are denoted by U or M, indicating either a uniform or a locally refined mesh, respectively. For example mesh U1 consists of 960 triangular elements with 21 nodes around each bubble surface. More details about all the meshes that are used for this problem are given in Table 1. In Fig. 16 we show three snapshots of the filament and the bubbles inside it at times $t = 0$, $t = 5$ and $t = 11.5$. The bubbles remain symmetric with respect to the mid-plane of the filament at all times. The mesh U1 is depicted here. Although this mesh is satisfactory for the relatively small initial deformations of the filament, at longer times and larger deformations non-physical angles appear at the surfaces of the bubbles and particularly at their poles closer to each disk. Clearly, more nodes are needed in order to describe accurately the highly deforming bubble surfaces. However, despite the coarse discret-

Table 1
Characteristics of the meshes used in the filament stretching problem. Triple local refinement has been performed for the M family of meshes around the bubble surface at the locations: $\xi = R_{c0}/20, R_{c0}/10, R_{c0}/5$ and single local refinement has been performed along the filament free surface in the range: $0.9R_{c0} \leqslant \xi \leqslant R_{c0}$.

Mesh	Number of radial elements before refinement	Number of axial elements before refinement	Number of triangles after refinement (if used)	Number of nodes on free surface after refinement	Δr around the free surface after refinement	Δz around the free surface after refinement
U1	10	48	960	21	0.2	0.314
U2	80	84	61440	161	0.025	0.04
M1	10	48	5664	161	0.025	0.04
M2	14	64	10752	225	0.018	0.028

ization, no remeshing techniques are required and even larger deformations of the domain boundaries can be followed. In order to increase the accuracy of the computations, one option is to increase the number of triangular elements throughout the domain to, for example, 61,440 and the number of nodes along each bubble surface to 161. This is mesh U2. Now the surfaces of the bubbles are captured very satisfactorily even after they are highly deformed, but the computational time increases to prohibitive levels. More specifically using 960 triangular elements the simulation requires almost 5 h in order to reach the same final time of $t = 11.5$, while using 61,440 elements the simulation takes almost 10 days. In order to reduce the memory requirements and simultaneously the computational time, while keeping the same accuracy in the calculations, we applied the local refinement technique described in Section 3.1. Here the local refinement was performed in one direction only, the radial one, resulting in unnecessarily finer mesh all along the axis of symmetry. Also three grading levels were introduced around the axis of symmetry and each bubble surface and a single grading level along the liquid–surrounding air interface. The number of the nodes along each bubble surface is retained at 161. However, the total number of the triangular elements decreases significantly from that in mesh U2 to 5664 (mesh M1). The resulting decrease in the CPU time is impressive, since the simulation in this case takes almost 12 h to reach the same final time. If the local refinement was performed in both the radial and the axial directions around the bubbles surfaces only, the reduction in both memory requirements and computation time would have been even larger. In Fig. 17 we show snapshots of mesh M1 at times $t = 0$, $t = 5$ and $t = 11.5$, while in Fig. 18 we show a magnified view around the top and the bottom of the upper bubble at

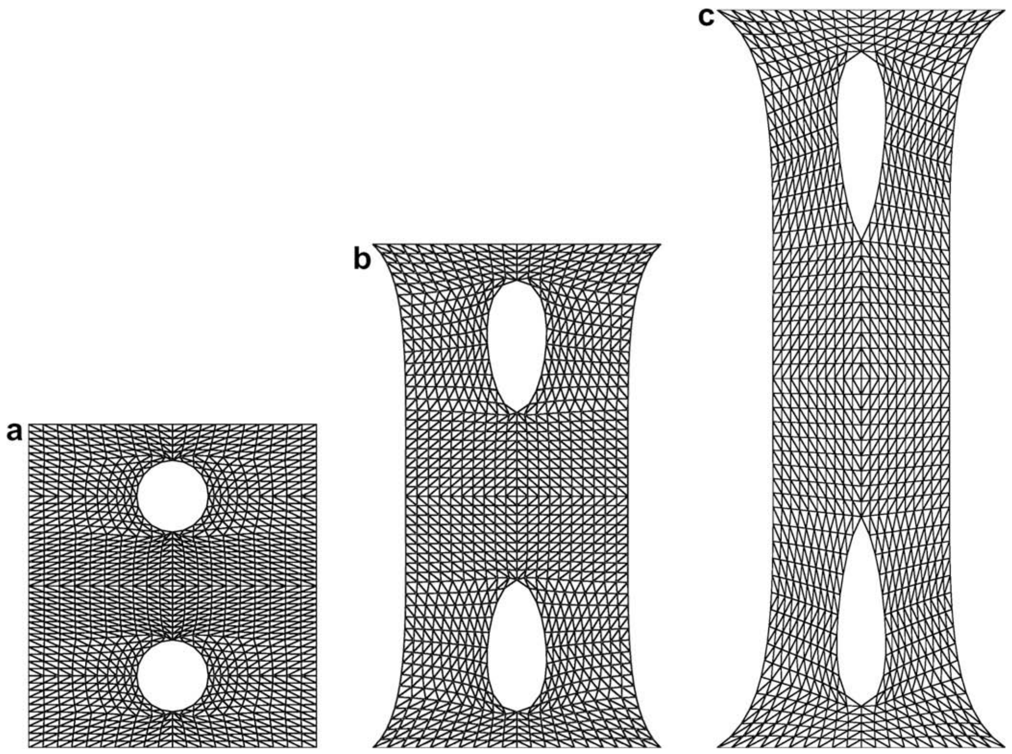

Fig. 16. Deformations of the mesh U1 with a uniform tessellation of 10×48 one-dimensional elements in the radial and axial direction, respectively, at times (a) $t = 0.0$, (b) $t = 5.0$ and (c) $t = 11.5$. The dimensionless parameters are $De = 1$, $Re = 0$, $Ca = 10$, $\varepsilon_{PTT} = 0.03$, $\beta = 0.01$, $\Lambda = 2.25$, $H_0 = 9$, $R_{b1} = R_{b2} = 1$, $h_1 = 2$ and $h_2 = 7$.

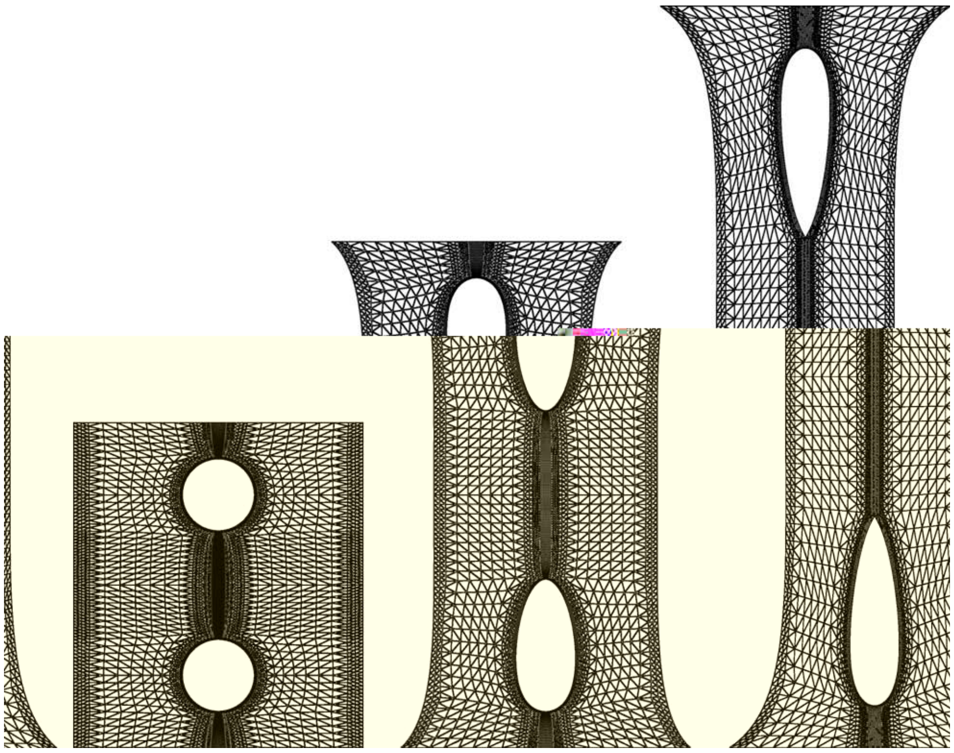

Fig. 17. Deformation of the mesh M1 for the flow and geometric parameters and times of Fig. 15 with triple local refinement around the bubble surfaces, $\xi \leqslant R_{c,o}/20$, $R_{c,o}/10$, $R_{c,o}/5$, and single along the liquid one, $0.9R_{c,o} \leqslant \xi \leqslant R_{c,o}$.

two different times, $t = 5$ and $t = 11.5$, for the same mesh, M1. From these figures we can clearly observe that the mesh remains very uniform around the bubble and that its skewness is reasonably restricted. In an even closer view around the bottom of the upper bubble (Fig. 18(c3)), one can observe the inception of a cusp at the bubble pole. This is similar to the well documented inverted teardrop shape that a bubble assumes when it rises in a viscoelastic liquid [61].

Fig. 19 shows the evolution of the pressure, P, and components of the stress tensor, τ_p, with time at the points on the bubble surface that are most critical for determining the accuracy of the calculations, i.e. the poles and the equatorial plane of the bubble, for four different meshes: U1, U2, M1 and M2. The model parameters are those already mentioned before. The results are given for one of the two bubbles, the upper one, because of the symmetry of the problem. Fig. 19(a) shows the pressure, P, and the stress tensors, $\tau_{\theta\theta,p}$, $\tau_{rz,p}$ and $\tau_{zz,p}$ at the north pole of this bubble. We can see that convergence of the results depends critically on the discretization on the free surface. Indeed meshes U2 and M1, where the discretization of the free surface is exactly the same, give identical and converged results, while using the finer mesh M2 does not offer higher accuracy at least in the scale shown in these figures, although computing with this mesh is more expensive. This means that using either a coarser mesh than M1 away from the bubbles, or a finer mesh than M1 on the bubbles surfaces does not affect the solution. In contrast, when the coarsest uniform mesh U1 is used, the solution shows a continuously increasing deviation from the solutions obtained with the other three meshes. Similar is the conclusion by examining the evolution of the solution for the pressure P and the stress components $\tau_{zz,p}$, $\tau_{rz,p}$ and $\tau_{rr,p}$, at the south pole of the bubble (Fig. 19(b)). While the results for the denser meshes (M1, M2 and U2) are in very good agreement, except close to the end of the calculations, the solution for the coarser mesh U1 differs from early on. The elements on the free surface of this mesh, given in Fig. 16, are not enough to accurately describe its large deformations at longer times. In other words, at short times where no significant deformation of the bubble occurs, the solution with mesh U1 is in agreement with the solutions with the finer meshes. As time increases and the deformations become significant, the solution with this mesh deviates and when the bubble forms a cusp at the south pole only the finest mesh on its surface remains accurate. In contrast with the upper and the lower pole of the bubble, the solution at the equatorial plane (Fig. 19(c)) seems not to be affected by the mesh as much. Indeed, the deformations at the equatorial plane are smoother and even the coarsest mesh is sufficient to describe accurately the free surface there.

Fig. 20 presents contour plots of the velocities, the pressure and the stress components at $t = 5.0$, obtained with mesh M1. We observe that all the contour lines are smooth and no 'wiggles' appear anywhere. We can also observe that all profiles are symmetric with respect to the instantaneous mid-plane of the filament, except of course for the velocity U_z. This velocity component takes its highest values at the upper disk, which is the one that causes the entire motion and deformation, while it takes zero values at the lower disk which remains motionless.

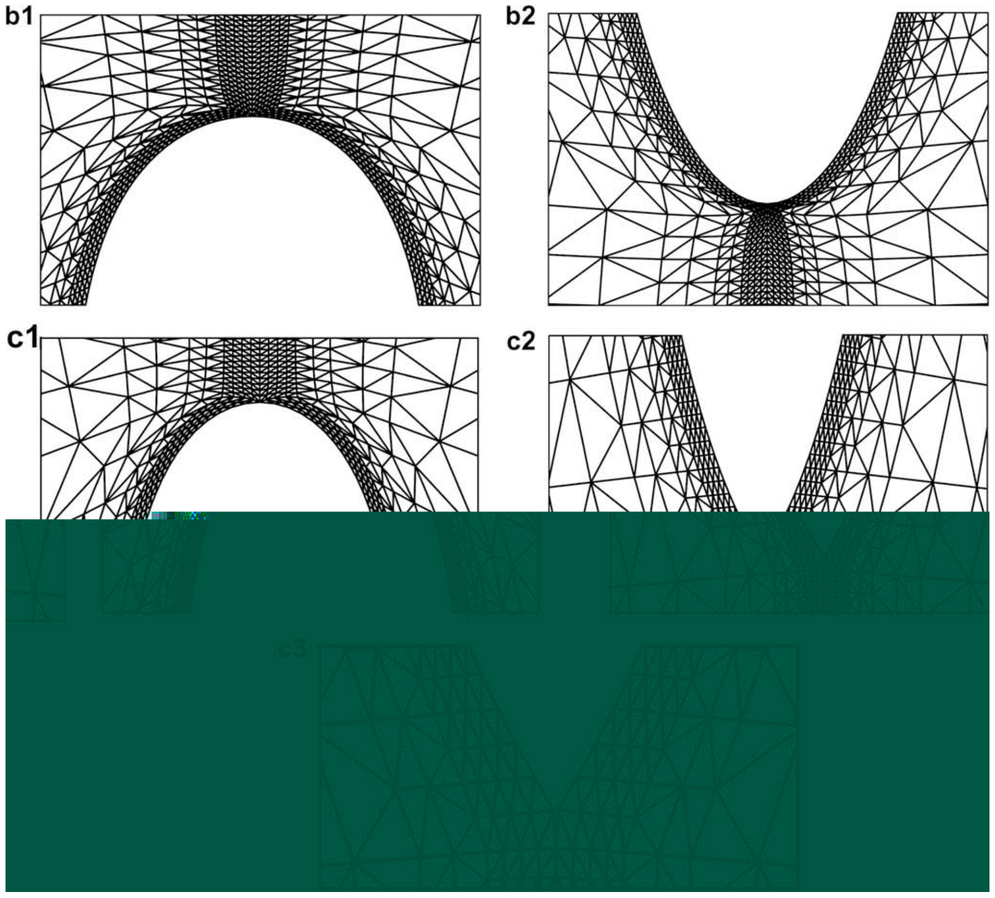

Fig. 18. Details of the mesh of Fig. 17(b) (b1, b2) and Fig. 16(c) (c1, c2, c3), around the top and the bottom of the upper bubble in the left and the right part of the figure, respectively.

5.2. Interacting bubbles in an infinite domain

Having generated the entire mesh by uniting the three parts of the domain, it is easier to use cylindrical coordinates to write the governing equations in each node. In this way, the following two tests have been performed.

5.2.1. Calculation of the eigenvalues of the system

Perhaps the most demanding and strict test to determine the accuracy of the simulations for this problem is to examine the convergence with mesh refinement of the eigenvalues of the corresponding free boundary problem. This is readily achieved by assuming that initially no flow exists and the bubbles are spherical due to capillarity and, then, subjecting all the flow variables including the bubble shapes to an infinitesimal disturbance. In addition to convergence with mesh refinement, we can compare our predictions with the analytically obtained eigenvalues for an isolated bubble surrounded by a viscous liquid, which have been reported by Miller and Scriven [62]. To this end, we only need to position the two bubbles as far apart as allowed by our current hardware/software configuration.

More specifically, the normal modes of the system are computed by assuming that all variables are split into their base values and a small disturbance.

$$\begin{bmatrix} \mathbf{u}(r,z,\theta,t) \\ P(r,z,\theta,t) \\ P_{gi}(r,z,\theta,t) \\ \mathbf{x}(r,z,\theta,t) \end{bmatrix} = \begin{bmatrix} \mathbf{u}_b(r,z) \\ P_b(r,z) \\ P_{gib}(r,z) \\ \mathbf{x}_b(r,z) \end{bmatrix} + \delta \begin{bmatrix} \mathbf{u}_p(r,z) \\ P_p(r,z) \\ P_{gip}(r,z) \\ \mathbf{x}_p(r,z) \end{bmatrix} e^{ct}, \tag{43}$$

where the new subscripts b and p indicate the equilibrium (base) state and the perturbed one, respectively, $\delta \ll 1$ is the amplitude of the infinitesimal disturbance and $\mathbf{x}$ denotes the position vector of the mesh nodes. As noted in Carvalho and Scriven [23], by invoking the domain perturbation approach only the positions of nodes on the free surfaces need to be perturbed as part of the overall disturbance. This is achieved by setting the domain perturbation to

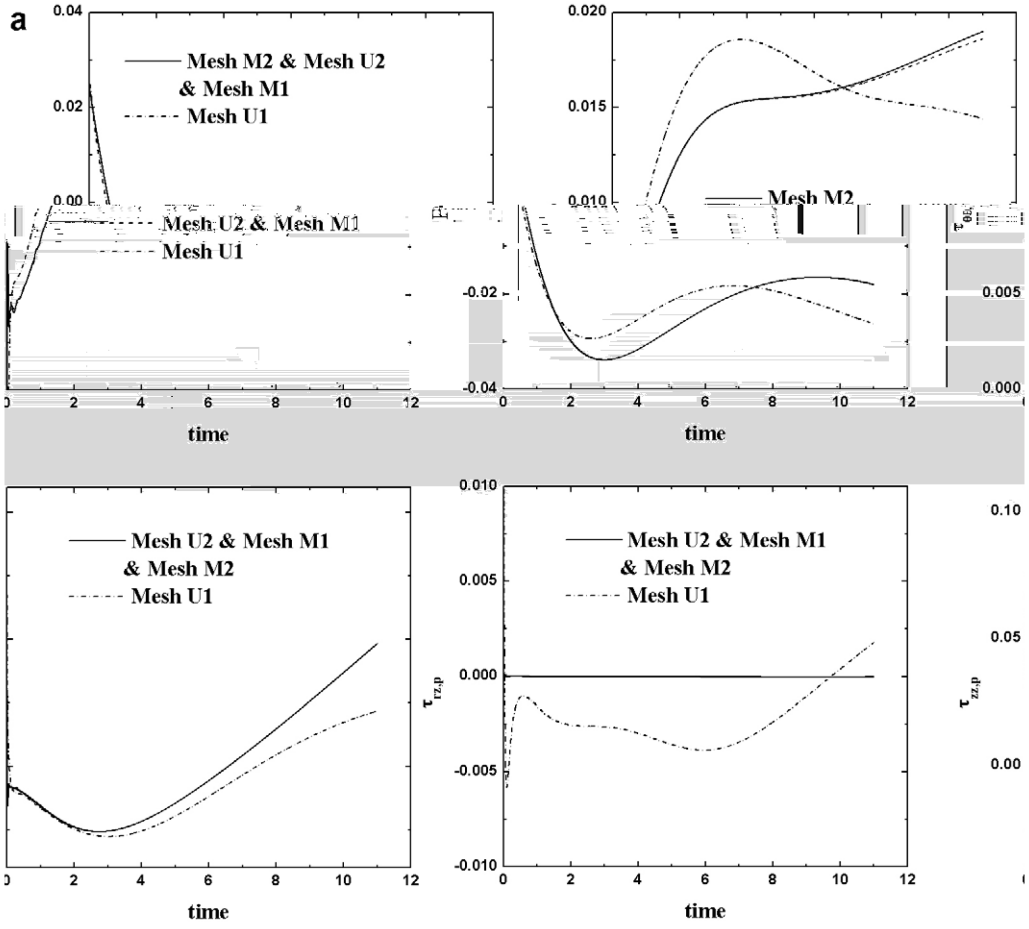

Fig. 19. Evolution with time of (a) P, $\tau_{\theta\theta,p}$, $\tau_{rz,p}$, $\tau_{zz,p}$ for the upper pole of the upper bubble, (b) P, $\tau_{zz,p}$, $\tau_{rz,p}$, $\tau_{rr,p}$ of the lower pole of the upper bubble and (c) u_z, u_r, $\tau_{rr,p}$, $\tau_{rz,p}$ of a point near the equatorial plane of the upper bubble for the indicated meshes. The dimensionless parameters are those in Fig. 16.

$$\mathbf{x}_p = H^{(0)}(\mathbf{x}_b)h'\mathbf{n}, \tag{44}$$

where $\mathbf{n}$ is the unit normal to the unperturbed boundary, h' is a scalar function related to the amplitude of the perturbation for the mesh nodes and $H^{(0)}$ is the Heaviside function

$$H^{(0)}(\mathbf{x}_b) = \begin{cases} 1, & \mathbf{x}_b \in \Gamma \\ 0, & \mathbf{x}_b \in \Omega \end{cases}. \tag{45}$$

Thus $H^{(0)}(\mathbf{x}_b)$ vanishes inside the unperturbed liquid domain, whereas it is equal to one on its boundaries, allowing only the nodes at the moving boundary to be displaced as determined by h'.

Substituting expressions (43) into the governing equations, including the kinematic Eq. (13) and the equation of state that describes the pressure inside the bubble (11), and neglecting terms of order higher than linear in the perturbation parameter, δ, we obtain a generalized eigenvalue problem of the form

$$\mathbf{JY} = c\mathbf{MY}, \tag{46}$$

where $\mathbf{J}$ is the Jacobian matrix, $\mathbf{M}$ is the mass matrix, c are the eigenvalues and $\mathbf{Y}$ the corresponding eigenvectors. In order to solve the eigenvalue problem, we used the Arnoldi method as it is implemented in the Arpack library [63]. This method is capable of iteratively computing a relatively small fraction of the entire spectrum of eigenvalues and in particular those that have the largest magnitude. Since we are interested in enhancing convergence to a specific portion of the spectrum and in order to limit the fraction of the eigenvalues to be computed each time, we used the shift-and-invert transformation

$$(\mathbf{J} - \lambda\mathbf{M})^{-1}\mathbf{MY} = v\mathbf{Y}, \quad \text{where } v = \frac{1}{c-\lambda}. \tag{47}$$

This transformation is efficient because not only it turns the generalized eigenvalue problem into a simple one but also it computes the eigenvalues that are closer to the user-provided complex number λ, since the eigenvalues v of Eq. (47) that

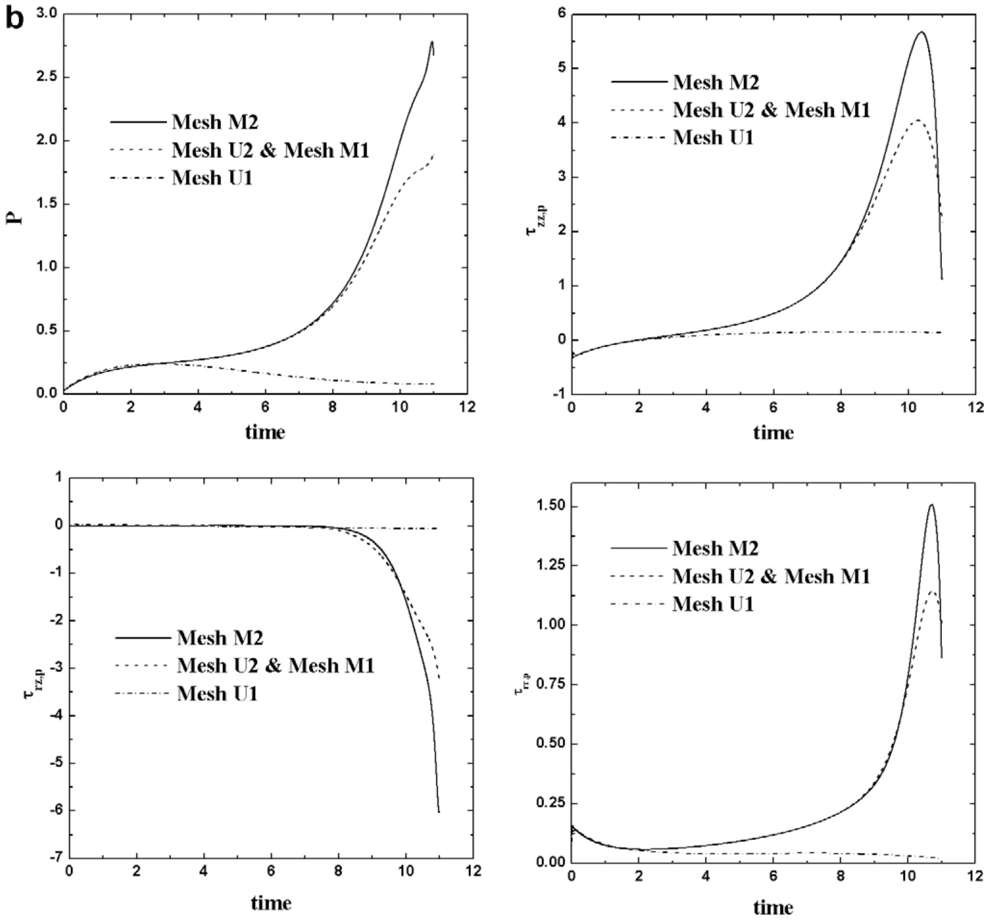

Fig. 19 (*continued*)

are largest in magnitude correspond to the eigenvalues c of the original problem that are nearest to the shift constant λ in absolute value. The accuracy of the converged eigenpairs is checked independently by evaluating a posteriori the residual $|\mathbf{J}\mathbf{Y}_n - c\mathbf{M}\mathbf{Y}_n|$.

The computed eigenvalues are compared with those that arise from the analytical solution of Miller and Scriven [62] for individual bubbles which gives the eigenvalues c by the implicit expression

$$\frac{c^{*2}}{c^2} = \frac{l+2}{\omega_o^{*2}}\left[\frac{(2l+1)\omega_o^{*2}R_{bi}^{*2} - 2(l-1)(l+1)(2l+1-\omega_o^* R_{bi}^* Q^H_{l+\frac{1}{2}})}{2l+1-\omega_o^* R_{bi}^* Q^H_{l+\frac{1}{2}} + \omega_o^{*2}R_{bi}^{*2}/2}\right] - 1, \quad i = A, B, \quad l = 2, 3, \ldots, \qquad (48)$$

where $\omega_o^* = (c^*\rho^*/\mu_s^*)^{1/2}$ and $c^* = \left(\frac{\sigma^*(l+1)(l-1)(l+2)}{R_{bi}^{*3}\rho^*}\right)^{1/2}$ is the frequency of oscillation, if the liquid is assumed to be inviscid and the gas in the bubble does not contribute to the dynamics of the problem, as we have already assumed. In these expressions, l corresponds to the index of the Legendre polynomial characterizing the shape of the bubble. The analytical solution does not apply for volume oscillations of the bubble ($l = 0$) or bubble translation ($l = 1$). The real part, c_R, of c is the amplification or decay factor, the imaginary part, c_I, is the angular frequency and $Q^H_{l+\frac{1}{2}} \equiv \frac{H^{(1)}_{l+\frac{3}{2}}(\omega_o^* R_{bi}^*)}{H^{(1)}_{l+\frac{1}{2}}(\omega_o^* R_{bi}^*)}$; $H^{(1)}_{l+\frac{1}{2}}$, $H^{(1)}_{l+\frac{3}{2}}$ are the half-integral-order Hankel functions of first kind. The oscillation frequency and decay factor are obtained numerically assuming the value of the Ohnesorge number and properties of water at 25 °C and atmospheric pressure by solving Eq. (48) using standard software.

The eigenvalue problem required 12–24 h to complete, depending on the flow parameters, before the local refinement technique was introduced, while it required only 1/2–1 h following this procedure. In addition, the maximum number of nodes that could be used before the local refinement with our current hardware, was up to 150 along each bubble surface due to the high computational cost, and the requirement of a gradual coarsening of the mesh away from the bubbles resulting in a total of 36,720 triangles. On the contrary up to 2250 nodes along each free surface can be used after the local refinement increasing the accuracy of our computations, as discussed below. Table 2 presents the characteristics of the meshes

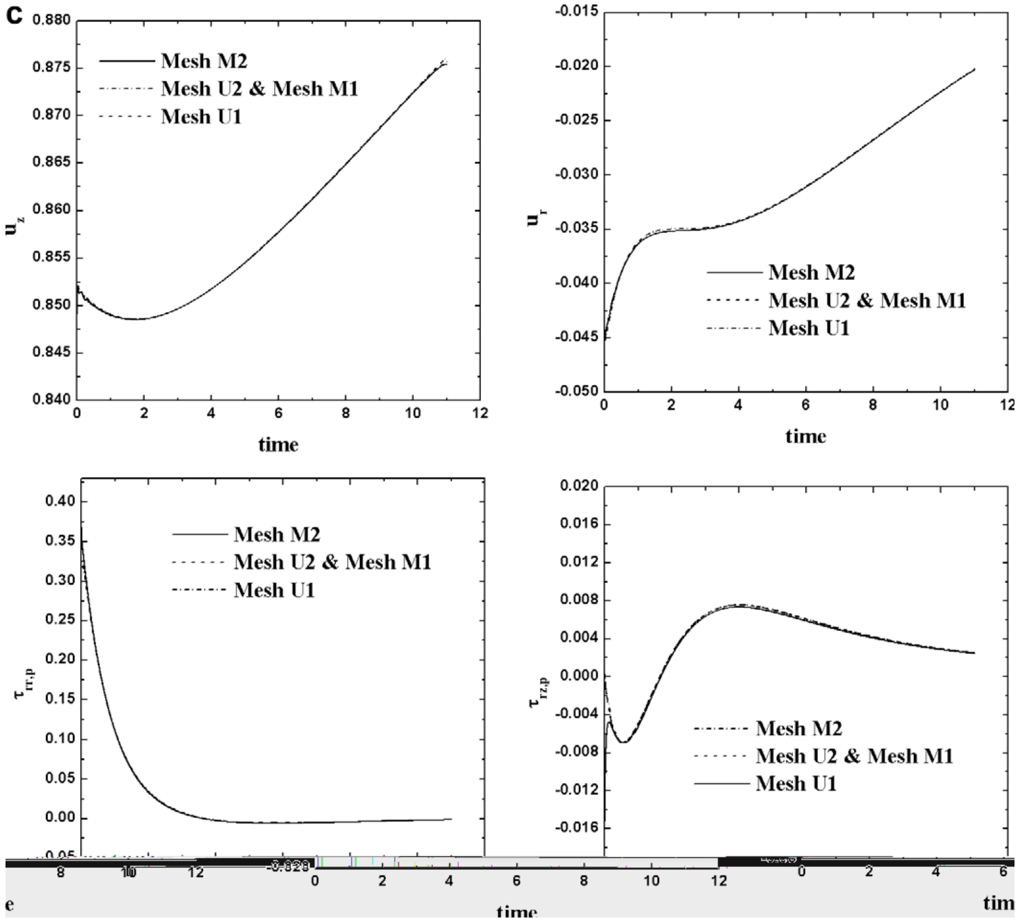

Fig. 19 (*continued*)

that we used. Only in the M family of meshes three levels of local refinement have been introduced near the bubble surfaces as detailed in the table. Special care has been paid so that the aspect ratio of the elements near the bubbles does not deviate a lot from unity. This can be seen in the minimum values of their sides in the r- and z-directions (in a cylindrical coordinate system). The reduction of both these values near the bubbles through local mesh refinement is noteworthy.

Table 3 summarizes the computed eigenvalues for bubbles of equal size, $R_{bB} = 1$, at a relatively large distance between their centers $D = 17$, when the far-field boundary is taken at $R_\infty = 30$. We have verified that neither one of the two latter parameter values affect the computed eigenvalues by increasing the distance to $D = 24$ and the boundary location to $R_\infty = 45$. Results with one relatively uniform mesh, U3, and three meshes with local refinement, M3–M5, are given, for various values of Oh^{-1}, which plays the role of a Reynolds number for this problem. In the same table the semi-analytically calculated eigenvalues using Eq. (48) for individual bubbles are also given. Clearly the very expensive calculations with the U3 mesh result in the least accurate values, whereas even the coarsest of the M family of meshes, M3, results in values closer to the semi-analytical ones for all values of Oh^{-1} numbers and all modes. Increasing the nodes on the free surface, the numerically computed eigenvalues tend monotonically to the semi-analytical ones. Even the higher eigenmodes, that involve larger contributions from higher Legendre polynomials and larger interfacial deformations, which should have been computed less accurately, are in very good agreement with the theoretical eigenvalues. On the other hand, the higher the Oh^{-1} number is, the lower the accuracy of the computed eigenvalues. This should have been anticipated since increasing Oh^{-1}, increases the inertial over the viscous forces producing a sharper boundary layer around each bubble, which requires an even finer mesh to capture accurately the eigensolution. We also performed tests for the aspect ratio of the elements around the free surfaces. It is found that when all the elements around the free surface have the same or quite the same aspect ratio higher accuracy is achieved. This remark may explain the variation between eigenvalues resulting from the analytical expression and numerically computed ones. In our mesh the aspect ratio of the elements located between the two bubbles is smaller than unity whereas those located at their other poles are larger than unity.

Fig. 21 shows the relative error of each numerically computed eigenvalue with respect to each analytical counterpart in logarithmic scales. Clearly for all modes, there is a linear decrease of the error as the number of surface elements increase. On

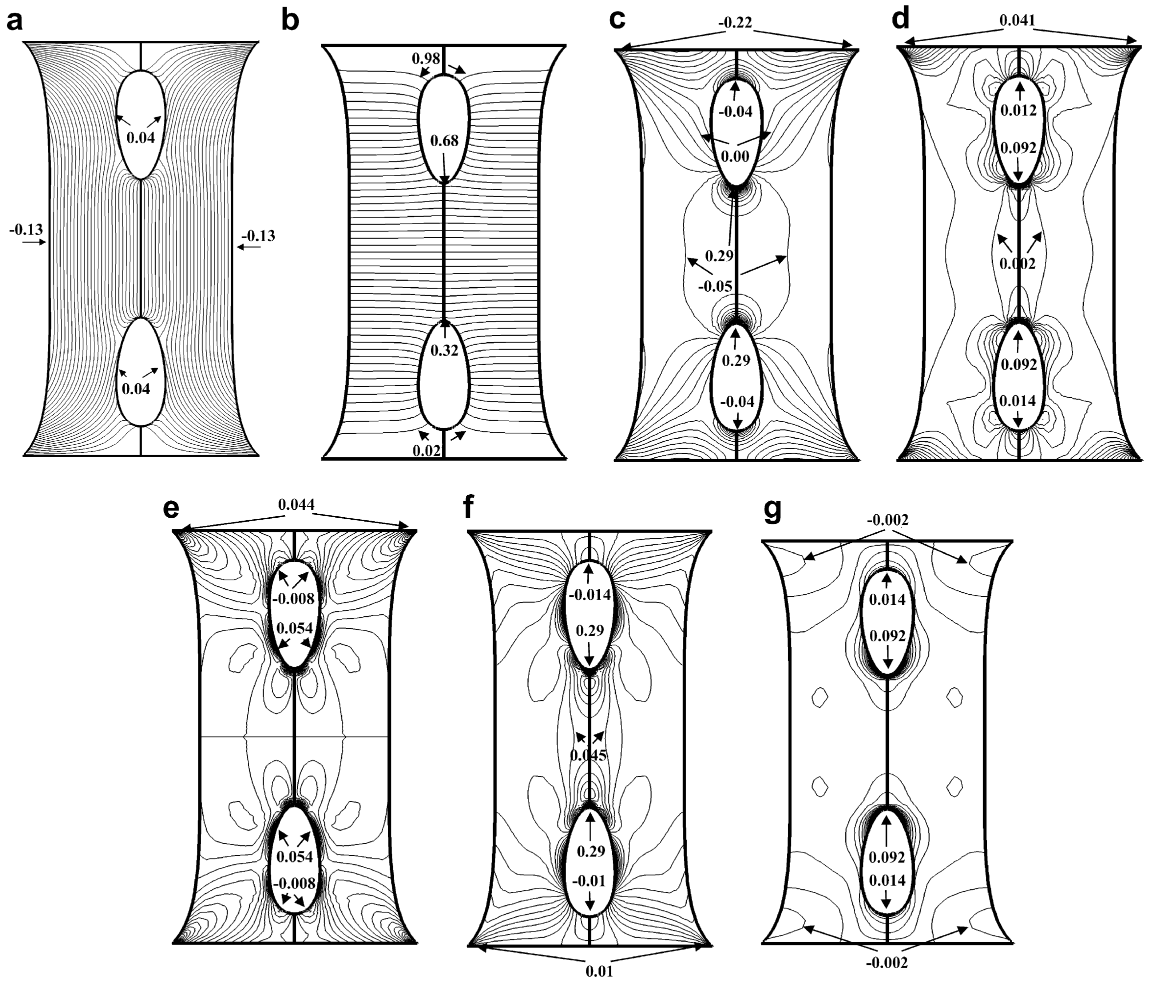

Fig. 20. Contours of (a) u_r, (b) u_z, (c) P, (d) $\tau_{rr,p}$, (e) $\tau_{rz,p}$, (f) $\tau_{zz,p}$ and (g) $\tau_{\theta\theta,p}$ at $t = 5.0$ for the dimensionless parameters of Fig. 12.

the contrary, the error does not depend in such a simple manner on the number of total nodes or elements, verifying that the number of surface elements is the most crucial factor for this computation. It is noteworthy that in all cases we have computed the inner product of the eigenvector for the modes $l = 2, 3, 4$ and 5 with Legendre polynomials of various degrees and found that each eigenvector contained only the Legendre polynomial of degree 2, 3, 4 and 5, correspondingly, testifying once more to the accuracy of the present calculations.

5.2.2. Transient simulations

Fig. 22, presents snapshots of the mesh around the bubbles at three different times: (a) $t = 0.28$, (b) $t = 0.4$ and (c) $t = 0.66$, and for two different meshes. At the left column of snapshots in Fig. 21 the mesh is finer near the bubble surfaces and becomes coarser slowly away from them (mesh U4 in Table 2). The total number of triangular elements is 25,632, which corresponds to 220,163 unknowns, while the number of nodes along each bubble surface is only 145. At the right column of snapshots in Fig. 22 the mesh in addition contains a three-level refinement around the bubbles (mesh M6 in Table 2). The total number of triangular elements is smaller 9792 which corresponds to less 85,991 unknowns, whereas the number of nodes along each bubble interface is much higher 385. The two bubbles are equal, $R_{b2} = 1$, their dimensionless initial distance is set to $D = 2.8$, which allows us to use a generally less refined mesh compared to that in the eigenvalue computations, and the outer boundary is $R_\infty = 30$. Moreover, the dimensionless static pressure, P_s, is equal to 100. As initial disturbance, a step change in pressure at the far-field is applied, with $\varepsilon = 1$ increasing the far-field pressure by a factor of 2. The two bubbles undergo volume oscillations and simultaneously approach each other with time, as expected, since they oscillate in phase and attractive forces develop [36,37,41]. From Fig. 22 we can see that when local refinement is used, the mesh remains dense around the two bubbles until the end of the simulations. On the other hand when no local refinement is used and as time passes, the elements are deformed near the poles of the bubbles, with quite large (quite small) aspect ratios between the two bubbles (in the other direction). If the simulations were allowed to continue further, remeshing would have been required.

Table 2
Characteristics of the meshes used for the bubbles in an acoustic field problem. Local refinement has been used for the M family of meshes only. In meshes M3–M5 a three-level local refinement has been used around each bubble surface at the following distances in the computational domain: $R_{bi} \leqslant \xi \leqslant 1.008R_{bi}$, $1.008R_{bi} \leqslant \xi \leqslant 1.026R_{bi}$, $1.026R_{bi} \leqslant \xi \leqslant 1.036R_{bi}$, while in meshes M6–M7 the three levels of refinement were introduced at: $R_{bi} \leqslant \xi \leqslant 1.014R_{bi}$, $1.014R_{bi} \leqslant \xi \leqslant 1.029R_{bi}$, $1.029R_{bi} \leqslant \xi \leqslant 1.039R_{bi}$.

Mesh	Part A or B (radial, azimouthal) elements before refinement	Part C (radial, azimouthal) elements before refinement	Number of triangles after refinement (if used)	Number of nodes on free surface after refinement	$\Delta r_{\min}$ around the free surface after refinement	$\Delta z_{\min}$ around the free surface after refinement
U3	(120,72)	(10,108)	36,720	145	0.044	0.004
M3	(37,40)	(10,60)	15,360	641	0.01	0.002
M4	(37,80)	(10,120)	30,720	1281	0.005	0.002
M5	(37,140)	(10,210)	53,760	2241	0.003	0.002
U4	(14,72)	(100,108)	25,632	145	0.044	0.063
M6	(16,24)	(50,36)	9792	385	0.016	0.004
M7	(16,40)	(50,60)	16,320	641	0.010	0.004

Table 3
Comparison of eigenvalues derived either semi-analytically [62] or numerically with the indicated meshes, for an isolated bubble at various Oh^{-1} numbers ($D = 17$, $R_\infty = 30$).

l	Miller and Scriven	M3	M4	M5	U3
$Oh^{-1}(= Re) = 5$					
2	1.5252 ± 1.9697*i*	1.5344 ± 1.9918*i*	1.5298 ± 1.9807*i*	1.5278 ± 1.9760*i*	1.5648 ± 2.0674*i*
3	2.7030 ± 3.7877*i*	2.7092 ± 3.8040*i*	2.7061 ± 3.7958*i*	2.7047 ± 3.7923*i*	2.7301 ± 3.8599*i*
4	4.0400 ± 5.6620*i*	4.0452 ± 5.6754*i*	4.0426 ± 5.6687*i*	4.0415 ± 5.6658*i*	4.0625 ± 5.7214*i*
5	5.5513 ± 7.6212*i*	5.5560 ± 7.6326*i*	5.5536 ± 7.6270*i*	5.5526 ± 7.6245*i*	5.5712 ± 7.6723*i*
$Oh^{-1}(= Re) = 20$					
2	0.7252 ± 3.2205*i*	0.7288 ± 3.3094*i*	0.7270 ± 3.2652*i*	0.7262 ± 3.2461*i*	0.7389 ± 3.5996*i*
3	1.2677 ± 5.8998*i*	1.2704 ± 5.9661*i*	1.2690 ± 5.9330*i*	1.2685 ± 5.9188*i*	1.2786 ± 6.1887*i*
4	1.9367 ± 8.8197*i*	1.9392 ± 8.8755*i*	1.9379 ± 8.8476*i*	1.9374 ± 8.8356*i*	1.9468 ± 9.0642*i*
5	2.7302 ± 11.9906*i*	2.7328 ± 12.0400*i*	2.7314 ± 12.0153*i*	2.7308 ± 12.0047*i*	2.7401 ± 12.2070*i*
$Oh^{-1}(= Re) = 40$					
2	0.4041 ± 3.3754*i*	0.4074 ± 3.6618*i*	0.4061 ± 3.5214*i*	0.4053 ± 3.4596*i*	0.4145 ± 4.5098*i*
3	0.7074 ± 6.1694*i*	0.7102 ± 6.3867*i*	0.7088 ± 6.2789*i*	0.7082 ± 6.2322*i*	0.7173 ± 7.0828*i*
4	1.0857 ± 9.2425*i*	1.0885 ± 9.4262*i*	1.0870 ± 9.3347*i*	1.0864 ± 9.2953*i*	1.0957 ± 10.0309*i*
5	1.5384 ± 12.6049*i*	1.5414 ± 12.7676*i*	1.5398 ± 12.6864*i*	1.5392 ± 12.6515*i*	1.5487 ± 13.3088*i*

One way to examine closer the accuracy of the present dynamic computations, it is to decompose the bubble surface into Legendre modes. These are computed with respect to a spherical coordinate system with its origin located at the instantaneous center of mass of each bubble. The Legendre coefficients of each mode are computed as follows:

$$C_{il} = \int_0^\pi F_{si}(\theta)P_l(\theta)\sin(\theta)d\theta, \quad i = A, B;\ l = 0, 1, 2, 3\ldots, \tag{49}$$

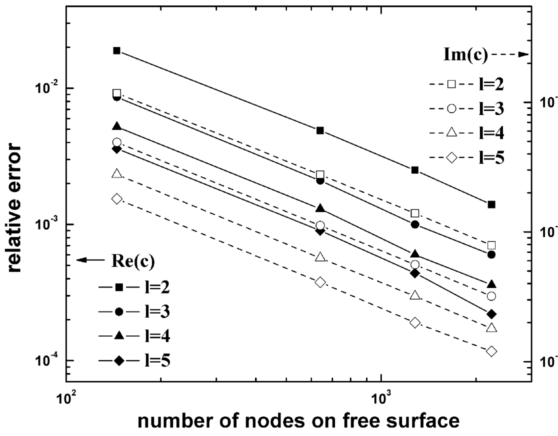

Fig. 21. Relative error of each numerically computed eigenvalue with respect to its semi-analytical counterpart for meshes U3, M3, M4 and M5 and for $Oh^{-1} = 20$.

where F_{si} is position of the interface of the i bubble. Then, it may be seen that the dominant modes for this case are those that correspond to the Legendre polynomials of degree zero and two. Fig. 23 shows the time evolution of the coefficient, C_0 (Fig. 23(a)), of the zeroth degree Legendre polynomial, P_0, that is associated with volume oscillations, and the coefficient, C_2 (Fig. 23(b)), of the second degree Legendre polynomial, P_2, for the same parameters that we mentioned earlier. Positive values of the coefficient C_2 correspond to prolate bubble shapes (elongated along the axis of symmetry) whereas negative values to oblate bubble shapes (flattened at the bubble poles). Hence the bubble initially oscillates retaining a prolate shape, whereas at later times it attains both prolate and oblate shapes and finally only oblate shapes, since then the bubbles have approached and flattened each other considerably. These coefficients are compared for three different meshes, U4, M6 and M7; where U4 stands for a slowly graded mesh (without local refinement), while in meshes M6 and M7, three levels of refinement are used around the bubble surfaces. As can be seen in Fig. 23, both coefficients calculated with the U4 mesh deviate with time from those calculated with the other two meshes indicating that a very fine mesh around the bubble surfaces is indeed required in order to solve accurately this problem for longer times.

In Table 4, we give the relative error in pressure, the most sensitive of the variables computed in Newtonian fluids, at three positions on the surface of the left bubble (near its two poles and its equatorial plane) for meshes U4 and M6 with respect to the values obtained with mesh M7, at three different times. As previously, the finer the mesh around the two bubbles is the more accurate the calculated pressure is. More precisely, the relative error in pressure for the mesh U3 at $t = 0.28$ is $\sim$ 1.3% and increases up to $\sim$8.5% at $t = 0.66$ when the bubble has deformed appreciably. On the contrary, the relative error for the mesh M6 starts with $\sim$0.1% at $t = 0.28$ and does not exceed $\sim$0.8% at the end of the simulations. In addition to the larger error, the smoothly varying mesh here has a larger number of elements distributed throughout the domain around the bubbles, which necessitates fewer elements adjacent to the free surfaces in order to keep the computational cost at a reasonable level. This is exactly what the local mesh refinement achieves, more elements on the highly deforming, free surfaces, where they are needed the most for higher accuracy in the calculations, and fewer elements away from them to the point that the total number of elements is smaller. The required computation times on a single node of our quad xeon machine are almost 2 weeks using the U4 mesh as opposed to 2 and 6 days for the M6 and M7 meshes, respectively.

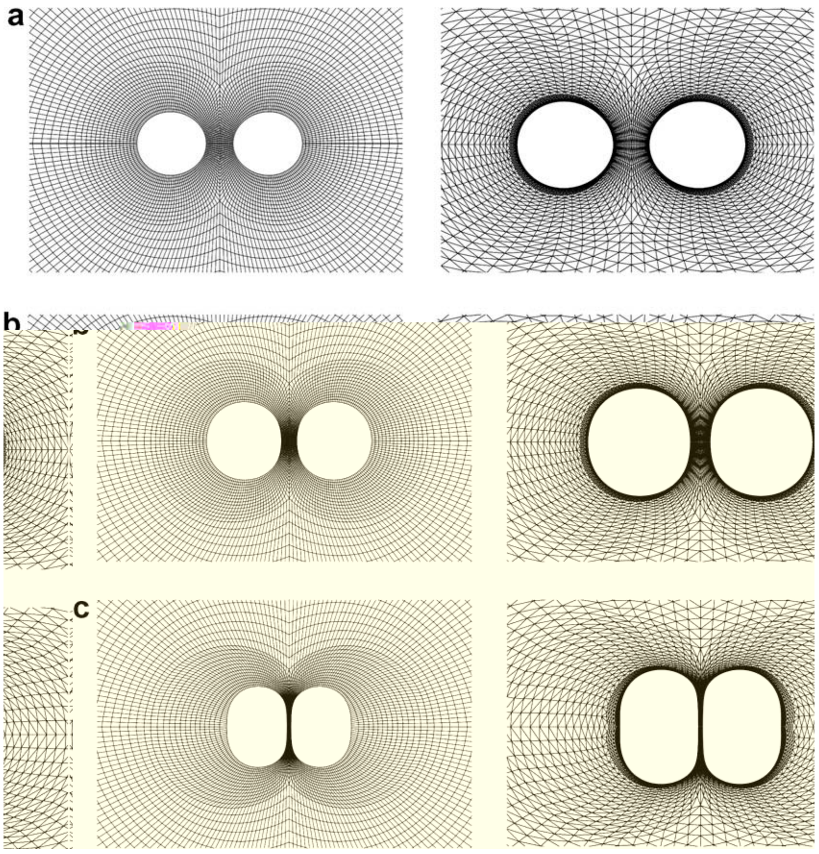

Fig. 22. Snapshots of the mesh around the two bubbles when the uniform mesh, U4, is used (left column) and when the three-level refinement mesh, M6, is used (right column) at (a) $t = 0.28$, (b) $t = 0.4$ and (c) $t = 0.66$. Dimensionless parameters are $R_{bB} = 1$, $Oh^{-1} = 10$, $P_s = 100$, $\varepsilon = 1$, $D = 2.8$ and $R_\infty = 30$. For mesh U4 only rectangular elements are shown.

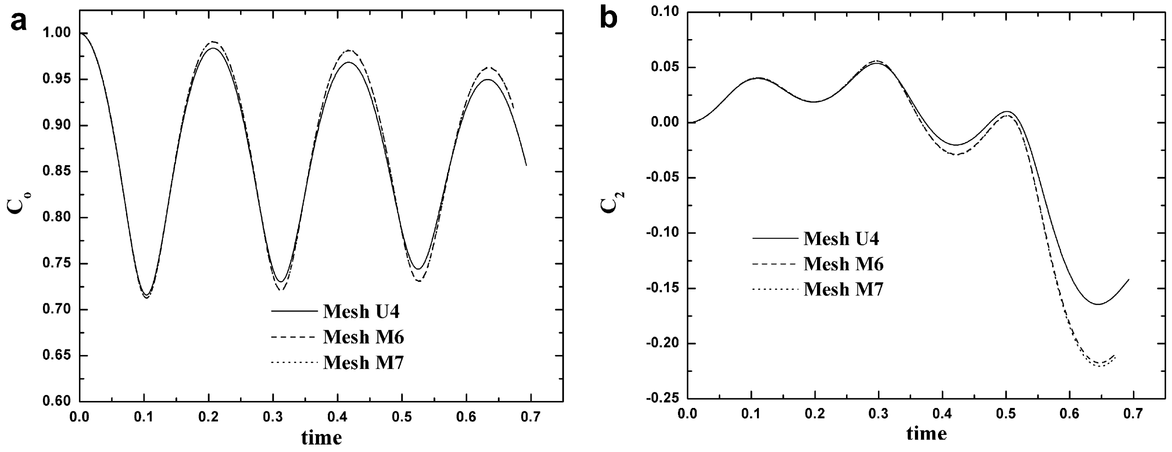

Fig. 23. Time evolution of the coefficients of the Legendre modes (a) P_0 and (b) P_2 for the decomposition of the left bubble interface, for three different meshes, U4, M6 and M7. Dimensionless parameters are those of Fig. 22.

Fig. 24 presents contour plots of both velocity components and the pressure at $t = 0.66$. The results are given in spherical coordinates with center located at the middle of the distance between the initial centroids of the two bubbles. The mesh used is M7 and no remeshing is needed until the end of the simulation. The radial velocity, Fig. 24(a) (upper half), takes the highest and positive values between the two bubbles, while it takes negative values at the rear side of the bubbles. This velocity distribution indicates that the bubbles are contracting and approaching each other at this time instant, squeezing fluid away from the gap between them. If Stokes flow prevailed, the radial velocity should drop like $\sim r^{-1}$ away from the bubbles. In the upper half of Fig. 24(a) we can see that it decreases with the radial distance, but does not exactly follow this rule, since the inverse Ohnesorge number is not small. The azimouthal velocity field, Fig. 24(a) (lower half), is symmetric with respect to $\theta = \pi/2$, because the two bubbles are equal and undergo in-phase oscillations. In agreement with the picture about the radial velocity, the azimouthal velocity takes negative values around the left bubble and positive values around the right bubble around their equators, but changes sign at their poles facing away from each bubble. It also takes zero values at $\theta = 0$, $\theta = \pi$ and $\theta = \pi/2$, as it should. Finally the pressure field, Fig. 24(b), attains radial symmetry not too far from the bubbles, verifying again that our choice to locate the outer boundary at $R_\infty = 30$ is a conservative one for this configuration. As can be seen in Fig. 24 the main changes in the velocity and pressure fields occur around and close to the two bubbles, explaining once more the need for a very fine mesh in that region and not in the area away from them.

Table 4
Relative error of the pressure at three different positions (around the poles and the equatorial plane) on the free surface of the left bubble at three different times: $error = \left|\frac{P|_{z,mesh} - P|_{z,M7}}{P|_{z,M7}}\right|$.

Mesh	z	*error* (%)
t = 0.28		
U3	−2.0	1.31
M6	−2.0	0.15
U3	−1.2	1.32
M6	−1.2	0.15
U3	−0.4	1.29
M6	−0.4	0.16
t = 0.4		
U3	−2.0	4.72
M6	−2.0	0.23
U3	−1.2	4.82
M6	−1.2	0.25
U3	−0.4	4.84
M6	−0.4	0.25
t = 0.66		
Mesh	z	*error* (%)
U3	−1.4	8.41
M6	−1.4	0.72
U3	−0.8	8.34
M6	−0.8	0.72
U3	−0.2	8.48
M6	−0.2	0.72

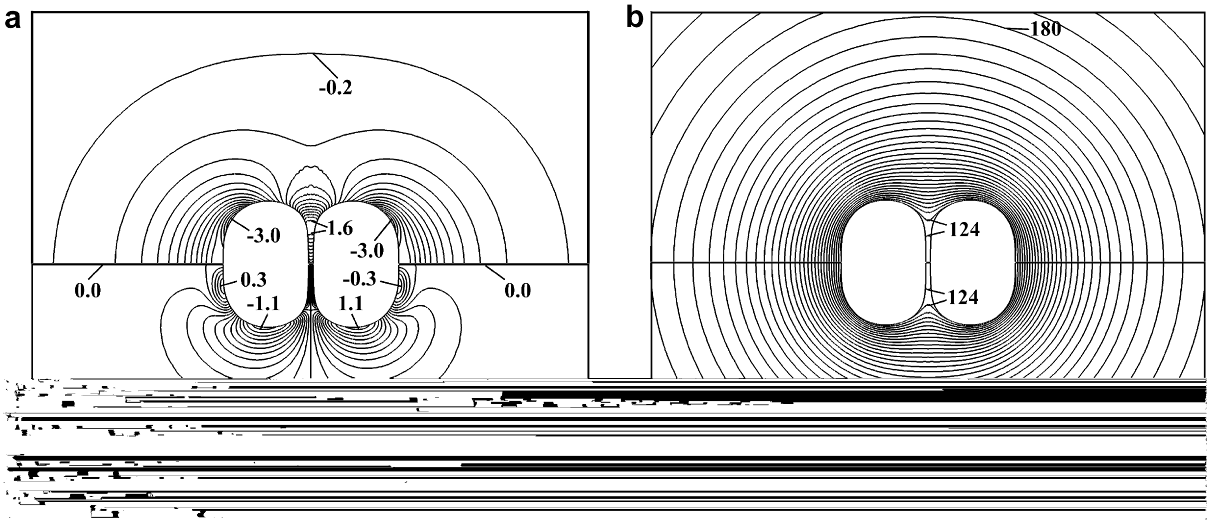

Fig. 24. Contours of (a) u_{rs} (upper half), u_θ (lower half) and (b) P at $t = 0.66$ for the dimensionless parameters of Fig. 22.

6. Conclusions

The elliptic grid generator scheme proposed by Dimakopoulos and Tsamopoulos [13] has been successfully extended to problems where domains with inclusions are involved, the interfaces undergo large deformations and hyperbolic equations must be solved. Depending on the geometry, it has been found that the entire fluid domain can be either mapped onto a single computational domain or it must be divided into subdomains in order to generate the most suitable mesh. The initial mesh is generated by continuation techniques. The fluid problem is always solved in the entire domain and does not require restructuring and interpolation of the variables in a new mesh in order that the simulations retain their high accuracy. For optimizing the quality of the mesh, constraints (e.g. generalized node distributions) along specific surfaces of the geometry were imposed, alleviating in this way the stiffening of the scheme when exponential attractive terms are used in the elliptic equations [13,17]. Another improvement that was incorporated herein is a local refinement algorithm, for increasing the resolution of the mesh along the free surfaces while keeping the computation cost as a low as possible.

The proposed schemes were tested in two highly demanding problems with free surfaces and complex geometries: (a) bubble growth in viscoelastic filaments undergoing stretching where a single domain with multiple mappings suffice and (b) interacting bubbles in a viscous medium, where splitting into subdomains and multiple mappings are necessary. In our first application the mesh is generated by enforcing the node distribution along lines passing through the singular points of the domain. In essence, these lines play the role of internal pseudo-boundaries, but the domain is not actually decomposed for generating the mesh. This approach may not always work as demonstrated by a couple of examples in our second application making the second method indispensable. This is based on decomposition of the physical domain to subdomains defined in local, even different, but most appropriate for each subdomain coordinate systems, while the flow equations are written in the entire domain. This method is the most powerful and general and we believe that it can be applied in any 3D problem. In both cases, we managed to reduce the memory requirements up to 30% and the computational time up to 80% by applying the local refinement method around the deforming interfaces for achieving accuracy similar with or higher than that of the finest mesh without local refinement. Despite the large deformation of the fluid volume, the accuracy of the calculations remained high throughout the simulations, and no remeshing was performed. Although the boundaries of the physical domain have actually moved a lot, the mesh generated by our quasi-elliptic method in the time-independent computational domain does not need to be readjusted. This is important in order to decrease numerically induced instabilities, numerical diffusion and perform numerical simulations even with viscoelastic fluids.

Currently, we are working on the extension of this methodology to 3D domains. In this case, for example, the motion and deformation of multiple bubbles away from the axis of symmetry can be computed and the applications studied here or others, such as the dynamics of emulsions in flow, growth of cavitating bubbles behind hydrofoils can be accurately simulated. Certainly then, this methodology would have a greater impact.

Acknowledgment

This work was partially supported under the PENED 2001 program of the General Secretariat of Research and Technology of Greece (Program No. PENED 2001-01ED558).

References

[1] A. Khamayseh, A. Kuprat, C.W. Mastin, Boundary orthogonality in elliptic grid generation, in: J.F. Thompson, B.K. Soni, N.P. Weatherhill (Eds.), Handbook of Grid Generation, CRC Press, Boca Raton, FL, 1999.

[2] P.H. Saksono, D. Peric, On finite element modeling of surface tension. Variational formulation and applications – part II: dynamic problems, Comput. Mech. 38 (2006) 251–263.

[3] C.W. Hirt, B.D. Nichols, Volume of Fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201–225.

[4] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys. 100 (1992) 25–37.

[5] M. Zacharioudaki, C. Kouris, Y. Dimakopoulos, J. Tsamopoulos, A direct comparison between volume and surface tracking methods with a boundary-fitted coordinate transformation and third-order upwinding, J. Comput. Phys. 227 (2007) 1428–1469.

[6] R.E. Smith, Algebraic grid generation, Appl. Math. Comput. 10–11 (1982) 137–170.

[7] J.F. Thompson, B.K. Soni, N.P. Weatherhill (Eds.), Handbook of Grid Generation, CRC Press, Boca Raton, FL, 1999.

[8] J.F. Thompson, Z.U.A. Warsi, W.C. Mastin, Boundary-fitted coordinate systems for numerical solutions partial differential equations – a review, J. Comput. Phys. 47 (1982) 1–108.

[9] G. Ryskin, L.G. Leal, Orthogonal mapping, J. Comput. Phys. 50 (1983) 71–100.

[10] K.N. Christodoulou, L.E. Scriven, Discretization of free surface flows and another moving boundary problems, J. Comput. Phys. 99 (1992) 39–55.

[11] K. Tsiveriotis, R.A. Brown, Boundary-conforming mapping applied to computations of highly deformed solidification interface, Int. J. Numer. Meth. Fluids 14 (1992) 981–1003.

[12] K. Tsiveriotis, R.A. Brown, Solution of free-boundary problems using finite-element/Newton methods and locally refined grids: application to analysis of solidification microstructure, Int. J. Numer. Meth. Fluids 16 (1993) 827–843.

[13] Y. Dimakopoulos, J.A. Tsamopoulos, A quasi-elliptic transformation for moving boundary problems with large anisotropic deformations, J. Comput. Phys. 192 (2003) 494–522.

[14] Y. Dimakopoulos, J.A. Tsamopoulos, Transient displacement of a viscoplastic material by air in straight and suddenly constricted tubes, J. Non-Newton. Fluid Mech. 112 (2003) 43–75.

[15] Y. Dimakopoulos, J.A. Tsamopoulos, Transient displacement of a Newtonian fluid by air in straight and suddenly constricted tubes, Phys. Fluids 15 (7) (2003) 1973–1991.

[16] Y. Dimakopoulos, J.A. Tsamopoulos, On the gas-penetration in straight tubes completely filled with a viscoelastic fluid, J. Non-Newton. Fluid Mech. 117 (2004) 117–139.

[17] Y. Dimakopoulos, J.A. Tsamopoulos, Transient displacement of a Newtonian and viscoplastic liquid by air in complex tubes, J. Non-Newton. Fluid Mech. 142 (2007) 162–182.

[18] K. Foteinopoulou, V. Mavrantzas, Y. Dimakopoulos, J.A. Tsamopoulos, Numerical simulation of multiple bubbles growing in a Newtonian liquid filament undergoing stretching, Phys. Fluids 18 (4) (2006). 042106(24pp.).

[19] K. Foteinopoulou, V. Mavrantzas, J.A. Tsamopoulos, Numerical simulation of bubble growth in Newtonian and viscoelastic filaments undergoing stretching, J. Non-Newton. Fluid Mech. 122 (2004) 177–200.

[20] G. Karapetsas, J.A. Tsamopoulos, Transient squeeze flow of viscoplastic materials, J. Non-Newton. Fluid Mech. 133 (2006) 35–56.

[21] J. Tsamopoulos, Y. Dimakopoulos, N. Chatzidai, G. Karapetsas, M. Pavlidis, Steady bubble rise and deformation in Newtonian and viscoplastic fluids and conditions for their entrapment, J. Fluid Mech. 601 (2008) 123–164.

[22] G. Karapetsas, J. Tsamopoulos, Steady extrusion of viscoelastic materials from an annular die, J. Non-Newton. Fluid Mech. 154 (2008) 136–152.

[23] M.S. Carvalho, L.E. Scriven, Three-dimensional stability analysis of free surface flows: application to forward deformable roll coating, J. Comput. Phys. 151 (1999) 534–562.

[24] Y. Kallinderis, C. Kontzialis, A priori mesh quality estimation via direct relation between truncation error and mesh distortion, J. Comput. Phys. 228 (3) (2009) 881–902.

[25] 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.

[26] S. Madasu, R.A. Cairncross, Effect of substrate flexibility on dynamic wetting: a finite element model, Comput. Meth. Appl. Mech. Eng. 192 (2003) 2671–2702.

[27] J.F. Thompson, Z.U.A. Warsi, C.W. Mastin, Numerical Grid Generation, North-Holland, Amsterdam, 1985.

[28] J.F. Thompson, F.C. Thames, C.W. Mastin, Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary 2D bodies, J. Comput. Phys. 15 (1974) 299–319.

[29] V. Villamizar, O. Rojas, J. Mabey, Generation of curvilinear coordinates on multiply connected regions with boundary singularities, J. Comput. Phys. 223 (2007) 571–588.

[30] J.F. Thompson, A general 3D elliptic grid generation system on a composite block structure, Comput. Meth. Appl. Mech. Eng. 64 (1987) 377–411.

[31] T.J. Baker, Mesh generation: art or science?, Prog Aerosp. Sci. 41 (2005) 29–63.

[32] C. Kim, Collapse of spherical bubbles in Maxwell fluids, J. Non-Newton. Fluid Mech. 55 (1994) 37–58.

[33] E.A. Brujan, A first order bubble dynamics in a compressible viscoelastic liquid, J. Non-Newton. Fluid Mech. 84 (1999) 83–103.

[34] A.C. Papanastasiou, L.E. Scriven, C.W. Macosko, Bubble growth and collapse in viscoelastic liquids analyzed, J. Non-Newton. Fluid Mech. 16 (1984) 53–75.

[35] D.W. Bousfield, R. Keunings, M.M. Denn, Transient deformation of an inviscid inclusion in a viscoelastic extensional flow, J. Non-Newton. Fluid Mech. 27 (1988) 205–221.

[36] V.F.K. Bjerknes, Fields of Force, Columbia University Press, 1906.

[37] V.F.K. Bjerknes, Die Craftfelder, Vieweg, 1909.

[38] U. Parlitz, R. Mettin, S. Luther, I. Akhatov, M. Voss, W. Lauterborn, Spatio-temporal dynamics of acoustic cavitation bubble clouds, Phil. Trans. R. Soc. Lond. 357 (1999) 313–334.

[39] M.S. Plesset, A. Prosperetti, Bubble dynamics and cavitation, Annu. Rev. Fluid Mech. 9 (1977) 145–185.

[40] A.A. Doinikov, S.T. Zavtrak, On the mutual interaction of two gas bubbles in a sound field, Phys. Fluids 7 (8) (1995) 1923–1930.

[41] N.A. Pelekasis, J.A. Tsamopoulos, Bjerknes forces between two bubbles. Part 1. Response to a step change in pressure, J. Fluid Mech. 254 (1993) 467–499.

[42] N.A. Pelekasis, J.A. Tsamopoulos, Bjerknes forces between two bubbles. Part 2. Response to an oscillatory pressure field, J. Fluid Mech. 254 (1993) 501–527.

[43] N. Phan-Thien, R.I. Tanner, A new constitutive equation derived from network theory, J. Non-Newton. Fluid Mech. 2 (1977) 353–365.

[44] N. Phan-Thien, R.I. Tanner, A nonlinear network viscoelastic model, J. Rheol. 22 (1978) 259–283.

[45] R.A. Brown, M.J. Szady, P.J. Northey, R.C. Armstrong, On the numerical stability of mixed finite-element methods for viscoelastic flows governed by differential constitutive equations, Theor. Comput. Fluid Dyn. 5 (1993) 77–106.

[46] T.C. Papanastasiou, N. Malamataris, K. Ellwood, A new outflow boundary condition, Int. J. Numer. Meth. Fluids 14 (1992) 587–608.

[47] J. Saltzman, J. Brackbill, Applications and generalizations of variational methods for generating adaptive meshes, in: Numerical Grid Generation, Elsevier, Amsterdam, 1982, pp. 865–884.

[48] A.S. Dvinsky, Adaptive grid generation from harmonic maps on Riemannian manifold, J. Comput. Phys. 95 (1991) 450–476.

[49] T.J. Chung, Computational Fluid Dynamics, Cambridge University Press, 2002. pp. 533–580.

[50] B. Szabo, I. Babuska, Finite Element Analysis, Wiley, New York, 1991.

[51] 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. Appl. Mech. Eng. 32 (1982) 199–259.
[52] O.C. Zienkiewicz, R.L. Taylor, The Finite Element Method, fifth ed., Butterworth-Heinemann, Oxford, 2000.
[53] R. Codina, Comparison of some finite element methods for solving the diffusion–convection–reaction equation, Comput. Meth. Appl. Mech. Eng. 156 (1998) 185–210.
[54] 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.
[55] M. Fortin, A. Fortin, A new approach for the FEM simulation of viscoelastic flows, J. Non-Newton. Fluid Mech. 32 (1989) 295–310.
[56] B. Cockburn, Devising discontinuous Galerkin methods for non-linear hyperbolic conservation laws, J. Comput. Appl. Math. 128 (2001) 187–204.
[57] O. Schenk, K. Gärtner, Solving unsymmetric sparse systems of linear equations with PARDISO, J. Future Generat. Comput. Syst. 20 (3) (2004) 475–487.
[58] O. Schenk, K. Gärtner, On fast factorization pivoting methods for symmetric indefinite systems, Electron. Trans. Numer. Anal. 23 (2006) 158–179.
[59] W. Jester, Y. Kallinderis, Numerical study of incompressible flow around fixed cylinder pairs, J. Fluids Struct. 17 (2003) 561–577.
[60] Y. Kallinderis, K. Nakajima, Finite element method for incompressible viscous flows with adaptive hybrid grids, AIAA J. 32 (1994) 1617–1625.
[61] R.B. Bird, R.C. Armstrong, O. Hassager, Dynamics of Polymeric Liquids, John Wiley and Sons, New York, 1987.
[62] C.A. Miller, L.E. Scriven, The oscillations of a fluid droplet immersed in another fluid, J. Fluid Mech. 32 (1968) 417–435.
[63] R.B. Lehoucq, D.C. Sorensen, C. Young, ARPACK User's Guide: Solution of Large Scale Eigenvalue Problems with Implicitly Restarted Arnoldi methods, SIAM, Philadelphia, PA, 1998.