

Numerical simulation of 3D viscoelastic flows with free surfaces

Andrea Bonito ^{a,1}, Marco Picasso ^{a,*}, Manuel Laso ^{b,2}

^a *Institut d'Analyse et Calcul Scientifique, Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland*

^b *Depart. de Ingeniería Química, Escuela Técnica Superior de Ingenieros Industriales, Universidad Politécnica de Madrid, Jose Gutierrez Abascal 2, 28006 Madrid, Spain*

Received 21 July 2005; received in revised form 7 November 2005; accepted 8 November 2005

Available online 20 December 2005

Abstract

A numerical model is presented for the simulation of viscoelastic flows with complex free surfaces in three space dimensions. The mathematical formulation of the model is similar to that of the volume of fluid (VOF) method, but the numerical procedures are different.

A splitting method is used for the time discretization. The prediction step consists in solving three advection problems, one for the volume fraction of liquid (which allows the new liquid domain to be obtained), one for the velocity field, one for the extra-stress. The correction step corresponds to solving an Oldroyd-B fluid flow problem without advection in the new liquid domain.

Two different grids are used for the space discretization. The three advection problems are solved on a fixed, structured grid made out of small cubic cells, using a forward characteristics method. The Oldroyd-B problem without advection is solved using continuous, piecewise linear stabilized finite elements on a fixed, unstructured mesh of tetrahedrons.

Efficient post-processing algorithms enhance the quality of the numerical solution. A hierarchical data structure reduces the memory requirements.

Convergence of the numerical method is checked for the pure extensional flow and the filling of a tube. Numerical results are presented for the stretching of a filament. Fingering instabilities are obtained when the aspect ratio is large. Also, results pertaining to jet buckling are reported.

© 2005 Elsevier Inc. All rights reserved.

1. Introduction

Nowadays, industrial codes are available to solve Newtonian flows with free surfaces. In a number of industrial applications – mold filling for instance – free surfaces with complex topological changes must be considered, which prevents the use of Lagrangian or ALE (Arbitrary Lagrangian–Eulerian) methods. An alternative is to use Eulerian methods and to solve an additional advection equation, namely

* Corresponding author.

E-mail address: marco.picasso@epfl.ch (M. Picasso).

¹ Supported by the Swiss National Science Foundation.

² Manuel Laso gratefully acknowledges funding from EC projects G5RD-CT-2000-00202 (DEFSAM) and G5RD-CT-2002-00720 (PMILS).

$$\frac{\partial \varphi}{\partial t} + \mathbf{u} \cdot \nabla \varphi = 0,$$

where \mathbf{u} is the fluid velocity. In the level-set [34,6,35] or pseudo-concentration [39] approach, the function φ is smooth and the free surface is defined to be the zero level-set. In the VOF (Volume of Fluid) [15] or the volume tracking method [32,33] approach, the function φ denotes the volume fraction of liquid and is a step function having value one in the liquid, zero in the surrounding vacuum.

Level-set and VOF methods have produced an enormous amount of literature, both methods having their advantages and drawbacks. Roughly speaking, VOF-like methods suffer from the lack of regularity of the volume fraction of liquid on the interface. For instance, post-processing algorithms such as SLIC or PLIC are used in order to reduce numerical diffusion, see for instance [33] for a review. Also, special care is needed when computing surface tension, see for instance [33,31]. Since the level-set method involves a continuous function, better accuracy should be expected. However, mass conservation is difficult to obtain so that special procedures must be added, see for instance [42].

Although the VOF model was initially solved using finite volumes, finite element implementations have been recently proposed in three space dimensions [25,20]. In [19,20,5,4], the VOF formulation was solved using an order one, implicit splitting algorithm, two different grids being used, see Fig. 1. In the prediction step, advection was solved on a structured grid of small cubic cells. In the correction step, diffusion was solved on an unstructured finite element tetrahedral mesh. The reason for using two different grids are the following. Firstly, the advection step is much easier to implement and requires less CPU time on a structured grid rather on a general unstructured finite element mesh. Secondly, the size of the structured cells should be small compared to the size of the unstructured finite element mesh (typically three to five times smaller), so that numerical diffusion of the volume fraction of liquid is as small as possible. Thirdly, the use of unstructured finite elements during the diffusion step of the algorithm allows computational domains with complex shapes to be considered. This approach has been successfully applied to Newtonian flows with complex free surfaces, see [19,20,5,4]. Our goal is to extend it to Non-Newtonian (viscoelastic) computations.

Recently, viscoelastic flows with free surfaces have received much attention, see for instance [40,43,37,29,2]. In a number of papers, the filament stretching rheometer was considered, this experiment being well suited to Lagrangian computations in two [43,37] and three space dimensions [29,2]. However, Lagrangian or ALE computations are not well suited to describe complex topology changes observed when cavitation or breakage occurs in the filament, whereas Eulerian formulations (level-set and VOF like methods) seem to be more appropriate. Two-dimensional computations of viscoelastic flows using the VOF method have already been presented. For instance in [40,16], the capabilities of the VOF method has been demonstrated for a number of problems such as jet swell, jet buckling or impacting drops. Also, two-dimensional computations using the VOF method and CONNFFESSIT like models have been presented in [13]. It should be mentioned that the level set method has also been successfully used for viscoelastic fluids with free surfaces in [11] using the hybrid particle level set method of [10].

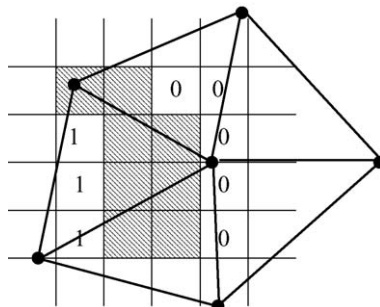


Fig. 1. Two grids are used for the computations. In order to reduce numerical diffusion and to simplify the implementation, the volume fraction of liquid is computed on a structured grid of small cells. The velocity, pressure and extra-stress are computed on an unstructured finite element mesh with larger size. The symbol 1 (resp. 0) denotes a cell completely filled (resp. empty). The cells which are partially filled are shaded.

The goal of this paper is to extend the three-dimensional Newtonian model described in [20] to viscoelastic computations. In the next section, the mathematical model is presented. The problem unknowns are the velocity, pressure and extra-stress in the liquid region, plus the volume fraction of liquid in the whole cavity containing the liquid. In Section 3, an order one, implicit splitting algorithm is proposed. The prediction step consists in solving three advection problems, one for the volume fraction of liquid, one for the velocity field, one for the extra-stress. The correction step then corresponds to solving an Oldroyd-B flow problem without advection. In Section 4, the space discretization is considered, two different grids being used. The three advection problems are solved on a fixed, structured grid made out of small cubic cells, using a forward characteristics method. The Oldroyd-B flow problem without advection is then solved using continuous, piecewise linear stabilized finite elements on a fixed, unstructured mesh of tetrahedrons. Efficient post-processing algorithms enhance the quality of the numerical solution. A hierarchical data structure reduces the memory requirements. Numerical results are presented in Section 5. First, convergence of the numerical model is checked for the pure extensional flow and the filling of a tube. Then, numerical results are reported for the stretching of a filament and for jet buckling.

2. The mathematical model

Let A be a cavity of \mathbb{R}^3 in which an Oldroyd-B fluid is confined, and let $T > 0$ be the final time of the simulation. At time t , the liquid region is denoted $\Omega(t)$. Finally, let \mathcal{Q}_T be the space–time domain containing the liquid

$$\mathcal{Q}_T = \{(x, t) \in A \times (0, T); x \in \Omega(t), 0 < t < T\},$$

and let Σ_T be the space–time free surface between the liquid and the surrounding air. The notation is reported in two space dimensions in Fig. 2.

In the liquid region, the velocity field $\mathbf{u} : \mathcal{Q}_T \rightarrow \mathbb{R}^3$, the pressure field $p : \mathcal{Q}_T \rightarrow \mathbb{R}$ and the symmetric extra-stress tensor field $\boldsymbol{\sigma} : \mathcal{Q}_T \rightarrow \mathbb{R}^{3 \times 3}$ must satisfy:

$$\begin{aligned} \rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} - 2\eta_s \operatorname{div} \boldsymbol{\epsilon}(\mathbf{u}) + \nabla p - \operatorname{div} \boldsymbol{\sigma} &= \rho \mathbf{g}, \\ \operatorname{div} \mathbf{u} &= 0, \\ \boldsymbol{\sigma} + \lambda \left(\frac{\partial \boldsymbol{\sigma}}{\partial t} + (\mathbf{u} \cdot \nabla)\boldsymbol{\sigma} - \nabla \mathbf{u} \boldsymbol{\sigma} - \boldsymbol{\sigma} \nabla \mathbf{u}^T \right) - 2\eta_p \boldsymbol{\epsilon}(\mathbf{u}) &= 0. \end{aligned}$$

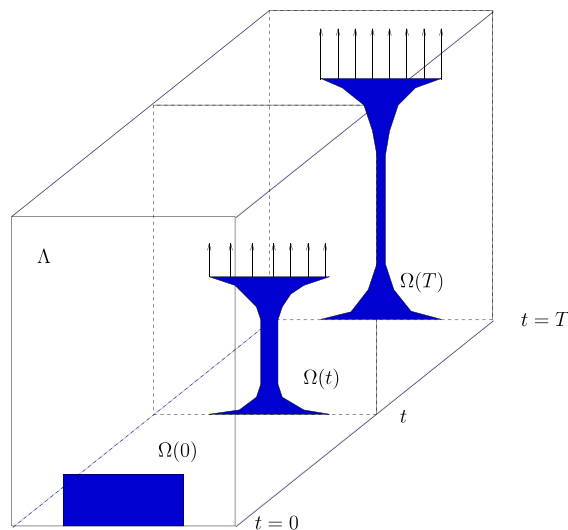


Fig. 2. Calculation domain for the stretching of a filament in two space dimensions. At initial time, the viscoelastic fluid is at rest and occupies the domain $\Omega(0)$. At $t > 0$, the upper plate moves at given velocity.

Here ρ is the density, \mathbf{g} is the gravity, $\eta_s \geq 0$ and $\eta_p > 0$ are the solvent and polymer viscosities, λ is the relaxation time, $(\nabla \mathbf{u})_{ij} = \partial u_i / \partial x_j$ is the velocity gradient, $\epsilon(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$ is the strain rate tensor.

Let $\varphi : A \times [0, T] \rightarrow \mathbb{R}$ be the volume fraction of liquid. The function φ is a step function, equals one if liquid is present and zero if it is not; thus φ is the characteristic function of the liquid region

$$\Omega(t) = \{x \in A; \varphi(x, t) = 1\}.$$

Since the interface moves with the liquid, the function φ must satisfy (in a weak sense)

$$\frac{\partial \varphi}{\partial t} + \mathbf{u} \cdot \nabla \varphi = 0 \quad \text{in } Q_T. \tag{1}$$

From a Lagrangian point of view, the function φ is constant along the trajectories of the fluid particles. More precisely, $\varphi(X(t), t) = \varphi(X(0), 0)$, where $X(t)$ is the trajectory of a fluid particle, thus $X'(t) = \mathbf{u}(X(t), t)$.

Initial and boundary conditions are as follows. At the initial time, the volume fraction of liquid $\varphi(\cdot, 0)$ is given, which defines the liquid region,

$$\Omega(0) = \{x \in A; \varphi(x, 0) = 1\},$$

see Fig. 2 for the notations in two space dimensions. The initial velocity field \mathbf{u} and extra-stress tensor σ are then prescribed in $\Omega(0)$. Let us now turn to the boundary conditions for the velocity field. It is assumed that no external forces act on the free surface Σ_T (effects of surface tension are neglected):

$$-p\mathbf{n} + (2\eta_s\epsilon(\mathbf{u}) + \sigma)\mathbf{n} = \mathbf{0} \quad \text{on } \Sigma_T, \tag{2}$$

where \mathbf{n} is the unit outer normal of Σ_T . Neglecting surface tension effects may not be correct in many applications however, we did not implement surface tension for the following reasons. Firstly, realistic results can be obtained when including viscoelastic effects and without considering surface tension forces, see the numerical results of Section 5. Secondly, even though accurate procedures are available in VOF-like methods to compute surface tension [14,17,33,30], the mesh size required to obtain a good approximation of curvature would be too small to allow viscoelastic computations in three space dimensions.

On the boundary of the liquid region being in contact with the walls (that is to say the boundary of the cavity A , see Fig. 2), essential boundary conditions (that is to say imposed velocity components) or natural boundary conditions (that is to say b.c. which are not explicitly enforced but which are implicitly included in the weak formulation) can be imposed for the velocity and the extra-stress. Let us, for instance, consider the two following situations: (i) a jet emerging from a die; (ii) the stretching of a filament, see Fig. 3. In case (i), both the velocity and extra-stress are imposed at the inflow boundary. Either slip or no-slip boundary conditions apply on the other boundaries of the cavity A . If no-slip conditions are enforced for the velocity, then

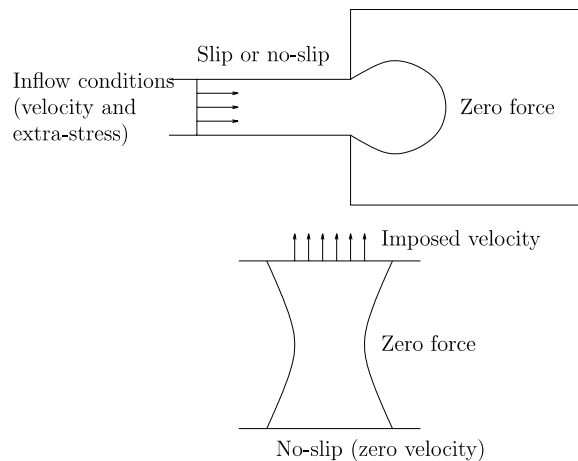


Fig. 3. Boundary conditions. Top: jet emerging from a die. Bottom: stretching of a filament.

no other conditions apply. If slip boundary conditions are enforced that is $\mathbf{u} \cdot \mathbf{n} = 0$, then the fluid tangent force should also be set to zero, namely

$$(-p\mathbf{n} + (2\eta_s \epsilon(\mathbf{u}) + \boldsymbol{\sigma})\mathbf{n}) \cdot \mathbf{t}_i = 0, \quad i = 1, 2, \tag{3}$$

where \mathbf{t}_1 and \mathbf{t}_2 are two unit vectors tangent to the boundary of the cavity. In case (ii), the velocity is imposed on the top and bottom sides of the cavity whereas (2) applies on the lateral side. Boundary conditions (2) and (3) are straightforward to implement in the framework of finite element methods since the corresponding terms vanish after integration by parts in the variational formulation, see Section 4.2 hereafter.

3. Time discretization: an implicit splitting algorithm

The implicit, order one, splitting algorithm described in [19,20,5] for Newtonian flows is extended here to viscoelastic situations. This splitting algorithm allows advection phenomena to be decoupled from other phenomena. The reader should note that a similar algorithm has already been presented for viscoelastic flow computations in fixed domains [23].

Let $0 = t^0 < t^1 < t^2 < \dots < t^N = T$ be a subdivision of the time interval $[0, T]$, define $\Delta t^n = t^n - t^{n-1}$ the n th time step, $n = 1, 2, \dots, N$, Δt the largest time step. At time t^{n-1} , assume that an approximation $\varphi^{n-1} : A \rightarrow \mathbb{R}$ of the volume fraction of liquid is known, which defines the liquid region:

$$\Omega^{n-1} = \{x \in A; \varphi^{n-1}(x) = 1\}.$$

Also assume that approximations of the velocity $\mathbf{u}^{n-1} : \Omega^{n-1} \rightarrow \mathbb{R}^3$ and the extra stress $\boldsymbol{\sigma}^{n-1} : \Omega^{n-1} \rightarrow \mathbb{R}^{3 \times 3}$ are available. Then $\varphi^n, \Omega^n, \mathbf{u}^n, \boldsymbol{\sigma}^n$ are computed by means of a splitting algorithm as illustrated in Fig. 4. The prediction step consists in solving three advection problems, which yields the new volume fraction of liquid φ^n , the new liquid region Ω^n , the predicted velocity $\mathbf{u}^{n-\frac{1}{2}} : \Omega^n \rightarrow \mathbb{R}^3$ and the predicted extra-stress $\boldsymbol{\sigma}^{n-\frac{1}{2}} : \Omega^n \rightarrow \mathbb{R}^{3 \times 3}$. Then, the correction step is performed, a generalized Stokes problem is solved, which yields the new velocity $\mathbf{u}^n : \Omega^n \rightarrow \mathbb{R}^3$ and pressure $p^n : \Omega^n \rightarrow \mathbb{R}$. The new extra-stress $\boldsymbol{\sigma}^n : \Omega^n \rightarrow \mathbb{R}^{3 \times 3}$ is then updated from the Oldroyd-B constitutive equation.

3.1. Prediction step: advection

The prediction step consists in solving between time t^{n-1} and t^n the three advection problems:

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = 0, \tag{4}$$

$$\frac{\partial \boldsymbol{\tau}}{\partial t} + (\mathbf{v} \cdot \nabla) \boldsymbol{\tau} = 0, \tag{5}$$

$$\frac{\partial \psi}{\partial t} + \mathbf{v} \cdot \nabla \psi = 0, \tag{6}$$

with initial conditions

$$\mathbf{v}(t^{n-1}) = \mathbf{u}^{n-1},$$

$$\boldsymbol{\tau}(t^{n-1}) = \boldsymbol{\sigma}^{n-1},$$

$$\psi(t^{n-1}) = \varphi^{n-1}.$$

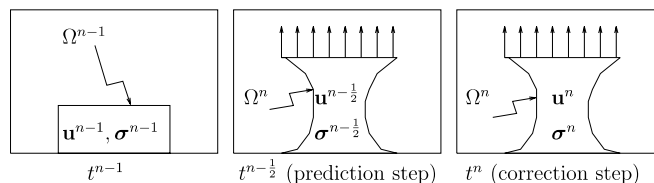


Fig. 4. The splitting algorithm.

These three problems can be solved exactly using the method of characteristics [26–28], the trajectories of the velocity field being straight lines. Indeed, the trajectories are given by $X'(t) = \mathbf{v}(X(t), t)$, but since \mathbf{v} is constant along the trajectories, we have $X'(t) = \mathbf{v}(X(t^{n-1}), t^{n-1}) = \mathbf{u}^{n-1}(X(t^{n-1}))$. Let $\mathbf{u}^{n-\frac{1}{2}}, \boldsymbol{\sigma}^{n-\frac{1}{2}}, \varphi^n$ denote the solution at time t^n of 4, 5, 6, respectively. We thus have

$$\mathbf{u}^{n-\frac{1}{2}}(x + \Delta t^n \mathbf{u}^{n-1}(x)) = \mathbf{u}^{n-1}(x), \quad (7)$$

$$\boldsymbol{\sigma}^{n-\frac{1}{2}}(x + \Delta t^n \mathbf{u}^{n-1}(x)) = \boldsymbol{\sigma}^{n-1}(x), \quad (8)$$

$$\varphi^n(x + \Delta t^n \mathbf{u}^{n-1}(x)) = \varphi^{n-1}(x) \quad (9)$$

for all x belonging to Ω^{n-1} . Once φ^n is known in the cavity A , then the liquid region at time t^n is defined by

$$\Omega^n = \{y \in A; \varphi^n(y) = 1\}.$$

At this point it should be stressed that after the prediction step, the obtained velocity $\mathbf{u}^{n-\frac{1}{2}}$ is not divergence free. The divergence free property is obtained after the correction step, see Eqs. (10) and (11).

3.2. Correction step: Stokes and Oldroyd-B

The new liquid region Ω^n being known, the predicted velocity $\mathbf{u}^{n-\frac{1}{2}} : \Omega^n \rightarrow \mathbb{R}^3$ and the extra-stress $\boldsymbol{\sigma}^{n-\frac{1}{2}} : \Omega^n \rightarrow \mathbb{R}^{3 \times 3}$ being also known, the new velocity \mathbf{u}^n is obtained by solving a generalized Stokes problem:

$$\rho \frac{\mathbf{u}^n - \mathbf{u}^{n-\frac{1}{2}}}{\Delta t^n} - 2\eta_s \operatorname{div} \boldsymbol{\epsilon}(\mathbf{u}^n) + \nabla p^n - \operatorname{div} \boldsymbol{\sigma}^{n-\frac{1}{2}} = \rho \mathbf{g} \quad \text{in } \Omega^n, \quad (10)$$

$$\operatorname{div} \mathbf{u}^n = 0 \quad \text{in } \Omega^n, \quad (11)$$

then, the new extra-stress $\boldsymbol{\sigma}^n$ is obtained from Oldroyd-B constitutive equation:

$$\boldsymbol{\sigma}^n + \lambda \left(\frac{\boldsymbol{\sigma}^n - \boldsymbol{\sigma}^{n-\frac{1}{2}}}{\Delta t^n} - \nabla \mathbf{u}^n \boldsymbol{\sigma}^{n-\frac{1}{2}} - \boldsymbol{\sigma}^{n-\frac{1}{2}} (\nabla \mathbf{u}^n)^T \right) - 2\eta_p \boldsymbol{\epsilon}(\mathbf{u}^n) = 0. \quad (12)$$

4. Space discretization and implementation

Two distinct grids are used to solve the prediction and correction steps. Since the shape of the cavity A can be complex (this is for instance the case in mold filling or extrusion processes), finite element techniques are well suited for solving (10)–(12) using an unstructured mesh. On the other hand, a structured grid of cubic cells is used to implement (7)–(9). The reasons for using a structured grid is the following. Firstly, the method of characteristics can be easily implemented on structured grids. Secondly, the size of the cells can be tuned in order to control numerical diffusion when projecting (7)–(9) on the structured grid. Numerical experiments reported in [19,20,5] have shown that choosing the cells spacing three to five times smaller than the mesh spacing is a good trade-off between numerical diffusion and computational cost or memory storage.

4.1. Advection step: structured grid of cubic cells

The implementation of (7)–(9) is now discussed. Assume that the grid is made out of cubic cells C_{ijk} of size h . Let $\varphi_{ijk}^{n-1}, \mathbf{u}_{ijk}^{n-1}$ and $\boldsymbol{\sigma}_{ijk}^{n-1}$ be the approximate value of φ, \mathbf{u} and $\boldsymbol{\sigma}$ at center of cell number (ijk) and time t^{n-1} . According to (7)–(9), the advection step on cell number (ijk) consists in advecting $\varphi_{ijk}^{n-1}, \mathbf{u}_{ijk}^{n-1}$ and $\boldsymbol{\sigma}_{ijk}^{n-1}$ by $\Delta t^n \mathbf{u}_{ijk}^{n-1}$ and then projecting the values onto the structured grid. An example of cell advection and projection is presented in Fig. 5 in two space dimensions.

This advection algorithm is unconditionally stable with respect to the CFL condition – velocity times the time step divided by the cells spacing h – and $O(\Delta t + h^2/\Delta t)$ convergent, according to the theoretical results

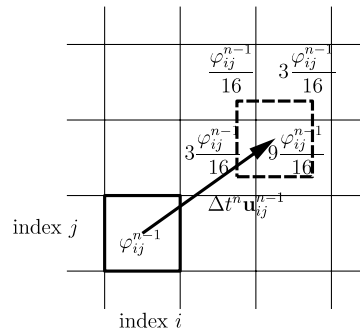


Fig. 5. An example of two-dimensional advection of φ_{ij}^{n-1} by $\Delta t^n \mathbf{u}_{ij}^{n-1}$, and projection on the grid. The advected cell is represented by the dashed lines. The four cells containing the advected cell receive a fraction of φ_{ij}^{n-1} , according to the position of the advected cell. In this example, the new values of the volume fraction of liquid φ^n are updated as follows: $\varphi_{i+1,j+1}^n = \varphi_{i+1,j+1}^{n-1} + 3/16\varphi_{ij}^{n-1}$; $\varphi_{i+2,j+1}^n = \varphi_{i+2,j+1}^{n-1} + 9/16\varphi_{ij}^{n-1}$; $\varphi_{i+1,j+2}^n = \varphi_{i+1,j+2}^{n-1} + 1/16\varphi_{ij}^{n-1}$; $\varphi_{i+2,j+2}^n = \varphi_{i+2,j+2}^{n-1} + 3/16\varphi_{ij}^{n-1}$.

available for the characteristics-Galerkin method [26–28]. However, this algorithm has two drawbacks. Indeed, numerical diffusion is introduced when projecting the values of the advected cells on the grid (remember that the volume fraction of liquid is discontinuous across the interface). Moreover, if the time step is too large, two cells may arrive at the same place, producing numerical (artificial) compression.

In order to enhance the quality of the volume fraction of liquid, two post-processing procedures have been implemented. We refer to [19,20,5] for a description in two and three space dimensions. The first procedure reduces numerical diffusion and is a simplified implementation of the SLIC (Simple Linear Interface Calculation) algorithm [7,22,33], see Figs. 6 and 7 for a simple example. In the SLIC procedure, if a cell is partially filled with liquid, then the volume fraction of liquid is condensed along the cells faces, edges or corners (see Fig. 8), according to the volume fraction of liquid of the neighbouring cells (see Fig. 9).

The second procedure removes artificial compression (that is values of the volume fraction of liquid greater than one), which may happen when the volume fraction of liquid advected in two cells arrive at the same place, see Fig. 10. The aim of this procedure is to produce new values φ_{ijk}^n which are between zero and one and is as follows. At each time step, all the cells having values φ_{ijk}^n greater than one (strictly) or between zero and one (strictly) are sorted according to their values φ_{ijk}^n . This can be done in an efficient way using quick sort algorithms. The cells having values φ_{ijk}^n greater than one are called the dealer cells, whereas the cells having values φ_{ijk}^n between zero and one are called the receiver cells. The second procedure then consists in moving the fraction of liquid in excess in the dealer cells to the receiver cells, see [19,18] for details.

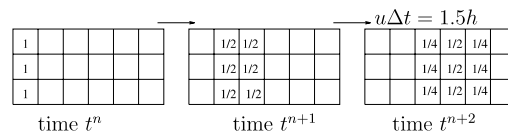


Fig. 6. Numerical diffusion during the advection step. At time t^n , the cells have volume fraction of liquid one or zero. The velocity u is horizontal, the time step Δt is chosen so that $u\Delta t = 1.5h$ where h is the cells spacing.

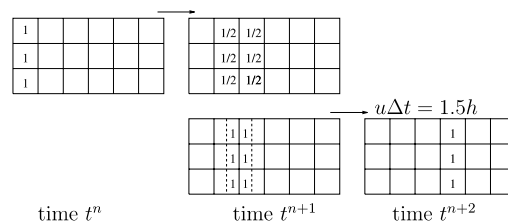


Fig. 7. Reducing numerical diffusion using the SLIC algorithm. Before advecting a cell partially filled with liquid, the volume fraction of liquid is condensed along the cells boundaries, according to the neighbouring cells.

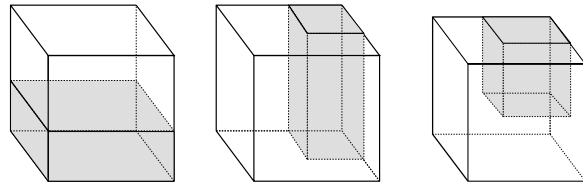


Fig. 8. SLIC algorithm. If the cell is partially filled with liquid, the liquid is pushed along a face, an edge, or a vertex of the cell, according to the neighbours volume fraction of liquid.

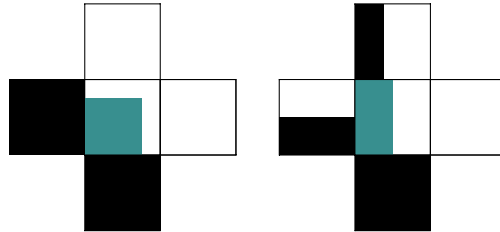


Fig. 9. SLIC algorithm. The volume fraction of liquid in a cell partially filled with liquid is pushed according to the volume fraction of liquid of the neighbouring cells. Two examples are proposed. Left: the left and bottom neighbouring cells are full of liquid, the right and top neighbouring cells are empty, the liquid is pushed at the bottom left corner of the cell. Right: the bottom neighbouring cell is full of liquid, the right neighbouring cell is empty, the other two neighbouring cells are partially filled with liquid, the volume fraction of liquid is pushed along the left side of the cell.

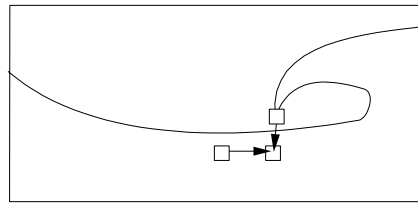


Fig. 10. An example of numerical (artificial) compression.

Validation of these procedures using standard two-dimensional test cases taken from [1,32] have been performed in [5]. Translation, rotation and stretching of a circular region of fluid are shown in Fig. 11. For more details we refer to Section 5.1 of [5].

In a number of industrial applications, the shape of the cavity containing the liquid is complex. Therefore, a special data structure has been implemented in order to reduce the memory requirements used to store the cell data. An example is proposed in Fig. 12. The cavity containing the liquid is meshed into tetrahedrons. Without

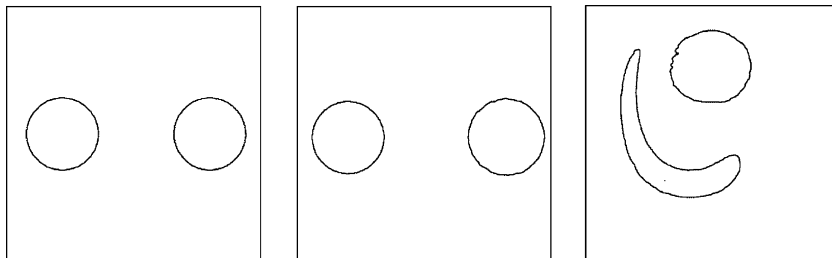


Fig. 11. Validation of the advection step. Left: Translation of a circular region of liquid, the interface is shown at time $t = 0$ and $t = 0.06$ s. Middle: Rotation of a circular region of liquid, the interface is shown at time $t = 0$ and $t = 0.126$ s. Right: Single vortex test case, the interface is shown at time $t = 1$ (maximal deformation) and $t = 2$ s (return to initial circular shape).

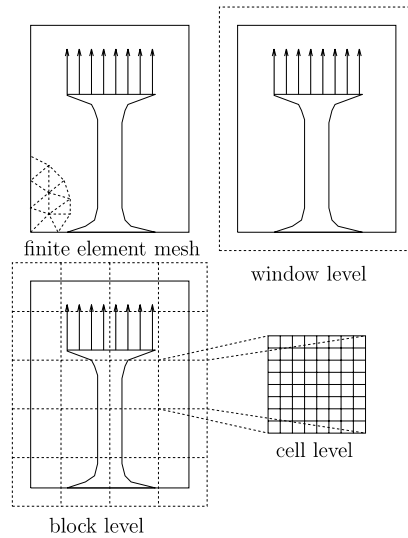


Fig. 12. The hierarchical window-block-cell data structure used to implement cells advection in the framework of the 2D filament stretching.

any particular cells data structure, a great number of cells would be stored in the memory without ever being used. The data structure makes use of three hierarchical levels to define the cells. At the coarsest level, the cavity is meshed into windows which can be glued together. Each window is then subdivided into blocks. Finally, a block is cut into smaller cubes, namely the cells (ijk) . When a block is free of liquid ($\varphi = 0$), it is switched off, that is to say the memory corresponding to the cells is not allocated. When liquid enters a block, the block is switched on, that is to say the memory corresponding to the cells is allocated.

Once values φ_{ijk}^n , $\mathbf{u}_{ijk}^{n-\frac{1}{2}}$ and $\boldsymbol{\sigma}_{ijk}^{n-\frac{1}{2}}$ have been computed on the cells (ijk) , values are interpolated at the vertexes P of the finite element mesh. More precisely, the volume fraction of liquid at vertex P is computed by considering all the cells (ijk) contained in the triangles K containing the vertex P , see Fig. 13, using the following formula:

$$\varphi^n(P) = \frac{\sum_{K \ni P} \sum_{(ijk) \subset K} \phi_P(x_{ijk}) \varphi_{ijk}^n}{\sum_{K \ni P} \sum_{(ijk) \subset K} \phi_P(x_{ijk})}, \tag{13}$$

where x_{ijk} denotes the center of cell (ijk) and ϕ_P is the finite element basis function attached to vertex P . Similar formula hold for the velocity and extra-stress. Then, the liquid region is defined as follows. An element (tetrahedron) of the mesh is said to be liquid if (at least) one of its vertexes has a volume fraction of liquid $\varphi^n > 0.5$, see Fig. 14. The computational domain Ω^n used for solving (10)–(12) is then defined to be the union of all liquid elements. At this point, we would like to stress that the values of the volume fraction of liquid on the unstructured finite element mesh are only used in order to define the liquid region. Again, advection of the volume fraction of liquid only occurs on the structured cells, and not on the unstructured

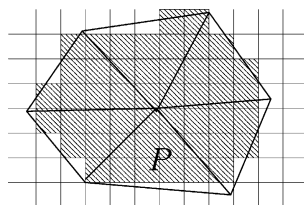


Fig. 13. Interpolation of the volume fraction of liquid from the structured cells to the unstructured finite element mesh. The volume fraction of liquid at vertex P depends on the volume fraction of liquid in the shaded cells.

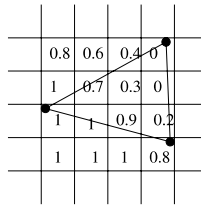


Fig. 14. A two-dimensional example of liquid element. The values of the volume fraction of liquid φ at the center of the cells are known. A value φ is then interpolated at the vertexes of the finite element mesh. The displayed triangle has at least one vertex with value φ greater than 0.5. Therefore, the triangle is liquid and the velocity, the pressure and the extra-stress will be computed at the three vertexes of the triangle.

finite element mesh. Also, the volume constraint is not directly enforced in the numerical model. However, if numerical diffusion of the volume fraction of liquid is small, then the volume constraint will be satisfied. This is precisely the goal of the two post-processing procedures that have been added. In all the computations, we have observed that the (numerical) diffusion layer of the volume fraction of liquid ($0 < \varphi < 1$) is of the order of one or two cells and that the volume constraint is satisfied up to 1%. In order to achieve this goal, the two post-processing procedures must be switched on and the cells spacing must be three to five times smaller than the mesh spacing.

4.2. Correction step: Stokes and Oldroyd-B with finite elements

Let us now turn to the finite element techniques used for solving (10)–(12). We follow [3,24] and use an EVSS (Elastic Viscous Split Stress) formulation with continuous, piecewise linear stabilized finite elements. More precisely, given the predicted velocity $\mathbf{u}^{n-\frac{1}{2}}$, an extra-variable $\mathbf{D}^{n-\frac{1}{2}}$ defined by

$$\int_{\Omega^n} \mathbf{D}^{n-\frac{1}{2}} : \mathbf{E} \, dx = \int_{\Omega^n} \boldsymbol{\epsilon}(\mathbf{u}^{n-\frac{1}{2}}) : \mathbf{E} \, dx \quad \forall \mathbf{E},$$

is introduced for stability purposes. No boundary conditions apply to $\mathbf{D}^{n-\frac{1}{2}}$. This equation results in solving a diagonal linear system provided a mass lumping quadrature formula is used. Since the mass lumping quadrature formula is order two accurate in space, the global accuracy of the method is not affected. Once $\mathbf{D}^{n-\frac{1}{2}}$ is computed, the new velocity \mathbf{u}^n and pressure p^n are obtained by solving the following Stokes problem:

$$\begin{aligned} & \int_{\Omega^n} \rho \frac{\mathbf{u}^n - \mathbf{u}^{n-\frac{1}{2}}}{\Delta t^n} \cdot \mathbf{v} \, dx + 2(\eta_s + \eta_p) \int_{\Omega^n} \boldsymbol{\epsilon}(\mathbf{u}^n) : \boldsymbol{\epsilon}(\mathbf{v}) \, dx - \int_{\Omega^n} p^n \operatorname{div} \mathbf{v} \, dx \\ & = \int_{\Omega^n} (2\eta_p \mathbf{D}^{n-\frac{1}{2}} - \boldsymbol{\sigma}^{n-\frac{1}{2}}) : \boldsymbol{\epsilon}(\mathbf{v}) \, dx + \int_{\Omega^n} \rho \mathbf{g} \cdot \mathbf{v} \, dx \quad \forall \mathbf{v}, \\ & \int_{\Omega^n} \operatorname{div} \mathbf{u}^n q \, dx + \sum_{K \subset \Omega^n} \alpha_K \int_K \left(\rho \frac{\mathbf{u}^n - \mathbf{u}^{n-\frac{1}{2}}}{\Delta t^n} + \nabla p^n - \operatorname{div} \boldsymbol{\sigma}^{n-\frac{1}{2}} - \rho \mathbf{g} \right) \cdot \nabla q \, dx = 0 \quad \forall q. \end{aligned} \tag{14}$$

Here K denotes a tetrahedron, α_K is the local stabilization coefficient defined by

$$\alpha_K = \begin{cases} \frac{|K|^{\frac{2}{3}}}{12(\eta_s + \eta_p)} & \text{if } Re_K \leq 3, \\ \frac{|K|^{\frac{2}{3}}}{4Re_K(\eta_s + \eta_p)} & \text{else,} \end{cases}$$

where the local Reynolds number Re_K is defined by

$$Re_K = \frac{\rho |K|^{\frac{1}{3}} \max_{x \in K} |\mathbf{u}^{n-\frac{1}{2}}(x)|}{2(\eta_s + \eta_p)}.$$

Note that in (14) the corrected velocity \mathbf{u}^n can be prescribed on the boundary of the cavity Λ whenever needed, see Fig. 3 for a discussion related to boundary conditions. Also note that the boundary condition (2) is implicitly contained in the above variational formulation. Indeed, (14) has been obtained by multiplying the momentum equation with a test function \mathbf{v} , integrating by parts, and using the boundary condition (2). Thus, from the implementation point of view, no additional work is required to enforce (2). All the degrees of freedom corresponding to velocity and pressure are stored in a single matrix and the linear system is solved using the GMRES algorithm with a classical incomplete LU preconditioner and no restart.

It then remains to update the extra-stress $\boldsymbol{\sigma}^n$ from Oldroyd-B constitutive equation:

$$\left(1 + \frac{\lambda}{\Delta t^n}\right) \int_{\Omega^n} \boldsymbol{\sigma}^n : \boldsymbol{\tau} \, dx = \lambda \int_{\Omega^n} \left(\frac{1}{\Delta t^n} \boldsymbol{\sigma}^{n-\frac{1}{2}} + \nabla \mathbf{u}^n \boldsymbol{\sigma}^{n-\frac{1}{2}} + \boldsymbol{\sigma}^{n-\frac{1}{2}} (\nabla \mathbf{u}^n)^T \right) : \boldsymbol{\tau} \, dx + 2\eta_p \int_{\Omega^n} \boldsymbol{\epsilon}(\mathbf{u}^n) : \boldsymbol{\tau} \, dx \quad \forall \boldsymbol{\tau}.$$

Here $\boldsymbol{\sigma}^n$ must be prescribed at the inflow boundary, if there is one, see Fig. 3. Again, this equation results in solving a diagonal linear system whenever a mass lumping quadrature formula is used. In [3,24] it is proved that this finite element scheme is convergent for stationary problems in fixed computational domains. More precisely, it is proved that the approximate velocity gradient, the approximate pressure and the approximate extra-stress converge with order one in space in the L^2 norm, even when the solvent viscosity is small compared to the polymer viscosity.

Finally, once the new velocity \mathbf{u}^n and extra-stress $\boldsymbol{\sigma}^n$ are computed at the vertexes of the finite element mesh, values are interpolated at the center of the cells (ijk) :

$$\mathbf{u}_{ijk}^n = \sum_P \phi_P(x_{ijk}) \mathbf{u}_P^n, \tag{15}$$

where P denotes a mesh vertex, x_{ijk} denotes the center of cell (ijk) , ϕ_P denotes the finite element basis function corresponding to vertex P and \mathbf{u}_P^n is the velocity at vertex P . A similar formula is used for the extra-stress $\boldsymbol{\sigma}_{ijk}^n$. Please note that the volume fraction of liquid is not interpolated from the finite element mesh to the cells. Indeed, the volume fraction of liquid is only computed on the structured cells. It is interpolated on the unstructured finite element mesh only in order to define the liquid region after the prediction step, see the end of Section 4.1.

4.3. Implementations details

The memory storage is the following. For each cubic cell, the volume fraction of liquid, the velocity and the extra-stress must be stored in order to implement (7)–(9), therefore $1 + 3 + 6 = 10$ values. For each vertex of the finite element mesh, the velocity, the pressure, the extra-stress and the EVSS field $D = \boldsymbol{\epsilon}(\mathbf{u})$ must be stored, therefore $3 + 1 + 6 + 6 = 16$ values. The code is written in the C++ programming language and the finite element data structure is classical. The data structure of the cells is as follows. Each cell is labelled by indices (ijk) within a block. Also, each block is labelled by indices (ijk) within a window, see Fig. 12.

In order to perform efficient interpolation between the two grids (structured cells/unstructured finite elements), the following data structure is needed. In order to implement interpolation from the finite element mesh to the cells, Eq. (15), the index of the finite element (tetrahedron) containing each cell is needed. Alternatively, in order to implement interpolation from the cells to the finite element mesh, Eq. (13), the list of the cells contained in each finite element (tetrahedron) is required. This additional data structure is built at the beginning of each computation. It can be stored in case several computations are performed with the same grids. The additional CPU time required to build this data structure is small (less than 1%) compared to the total CPU time.

5. Numerical results

Several tests are presented in this section. Firstly, our numerical model is validated for two simple flows, namely an elongational flow and the filling of a tube. Then, numerical experiments corresponding to the stretching of a filament and jet buckling are considered.

5.1. Elongational flow

At the initial time, liquid at rest occupies a cylinder with radius $R_0 = 0.0034$ m and height $L_0 = 0.0019$ m. Then, the velocity field on the top and bottom sides of the cylinder is imposed to be

$$\mathbf{u}(x, y, z, t) = \begin{pmatrix} -\frac{1}{2}\dot{\epsilon}_0 x \\ -\frac{1}{2}\dot{\epsilon}_0 y \\ \dot{\epsilon}_0 z \end{pmatrix},$$

with $\dot{\epsilon}_0 = 4.68 \text{ s}^{-1}$, whereas (2) applies on the lateral sides. Since there is no inflow velocity, no boundary conditions have to be enforced for the extra-stress. A simple calculation shows that, for all time t , the above velocity field satisfies the momentum equations, that the extra-stress tensor is homogeneous, for instance

$$\sigma_{zz}(x, y, z, t) = \frac{2\eta_p \dot{\epsilon}_0}{1 - 2\dot{\epsilon}_0 \lambda} \left(1 - e^{-(\frac{1}{2} - 2\dot{\epsilon}_0)t} \right),$$

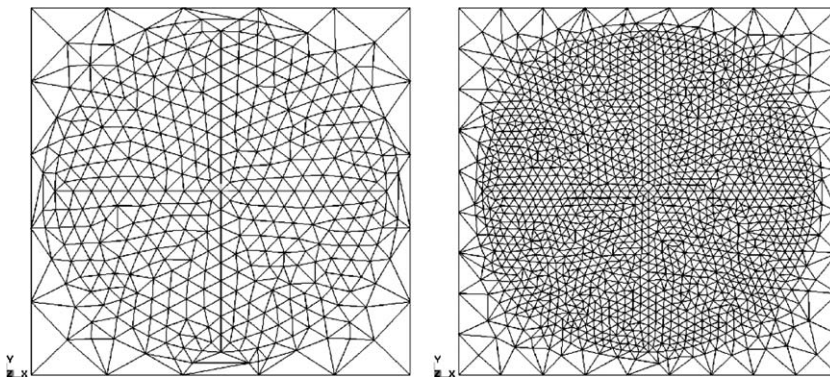


Fig. 15. Elongational flow: 2D cut of the mesh at $z = 0$; left: coarse mesh; right: fine mesh.

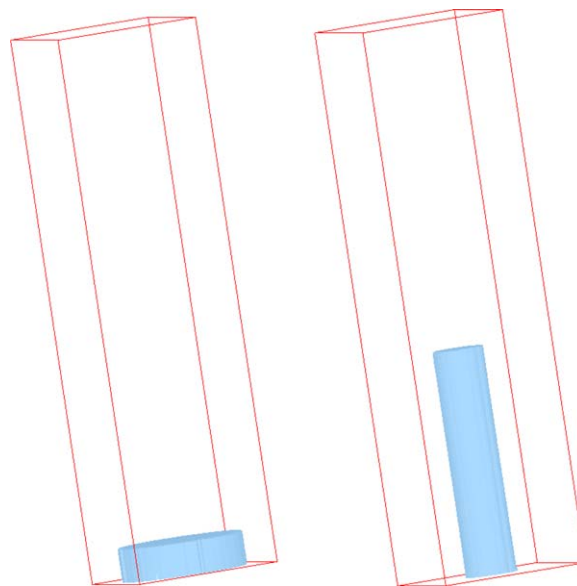


Fig. 16. Elongational flow: shape of the liquid region (the volume corresponding to volume fraction of liquid $\varphi > 0.5$ is shown); left: $t = 0$; right: $t = 0.3$ s.

and that the liquid region remains a cylinder with radius $R(t) = R_0 e^{-\frac{1}{2}\epsilon_0 t}$. Indeed, the trajectories of the fluid particles are defined by $X'(t) = \mathbf{u}(X(t), t)$ which yields

$$\begin{pmatrix} X(t) = X(0)e^{-\frac{1}{2}\epsilon_0 t} \\ Y(t) = Y(0)e^{-\frac{1}{2}\epsilon_0 t} \\ Z(t) = Z(0)e^{\epsilon_0 t} \end{pmatrix}.$$

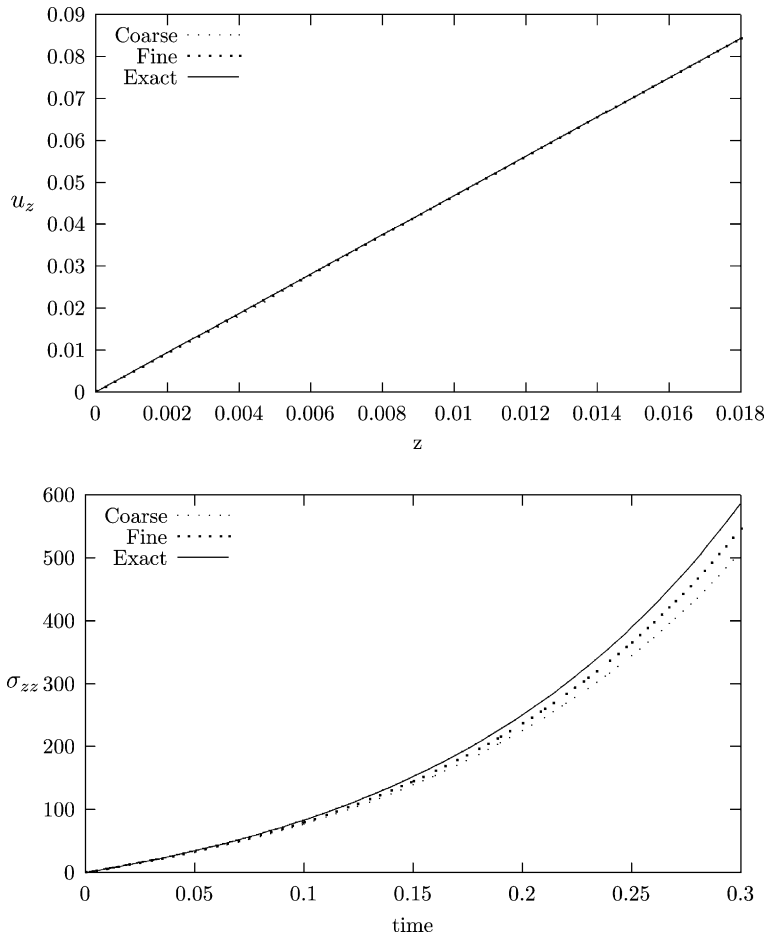


Fig. 17. Elongational flow; top: vertical velocity u_z along the vertical axis Oz at final time $t = 0.3$ s; bottom: extra-stress σ_{zz} at $z = 0.0006$ m as a function of time.

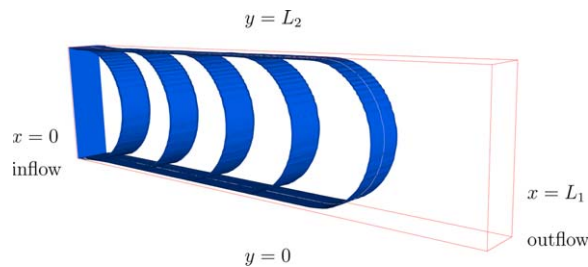


Fig. 18. Filling of a pipe; notations and isovalue $\phi = 0.5$ for a Newtonian fluid at times $t = 0, 0.6, 1.2, 1.8, 2.4, 3.0$ s.

Two meshes are used for the computations. The computational domain is the block $[-0.004 \text{ m}, 0.004 \text{ m}] \times [-0.004 \text{ m}, 0.004 \text{ m}] \times [0 \text{ m}, 0.03 \text{ m}]$ in the xyz directions. The 3D meshes are obtained by extruding the 2D meshes shown in Fig. 15, from $z = 0$ to $z = 0.03 \text{ m}$, and then cutting the prisms into tetrahedrons. The coarse (resp. fine) mesh has 62,000 (resp. 462,000) vertexes and mesh size 0.00035 m (resp. 0.000175 m). When using the coarse (resp. fine) mesh, the cell size is 0.0001 m (resp. 0.00005 m). The time step was $\Delta t = 0.01 \text{ s}$ for the coarse mesh (resp. $\Delta t = 0.005 \text{ s}$ for the fine mesh) so that the CFL number of the cells – velocity times the time step divided by the cells spacing – equals 0.9 at time $t = 0$ and 3.7 at time $t = 0.3$.

Numerical results corresponding to 0.05% by weight Polystyrene (the parameter values are taken from [8], $\rho = 1030 \text{ kg/m}^3$, $\eta_s = 9.15 \text{ Pa s}$, $\eta_p = 25.8 \text{ Pa s}$, $\lambda = 0.421 \text{ s}$, thus $De = \lambda \dot{\epsilon}_0 = 1.97$) are reported in Figs. 16 and 17. Clearly the computed velocity agrees perfectly with the exact velocity whereas the error for the extra-stress is within 10% on the fine grid. The fact that the velocity is more precise than the extra-stress is not surprising since the finite element method is expected to be of order two (in the L^2 norm and in a fixed domain) for the velocity but only of order one for the extra-stress.

5.2. Filling of a pipe

Consider a rectangular pipe of dimensions $[0, L_1] \times [0, L_2] \times [0, L_3]$ in the xyz directions, where $L_1 = 4 \text{ m}$, $L_2 = 1 \text{ m}$, $L_3 = 0.3 \text{ m}$. At the initial time, the pipe is empty. Then, fluid enters from the left side ($x = 0$) with velocity and extra-stress given by

$$\mathbf{u}(x, y, z, t) = \begin{pmatrix} u_x \\ 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{\sigma}(x, y, z, t) = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & 0 \\ \sigma_{xy} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (16)$$

with $u_x(y) = 6y(L_2 - y)$, $\sigma_{xx}(y) = 72\eta_s\lambda(2y - L_2)^2$ and $\sigma_{xy}(y) = -6\eta_p(2y - L_2)$. The boundary conditions are detailed in Fig. 18 and are the following. On the top and bottom sides ($y = 0$ and $y = L_2$), no-slip boundary conditions apply. On the front and rear sides ($z = 0$ and $z = L_3$), slip boundary conditions apply. On the right side ($x = L_1$) the fluid is free to exit the pipe with zero vertical velocity. The parameter values are taken from [40] subsection 6.1 and are the following: $\rho = 1 \text{ kg/m}^3$, $\eta_s = \eta_p = 0.5 \text{ Pa s}$. Three finite element meshes are used in this subsection, see Table 1 and Fig. 19 for details. The cells spacing is five times smaller than the finite element mesh spacing.

Table 1
Filling of a pipe; the three mesh used to check convergence

Mesh	Subdivisions	Vertexes	Tetrahedrons
Coarse	$40 \times 10 \times 3$	1804	7200
Intermediate	$80 \times 20 \times 6$	11,900	57,600
Fine	$160 \times 40 \times 12$	85,813	460,800

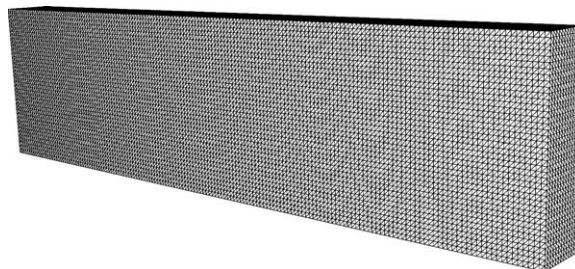


Fig. 19. Filling of a pipe; fine mesh.

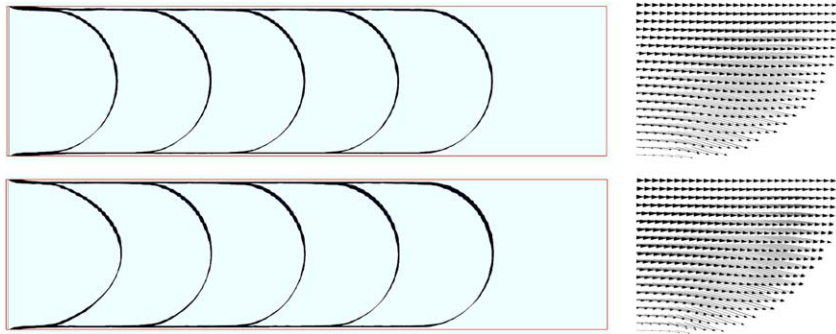


Fig. 20. Filling of a pipe. Left: position of the free surface at time $t = 0, 0.6, 1.2, 1.8, 2.4, 3.0$ s. Right: Velocity field close to the free surface at time $t = 1.8$ s. Top: Newtonian flow. Bottom: viscoelastic flow ($\lambda = 5$ s thus $De = 5$).

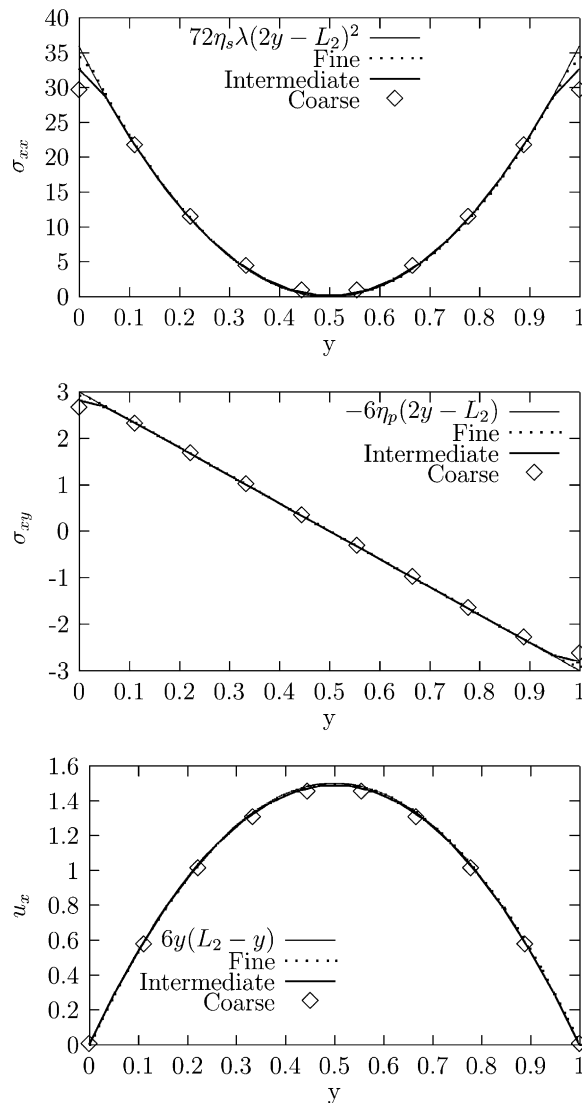


Fig. 21. Filling of a pipe; top: σ_{xx} along the vertical line $x = L_1/2, 0 \leq y \leq L_2, z = L_3/2$, middle: σ_{xy} , bottom: horizontal velocity u_x .

We first consider the filling of the pipe, starting from an empty pipe. This experiment has been considered in [25,40] and is sometimes called fountain flow. The imposed velocity and extra-stress profile at the inlet are those corresponding to Poiseuille flow, see (16). Following [12], after some time the shape of free surface should be close to a half circle. In Fig. 18, the velocity and the shape of the free surface is shown at several times for a Newtonian flow. The mesh is the finest one and the time step is $\Delta t = 0.03$ s, so that the CFL number of the cells – velocity times the time step divided by the cells spacing – equals 4.5. Away from the inlet, the position of the free surface is the same for both Newtonian and viscoelastic ($\lambda = 5$ s) flows, see Fig. 20. As predicted theoretically [12], the shape is almost circular. Details of the fountain flow at the free surface is provided in Fig. 20.

Once totally filled with liquid, the velocity and extra-stress must satisfy (16) in the whole pipe. Convergence of the stationary solution is checked with $\lambda = 1$ s, thus $De = \lambda U/L_2 = 1$, where $U = 1$ m/s is the average velocity. In Fig. 21, σ_{xx} , σ_{xy} and u_x are plotted along the vertical line $x = L_1/2$, $0 \leq y \leq L_2$, $z = L_3/2$. Convergence can be observed even though boundary layer effects are present, this being classical with low order finite elements. In Fig. 22, the error in the L^2 norm of σ_{xx} , σ_{xy} and u_x is plotted versus the mesh size. Clearly order one convergence rate is observed for the extra-stress, order two for the velocity, this being consistent with theoretical predictions on simpler problems [3].

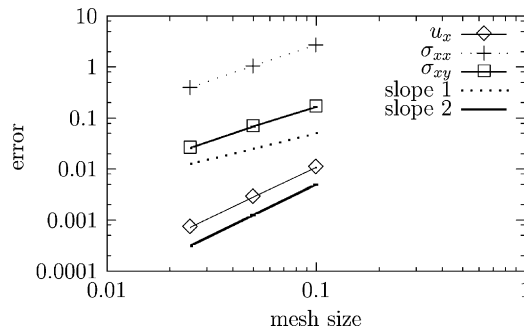


Fig. 22. Filling of a pipe; error in the L^2 norm with respect to the mesh size.

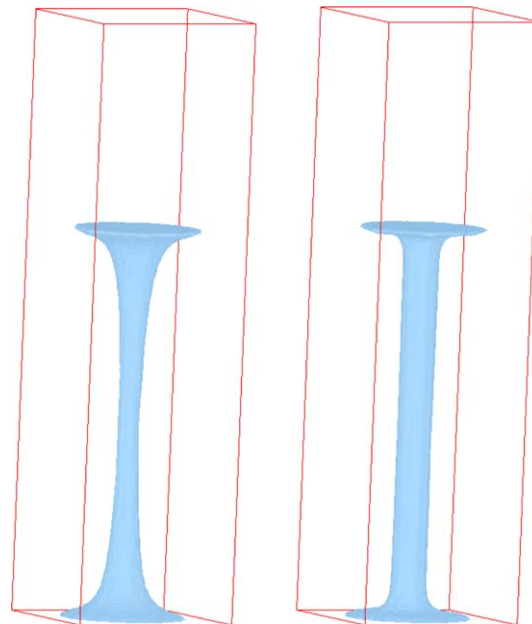


Fig. 23. Filament stretching. Aspect ratio $L_0/R_0 = 19/30$. Shape of the liquid region at time $t = 0.5$ (the Hencky strain is $\epsilon = \dot{\epsilon}_0 t = 2.34$); left: Newtonian fluid; right: $\lambda = 0.421$ s ($De = 1.97$).

5.3. Stretching of a filament

The flow of an Oldroyd-B fluid contained between two parallel coaxial circular disks with radius $R_0 = 0.003$ m is considered. At the initial time, the distance between the two end-plates is $L_0 = 0.0019$ m and the liquid is at rest. Then, the top end-plate is moved vertically with velocity $L_0 \dot{\epsilon}_0 e^{\dot{\epsilon}_0 t}$. The model data ($\rho, \eta_s, \eta_p, \lambda, \dot{\epsilon}_0$) are those of Section 5.1. The fine mesh of Section 5.1 was used with an initial time step $\Delta t^0 = 0.005$ s, thus the initial CFL number of the cells – velocity times the time step divided by the cells spacing – is close to one, the time step at time t^n being such that the distance of the moving end-plate between two time steps is constant, that is

$$\Delta t^n = \Delta t^{n-1} e^{-\dot{\epsilon}_0 \Delta t^{n-1}}.$$

Therefore, the CFL number remains constant throughout the simulation. The shape of the liquid region at time $t = 0.5$ s is represented in Fig. 23, for both Newtonian and non-Newtonian computations. 2D cuts along plane $y = 0$ are shown in Fig. 24. As reported in [43], the ‘necking’ phenomena occurring in the central part of the liquid for Newtonian fluids is not observed for viscoelastic fluids, due to strain hardening. This calculation requires 2 h (resp. 24 h) on the coarse mesh (resp. fine mesh) using a single user Pentium 4 CPU 2.8 GHz, with 2Gb memory, under the Linux operating system. Most of the time is spent in solving the associated Stokes problem. The memory usage is 200 Mb for the coarse mesh, resp. 1.6 Gb for the fine mesh.

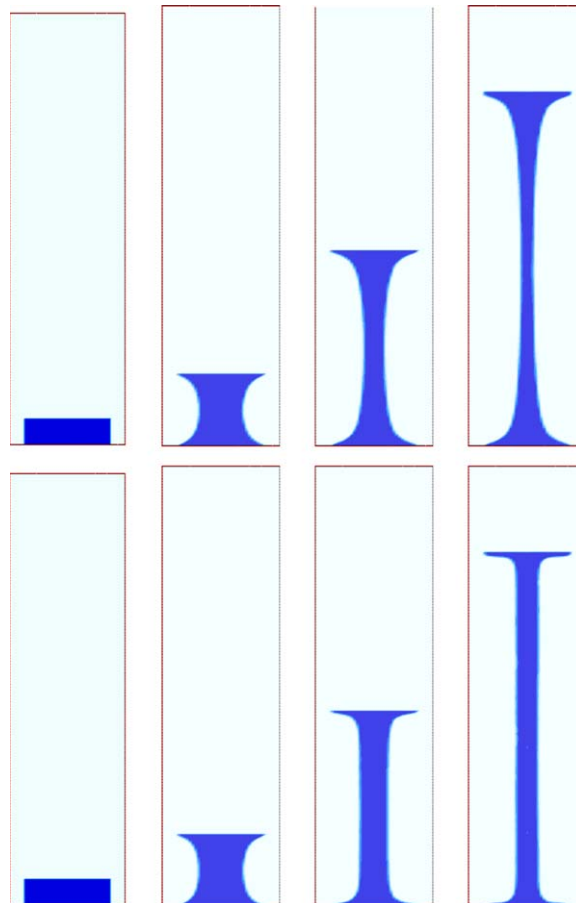


Fig. 24. Filament stretching. Aspect ratio $L_0/R_0 = 19/30$. Shape of the liquid region in the $y = 0$ plane (the isovalues of φ are shown); column 1: $\epsilon = \dot{\epsilon}_0 t = 0$; column 2: $\epsilon = \dot{\epsilon}_0 t = 0.96$; column 3: $\epsilon = \dot{\epsilon}_0 t = 1.96$; column 4: $\epsilon = \dot{\epsilon}_0 t = 2.55$; top row: Newtonian fluid; bottom row: $\lambda = 0.421$ s ($De = 1.97$).

Unfortunately, we have not been able to reach high Hencky strains as reported in [43,8] for 2D axisymmetric computations. The reasons may be the following: (i) When the Hencky strain is large, the filament is highly stretched and the number of vertexes in the thinnest region of the filament decreases, and so does the accuracy. Therefore, Lagrangian numerical models should be more accurate than Eulerian ones provided the mesh is not too distorted. (ii) Surface tension, which should stabilize the filament shape, is not included in our model. (iii) Since the filament breaking is due to 3D instabilities, 2D axisymmetric computations should be more stable than 3D computations. At this point, it should be mentioned that in [29], the authors have also reported the same discrepancies when comparing results output by their 3D Lagrangian model with experiments [38]. Moreover, from Fig. 11 of [29], it is predicted that the onset of instability when $De = 2$ is obtained for Hencky strains $\epsilon \approx 2$. This is in accordance with the results of Fig. 24.

We now show that our numerical model is capable to reproduce fingering instabilities reported in [29,2,21,9] for non-Newtonian flows. Following Section 4.4 in [21], we take an aspect ratio $L_0/R_0 = 1/20$ ($R_0 = 0.003$ m,

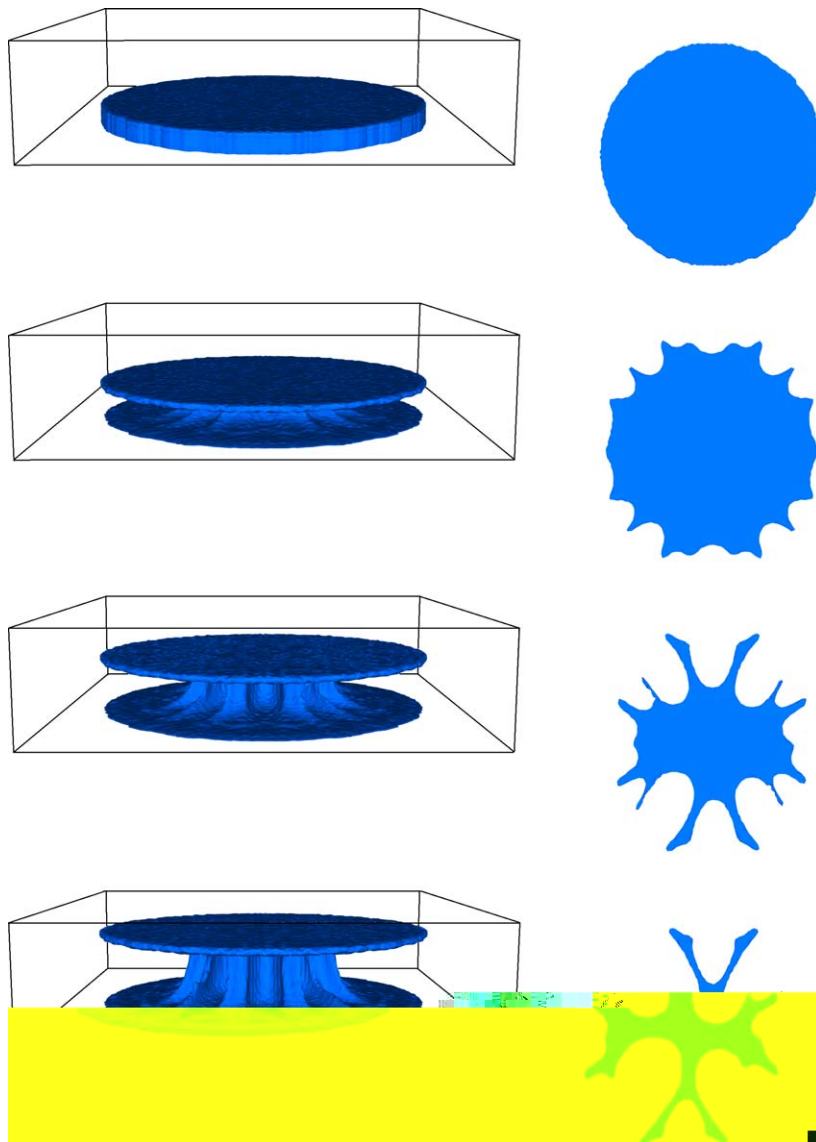


Fig. 25. Filament stretching, $\lambda = 0.421$ s ($De = 1.97$), aspect ratio $L_0/R_0 = 1/20$. Left: shape of the liquid region at time $t = 0$ s, $t = 0.33$ s, $t = 0.66$ s and $t = 1$ s. Right: horizontal cut at the middle of the liquid region.

$L_0 = 0.00015$ m), so that the Weissenberg number $We = DeR_0^2/L_0^2$ is large. The finite element mesh has 50 vertexes along the radius and 25 vertexes along the height, thus the mesh size is 0.00006 m. The cells size is 0.00001 m and the initial time step is $\Delta t^0 = 0.01$ s thus the CFL number of the cells – velocity times the time step divided by the cells spacing – is close to one. The shape of the filament is reported in Fig. 25 and fingering instabilities can be observed from the very beginning of the stretching, leading to branched structures, as described in [21,2,9]. Clearly, such complex shapes cannot be obtained using Lagrangian models, the mesh distortion would be too large. In Fig. 26, the mesh is changed in order to check the dependence of the results with respect to the mesh topology. In Fig. 27, the same simulation is performed for a Newtonian fluid and no fingering instabilities can be seen. We therefore conclude that these instabilities are essentially elastic, as reported in [29]. However, it should be noted that fingering instabilities can also be obtained for Newtonian flows, see for instance [36].

5.4. Jet buckling

The transient flow of a jet injected into a parallelepiped cavity is now reproduced. First, our 3D computations are compared to the 2D results reported in [40, Section 7.3]. Then, 3D computations are shown.

In order to compare our 3D computations to those of [41,40], we consider a thin cavity of width 0.05 m, variable height H and depth 0.004 m, the width of the jet being $D = 0.005$ m and the vertical gravity $g = 9.81$ m/s². Slip boundary conditions apply whenever the jet hits the boundary of the cavity. The finite element mesh can be

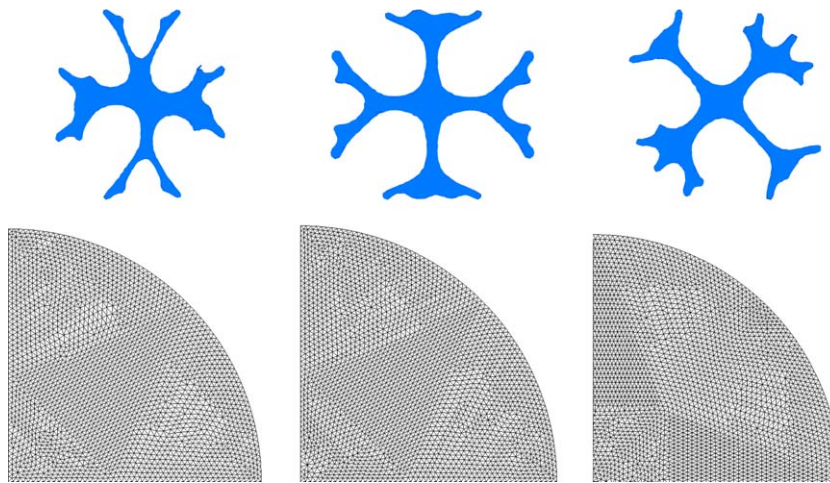


Fig. 26. Filament stretching, $\lambda = 0.421$ s ($De = 1.97$), aspect ratio $L_0/R_0 = 1/20$, time $t = 1$ s. Horizontal cut at the middle of the liquid region. Left: the mesh has 200 vertexes along the diameter. Middle: the mesh has 204 vertexes along the diameter. Right: the middle mesh is rotated by $\pi/4$.

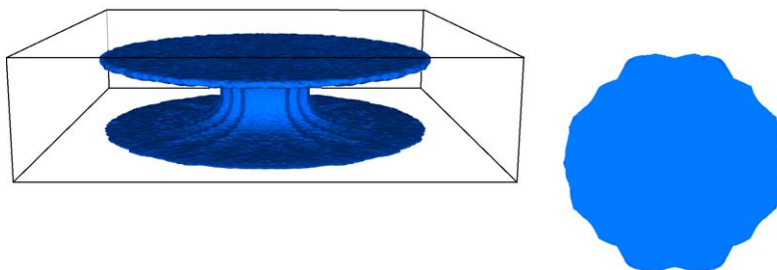
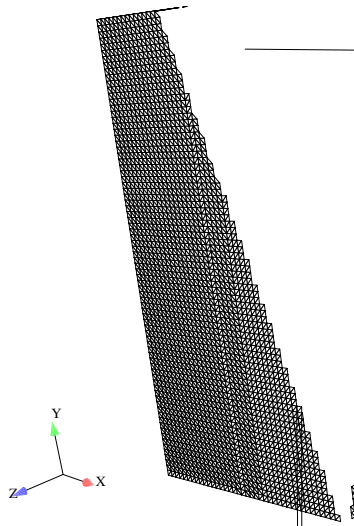


Fig. 27. Filament stretching, Newtonian fluid, aspect ratio $L_0/R_0 = 1/20$. Left: shape of the liquid region at time $t = 1$ s. Right: horizontal cut at the middle of the liquid region. Compare with Fig. 25.

se
er
tl
ii

V
1
1
1
1
1

11/17/2019. Secondly, we determine the maximum Reynolds number for which buckling occurs. The results are reported in Fig. 29. The jet buckles



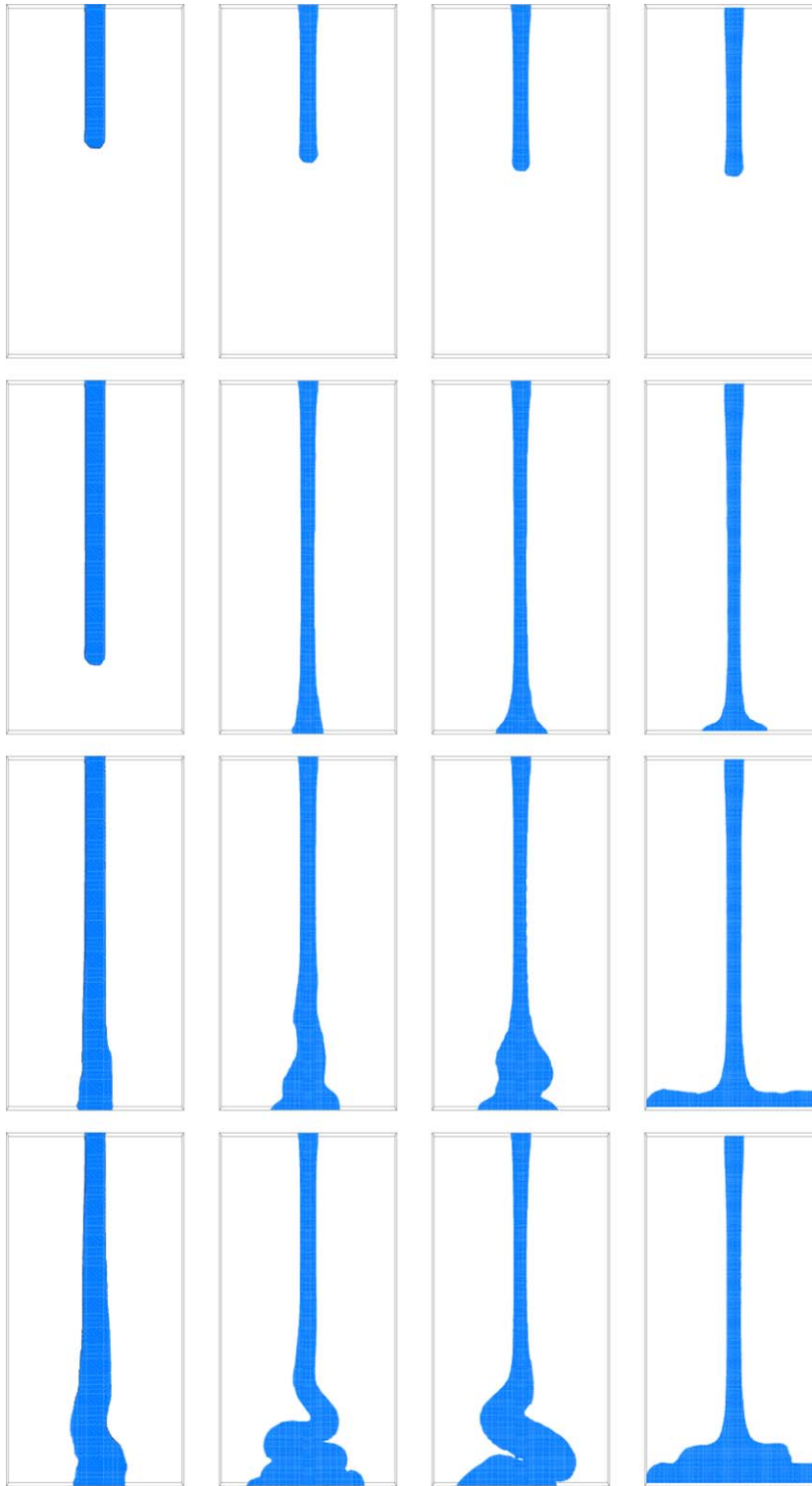


Fig. 30. Jet buckling in a thin cavity. Shape of the jet at time $t = 0.1$ s (first row), $t = 0.2$ s (second row), $t = 0.3$ s (third row), $t = 0.4$ s (last row), Newtonian flow (first column), $\lambda = 0.01$ s (second column), $\lambda = 0.1$ s (third column), $\lambda = 1$ s (last column).

when $Re = 0.2$ and $Re = 0.5$ but does not buckle when $Re = 0.7$. This is again consistent with condition (17) which yields buckling whenever $Re \leq 0.53$. Finally, we set $Re = 1$, $H/D = 20$ so that no buckling occurs in the Newtonian case and perform a viscoelastic computation with $\eta_s = 1.03$, $\eta_p = 9.27$ Pa s, $\lambda = 0.1$ s. The result is

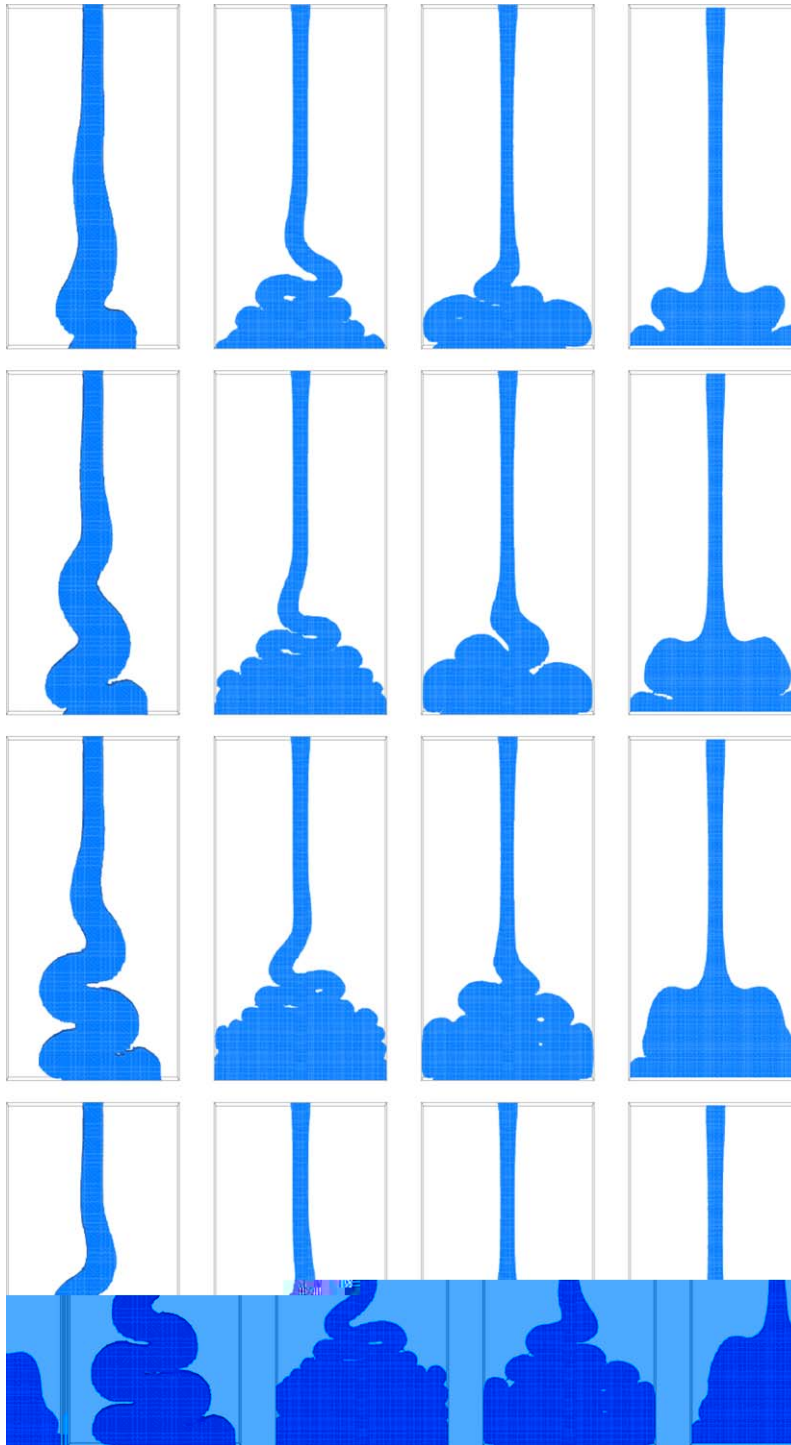


Fig. 31. Jet buckling in a thin cavity. Shape of the jet at time $t = 0.5$ s (first row), $t = 0.6$ s (second row), $t = 0.7$ s (third row), $t = 0.8$ s (last row), Newtonian flow (first column), $\lambda = 0.01$ s (second column), $\lambda = 0.1$ s (third column), $\lambda = 1$ s (last column).

shown in Fig. 29, right and obviously buckling occurs. Other computations show that the jet buckles whenever $\lambda < 0.005$ s. We therefore conclude that our numerical model yields results which agree with condition (17) in the Newtonian case and that this condition depends on λ for viscoelastic flows.

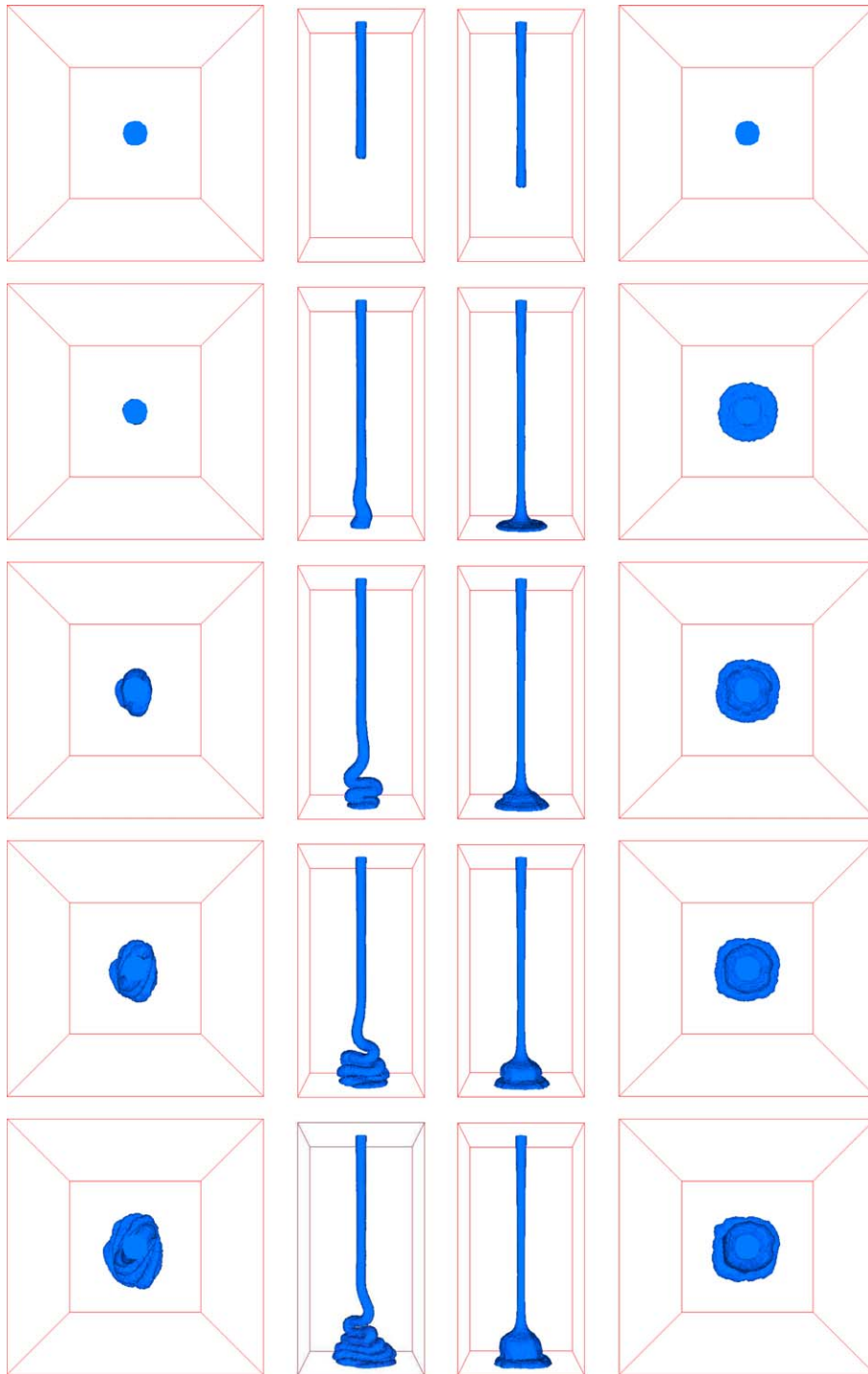


Fig. 32. Jet buckling in a thick cavity. Shape of the jet at time $t = 0.1$ s (row 1), $t = 0.2$ s (row 2), $t = 0.3$ s (row 3), $t = 0.4$ s (row 4), $t = 0.5$ s (row 5), Newtonian fluid (col. 1 and 2), $\lambda = 1$ s (col. 3 and 4).

We then compare our 3D viscoelastic computations to those 2D of [40], with $H = 0.1$ m ($H/D = 20$) and $U = 0.5$ m/s ($Re = 0.25$), λ ranges from 0 to 1 s so that $De = \lambda U/D$ ranges from 0 to 100. The shape of the jet is shown in Figs. 30 and 31. As in [40], when the Newtonian jet starts to buckle, the non-Newtonian jet

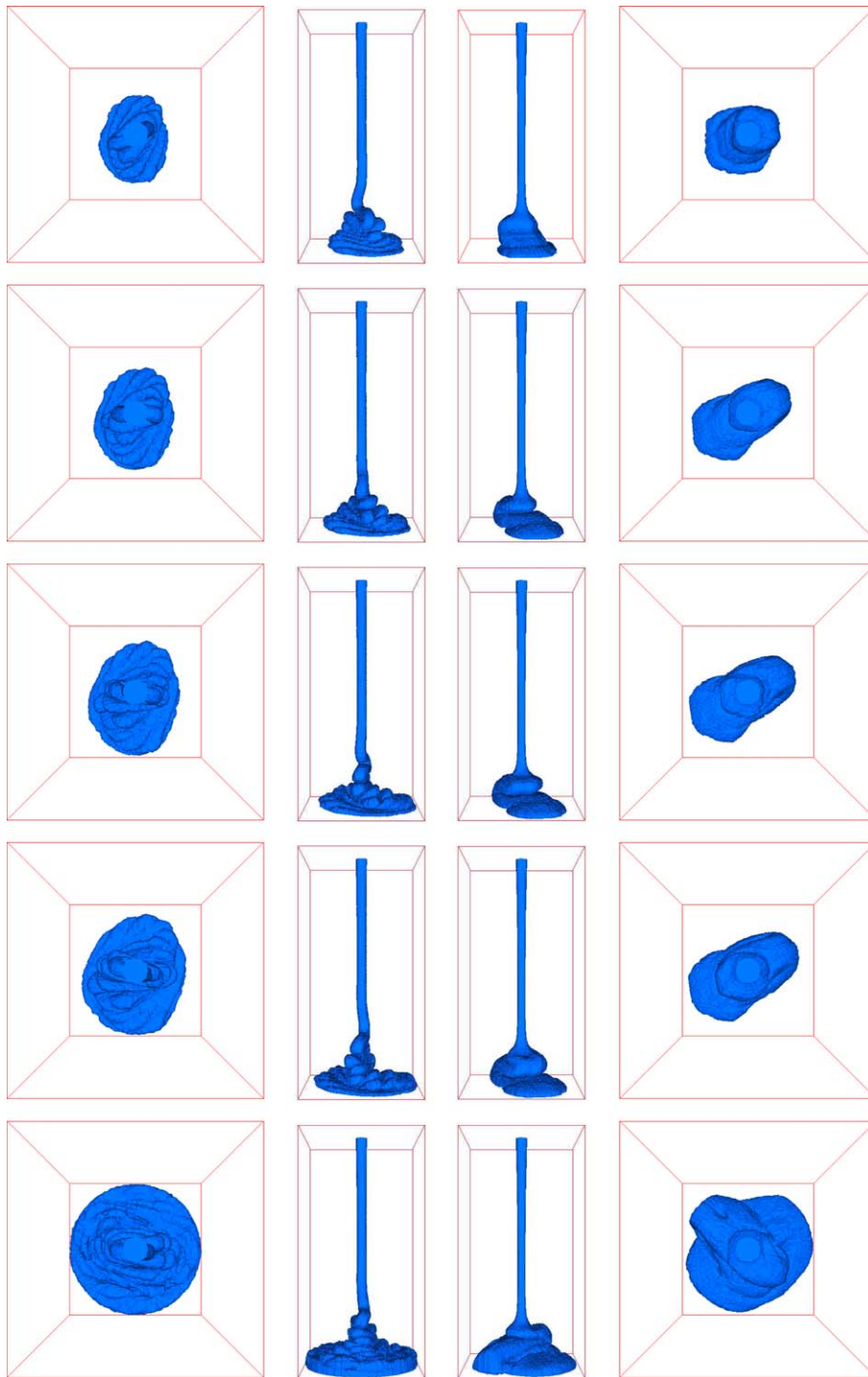


Fig. 33. Jet buckling in a thick cavity. Shape of the jet at time $t = 0.6$ s (row 1), $t = 0.7$ s (row 2), $t = 0.8$ s (row 3), $t = 0.9$ s (row 4), $t = 1.6$ s (row 5), Newtonian fluid (col. 1 and 2), $\lambda = 1$ s (col. 3 and 4).

has already produced many folds. However, we did not observe during the whole experiment a thin viscoelastic jet as in [40] (time $t = 0.3$ s).

Simulations are now reported when the cavity is a thick cavity of width 0.05 m, depth 0.05 m and height 0.1 m, the diameter of the jet being $D = 0.005$ m. Liquid enters from the top of the cavity with vertical velocity $U = 0.5$ m/s. The relaxation time $\lambda = 1$ s so that $De = 100$. The finite element mesh has 503,171 vertexes and 2,918,760 tetrahedrons. The cells size is 0.0002 m and the time step is 0.001 s thus the CFL number of the cells – velocity times the time step divided by the cells spacing – is 2.5. The shape of the jet is shown in Figs. 32 and 33 for Newtonian and viscoelastic flows. This computation took 64 h on a AMD opteron CPU with 8Gb memory.

6. Conclusions

An Eulerian model based on the VOF formulation has been presented for the simulation of viscoelastic flows with complex free surfaces in three space dimensions.

A splitting method is used for time discretization. The prediction step consists in solving three advection problems, one for the volume fraction of liquid, one for the velocity field, one for the extra-stress. The correction step corresponds to solving an Oldroyd-B flow problem without advection.

Two different grids are used for space discretization. The three advection problems are solved on a fixed, structured grid made out of small cubic cells, using a forward characteristics method. The viscoelastic flow problem without advection is solved using continuous, piecewise linear stabilized finite elements on a fixed, unstructured mesh of tetrahedrons.

Convergence of the numerical method is checked for two test cases, namely an elongational flow and the filling of a pipe. Numerical results are then presented in the framework of the stretching of a filament. When the aspect ratio is large, fingering instabilities are obtained. Jet buckling is also studied.

Acknowledgement

Vincent Maronnier, Calcom ESI Company, Parc Scientifique, CH-1015 Lausanne, is acknowledged for implementation support.

References

- [1] E. Aulisa, S. Manservigi, R. Scardovelli, A mixed markers and volume-of-fluid method for the reconstruction and advection of interfaces in two-phase and free-boundary flows, *J. Comput. Phys.* 188 (2) (2003) 611–639.
- [2] A. Bach, H.K. Rasmussen, P.-Y. Longin, O. Hassager, Growth of non-axisymmetric disturbances of the free surface in the filament stretching rheometer: experiments and simulation, *J. Non-Newtonian Fluid Mech.* 108 (2002) 163–186.
- [3] J. Bonvin, M. Picasso, R. Stenberg, GLS and EVSS methods for a three-field Stokes problem arising from viscoelastic flows, *Comput. Methods Appl. Mech. Engrg.* 190 (29–30) (2001) 3893–3914.
- [4] A. Caboussat, Numerical simulation of two-phase free surface flows, *Arch. Comput. Methods Engrg., State of the art reviews* 12 (2) (2005) 165–224.
- [5] A. Caboussat, M. Picasso, J. Rappaz, Numerical simulation of free surface incompressible liquid flows surrounded by compressible gas, *J. Comput. Phys.* 203 (2) (2005) 626–649.
- [6] Y.C. Chang, T.Y. Hou, B. Merriman, S. Osher, A level set formulation of Eulerian interface capturing methods for incompressible fluid flows, *J. Comput. Phys.* 124 (2) (1996) 449–464.
- [7] A.J. Chorin, Flame advection and propagation algorithms, *J. Comput. Phys.* 35 (1980) 1–11.
- [8] J. Cormenzana, A. Ledd, M. Laso, B. Debbaut, Calculation of free surface flows using CONNFFESSIT, *J. Rheol.* 45 (1) (2001) 237–258.
- [9] D. Derks, A. Lindner, C. Creton, D. Bonn, Cohesive failure of thin layers of soft model adhesives under tension, *J. Appl. Phys.* 93 (3) (2003) 1557–1566.
- [10] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A hybrid particle level set method for improved interface capturing, *J. Comput. Phys.* 183 (1) (2002) 83–116.
- [11] T. Goktekin, A. Bargteil, J. O'Brien, A method for animating viscoelastic fluids, *ACM Trans. Graphics (Proc. of ACM SIGGRAPH 2004)* 23 (3) (2004) 463–468.
- [12] H.J.J. Gramberg, J.C.W. van Vroonhoven, A.A.F. van de Ven, Flow patterns behind the free flow front for a Newtonian fluid injected between two infinite parallel plates, *Eur. J. Mech. B Fluids* 23 (4) (2004) 571–585.

- [13] E. Grande, M. Laso, M. Picasso, Calculation of variable-topology free surface flows using CONNFFESSIT, *J. Non-Newtonian Fluid Mech.* 113 (2003) 127–145.
- [14] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, S. Zaleski, Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows, *J. Comput. Phys.* 152 (2) (1999) 423–456.
- [15] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *J. Comput. Phys.* 39 (1981) 201–225.
- [16] J. Li, Y. Renardy, Shear-induced rupturing of a viscous drop in a Bingham liquid, *J. Non-Newtonian Fluid Mech.* 95 (2–3) (2000) 235–251.
- [17] D. Lorstad, L. Fuchs, High-order surface tension vof-model for 3d bubble flows with high density ratio, *J. Comput. Phys.* 200 (1) (2004) 153–176.
- [18] V. Maronnier, Simulation numérique d'écoulements de fluides incompressibles avec surface libre, Ph.D. Thesis, Département de Mathématiques, Ecole Polytechnique Fédérale de Lausanne, 2000.
- [19] V. Maronnier, M. Picasso, J. Rappaz, Numerical simulation of free surface flows, *J. Comput. Phys.* 155 (2) (1999) 439–455.
- [20] V. Maronnier, M. Picasso, J. Rappaz, Numerical simulation of 3d free surface flows, *Int. J. Numer. Methods Fluids* 42 (7) (2003) 697–716.
- [21] G.H. McKinley, T. Sridhar, Filament-stretching rheometry of complex fluids, *Annu. Rev. Fluid Mech.* 34 (2002) 375–415.
- [22] W.F. Noh, P. Woodward, SLIC (Simple Line Interface Calculation) Lectures Notes in Physics, vol. 59, Springer-Verlag, 1976, pp. 330–340.
- [23] T.N. Phillips, A.J. Williams, Viscoelastic flow through a planar contraction using a semi-lagrangian finite volume method, *J. Non-Newtonian Fluid Mech.* 87 (2–3) (1999) 215–246.
- [24] M. Picasso, J. Rappaz, Existence, a priori and a posteriori error estimates for a nonlinear three-field problem arising from Oldroyd-B viscoelastic flows, *M2AN Math. Model. Numer. Anal.* 35 (5) (2001) 879–897.
- [25] E. Picheli, T. Coupez, Finite element solution of the 3D mold filling problem for viscous incompressible fluid, *Comput. Methods Appl. Mech. Engrg.* 163 (1–4) (1998) 359–371.
- [26] O. Pironneau, *Finite Element Methods for Fluids*, Wiley, Chichester, 1989.
- [27] O. Pironneau, J. Liou, T. Tezduyar, Characteristic-Galerkin and Galerkin/least-squares space–time formulations for the advection-diffusion equation with time-dependent domains, *Comput. Methods Appl. Mech. Engrg.* 100 (1992) 117–141.
- [28] A. Quarteroni, A. Valli, *Numerical Approximation of Partial Differential Equations* Springer Series in Computational Mathematics, 23, Springer-Verlag, 1991.
- [29] H.K. Rasmussen, O. Hassager, Three-dimensional simulations of viscoelastic instability in polymeric filaments, *J. Non-Newtonian Fluid Mech.* 82 (1999) 189–202.
- [30] M. Renardy, Existence of slow steady flows of viscoelastic fluids of integral type, *Z. Angew. Math. Mech.* 68 (4) (1988) T40–T44.
- [31] Y. Renardy, M. Renardy, PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method, *J. Comput. Phys.* 183 (2) (2002) 400–421.
- [32] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, *J. Comput. Phys.* 141 (2) (1998) 112–152.
- [33] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, in: *Annual Review of Fluid Mechanics*, vol. 31, *Ann. Rev. Fluid Mech.*, pp. 567–603, Annual Reviews, Palo Alto, CA, 1999.
- [34] J.A. Sethian, *Level Set Methods*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 1996.
- [35] J.A. Sethian, P. Smereka, Level set methods for fluid interfaces, in: *Annual Review of Fluid Mechanics*, vol. 35, *Ann. Rev. Fluid Mech.*, pp. 341–372, Annual Reviews, Palo Alto, CA, 2003.
- [36] M.J. Shelley, F.-R. Tian, K. Wlodarski, Hele-Shaw flow and pattern formation in a time-dependent gap, *Nonlinearity* 10 (6) (1997) 1471–1495.
- [37] R. Sizaire, V. Legat, Finite element simulation of a filament stretching extensional rheometer, *J. Non-Newtonian Fluid Mech.* 71 (1–2) (1997) 89–107.
- [38] S.H. Spiegelberg, G.H. McKinley, Stress relaxation and elastic decohesion of viscoelastic polymer solutions in extensional flow, *J. Non-Newtonian Fluid Mech.* 77 (1996) 49–76.
- [39] E. Thompson, Use of pseudo-concentrations to follow creeping viscous flows during transient analysis, *Int. J. Numer. Methods Fluids* 6 (1986) 749–761.
- [40] M.F. Tomé, N. Mangiavacchi, J.A. Cuminato, A. Castelo, S. McKee, A finite difference technique for simulation unsteady viscoelastic free surface flows, *J. Non-Newtonian Fluid Mech.* 106 (2002) 61–106.
- [41] M.F. Tomé, S. McKee, Numerical simulation of viscous flow: buckling of planar jets, *Int. J. Numer. Methods Fluids* 29 (1999) 705–718.
- [42] S.P. van der Pijl, A. Segal, C. Vuik, P. Wesseling, A mass-conserving level-set method for modelling of multi-phase flows, *Int. J. Numer. Methods Fluids* 47 (4) (2005) 339–361.
- [43] M. Yao, G.H. McKinley, Numerical simulation of extensional deformations of viscoelastic liquid bridges in filament stretching devices, *J. Non-Newtonian Fluid Mech.* 74 (1–3) (1998) 47–88.