

# A semi-Lagrangian micro–macro method for viscoelastic flow calculations

Juan Luis Prieto<sup>a,c,\*</sup>, Rodolfo Bermejo<sup>c</sup>, Manuel Laso<sup>a,b</sup>

<sup>a</sup> Institute for Optoelectronics and Microsystems (ISOM), Universidad Politécnica de Madrid, Ciudad Universitaria, 28040 Madrid, Spain

<sup>b</sup> Department of Chemical Engineering, ETSII, Universidad Politécnica de Madrid, José Gutiérrez Abascal 2, 28006 Madrid, Spain

<sup>c</sup> Department of Applied Mathematics, ETSII, Universidad Politécnica de Madrid, José Gutiérrez Abascal 2, 28006 Madrid, Spain

## ARTICLE INFO

**Keywords:**  
Semi-Lagrangian scheme  
Micro–macro  
Characteristics  
Finite element

## ABSTRACT

We present in this paper a semi-Lagrangian algorithm to calculate the viscoelastic flow in which a dilute polymer solution is modeled by the FENE dumbbell kinetic model. In this algorithm the material derivative operator of the Navier–Stokes equations (the macroscopic flow equations) is discretized in time by a semi-Lagrangian formulation of the second order backward difference formula (BDF2). This discretization leads to solving each time step a linear generalized Stokes problem. For the stochastic differential equations of the microscopic scale model, we use the second order predictor–corrector scheme proposed in [22] applied along the forward trajectories of the center of mass of the dumbbells. Important features of the algorithm are (1) the new semi-Lagrangian projection scheme; (2) the scheme to move and locate both the mesh-points and the dumbbells; and (3) the calculation and space discretization of the polymer stress. The algorithm has been tested on the 2d 10:1 contraction benchmark problem and has proved to be accurate and stable, being able to deal with flows at high Weissenberg ( $Wi$ ) numbers; specifically, by adjusting the size of the time step we obtain solutions at  $Wi = 444$ .

## 1. Introduction

Micro–macro models for polymeric flows couple the continuum mechanics equations for conservation of mass and momentum (macro-scale) to the equations derived from the kinetic theory via a coarse-grained description of the molecular configurations (micro-scale). Numerical algorithms directly based on the kinetic theory for complex fluids have been developed since the 1970s, but due to the fact that the simulations are very demanding from a computational standpoint, they have become available only after the development of supercomputers. For the deterministic version of the micro–macro models one needs to solve a very high dimensional partial differential equation system in both position and configuration spaces. As another candidate approach, stochastic simulations have the virtue of being dimension-independent and parallel in nature, with the additional advantage that no ad hoc constitutive equation for the polymer stress is needed to close the model. Therefore, this is a very promising approach for multiscale simulations of complex configurations such as polymer chains. Laso and Öttinger [1] seem to be the first in designing a numerical algorithm for stochastic simulations of polymer flows, the so-called *CONNFESSIT* idea (Calculations Of Non-Newtonian Flow: Finite

Element and Stochastic Simulation Technique). Applications of this approach to dumbbell models of dilute polymer solutions are reported in [2] for two-dimensional stationary flows and [3,4] for two-dimensional non-stationary flows, for liquid crystalline polymers in [5] and for free surface flows in [6] and [7]. The advantages of such a method have had an important impact in the development of numerical micro–macro models; however, as recognized by [4,8], there are still some difficulties, such as the tracking of the “molecules” and the calculation of the polymer stress tensor, that have to be addressed to improve the efficiency and accuracy of the numerical calculations. A step forward in that direction is represented by the technique of Brownian configuration fields of Hulsen et al. [8], the adaptive configuration fields of Gigras and Khomami [9], and by the Lagrangian particle method (LPM) of Halin et al. [10] and Wapperom et al. [11]. (For additional details, see e.g., the revision by Keunings on micro–macro methods [12]).

In the present paper, we improve previous micro–macro approaches [2–7] by implementing new schemes at both macro and micro levels. For the macro level, we use a second order in time backward difference formula (hereafter, BDF2) semi-Lagrangian scheme in combination with P2/P1/P1 finite elements for space discretization; that is, the velocity is calculated with quadratic finite elements whereas the pressure and the polymer stress are both calculated with linear polynomials. As for the micro level, (1) we calculate more accurately and more efficiently the trajectories of the individual dumbbells by implementing a second order in time adaptive scheme which prevents the dumbbells from leaving the flow domain through impervious walls; and (2) we consider that

\* Corresponding author at: Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros Industriales, Jose Gutiérrez Abascal, no. 2, 28006 Madrid, Spain. Tel.: +34 913364261; fax: +34 913363001.

E-mail addresses: juanluis.prieto@upm.es (J.L. Prieto), rbermejo@etsii.upm.es (R. Bermejo), mlaso@etsii.upm.es (M. Laso).

the polymer stress is a piecewise linear continuous function whose point values are calculated by applying Kramers' expression for the polymer stress on the finite volume associated to each vertex of the elements. Semi-Lagrangian schemes to deal with the convective terms are not new in the realm of viscoelastic flows, in particular in numerical integrations of models based on the continuum approach, because several researchers such as [13–16], just to cite a few, have used semi-Lagrangian schemes or schemes based on the backward method of characteristics; however, to the best of our knowledge, this is the first time that the micro–macro approach is combined with semi-Lagrangian schemes. The result is a robust, accurate and highly stable method that is able to perform numerical simulations at high Weissenberg numbers in the 10:1 contraction flow example.

The layout of the paper is as follows: in Section 2 we formulate the micro–macro FENE model equations whose dimensionless version is presented in Section 3. Section 4 is devoted to the description of the numerical methods. The results of the numerical tests for the 10:1 contraction flow are reported and commented in Section 5. Conclusions are set in Section 6.

## 2. The micro–macro model governing equations

In this paper we focus on two-dimensional viscoelastic incompressible flows modeled by the kinetic theory approach: specifically, we shall use FENE dumbbell model to account for the dynamics effects at the micro-scale. In this context, a dumbbell is thought of as an object composed of two Brownian beads connected by an entropic spring. We consider the flow of a constant density diluted solution of polymeric liquid in a bounded domain  $D \subset \mathbb{R}^2$  with boundary  $\Gamma$  during a time interval  $[0, T]$ . The boundary  $\Gamma$  is composed of various pieces on which different types of boundary conditions are imposed. Thus,  $\Gamma = \Gamma^i \cup \Gamma^s \cup \Gamma^o$ , where  $\Gamma^i, \Gamma^s, \Gamma^o$  denote inflow, solid and outflow boundaries, respectively, and such that  $\Gamma^i \cap \Gamma^s = \emptyset$ ,  $\Gamma^i \cap \Gamma^o = \emptyset$  and  $\Gamma^s \cap \Gamma^o = \emptyset$ . The model consists of two sets of equations; one set, corresponding to the macro-scale, is formed by mass and momentum conservation equations of the flow; and the other set, corresponding to the micro-scale, is based on the kinetic theory of the molecular configurations to display a more or less accurate description of the flow-induced dynamics upon the non-interacting polymer chains.

### 2.1. Macro-scale equations

For the sake of simplicity, we shall consider no body forces acting on the flow, so that the governing equations are the following:

$$\begin{cases} \rho \frac{D\mathbf{u}}{Dt} + \nabla p = \eta_s \Delta \mathbf{u} + \nabla \cdot \boldsymbol{\tau}_p \text{ in } D \times (0, T], \\ \nabla \cdot \mathbf{u} = 0 \text{ in } D \times (0, T]. \end{cases} \quad (1)$$

These equations are to be solved subject to the initial conditions

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \quad \forall \mathbf{x} \in D \quad (2)$$

and the boundary conditions

$$\begin{aligned} \mathbf{u}(\mathbf{x}, t) &= \mathbf{0} \text{ on } \Gamma^s \quad \forall t, \\ -p\mathbf{n} + \eta_s \frac{\partial \mathbf{u}}{\partial \mathbf{n}} &= \mathbf{g}(\mathbf{x}, t) \text{ on } \Gamma^o \quad \forall t, \\ -\mathbf{n} \cdot \mathbf{u}(\mathbf{x}, t) &= a(\mathbf{x}, t) \text{ on } \Gamma^i \quad \forall t. \end{aligned} \quad (3)$$

Here,  $\mathbf{u}$  and  $p$  denote the flow velocity and the hydrodynamic pressure, respectively;  $\rho$  is the constant density of the solution;  $\mathbf{n}$  is the unitary outward normal vector at the boundary;  $\mathbf{g}$  and  $a$  are known data;  $D/Dt \equiv \partial/\partial t + \mathbf{u} \cdot \nabla$  represents the material derivative operator; the total stress tensor  $\boldsymbol{\tau}$  is defined as  $\boldsymbol{\tau} = \boldsymbol{\tau}_s + \boldsymbol{\tau}_p$ ; the Newtonian stress tensor  $\boldsymbol{\tau}_s = \eta_s (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$ , with  $\eta_s$  being

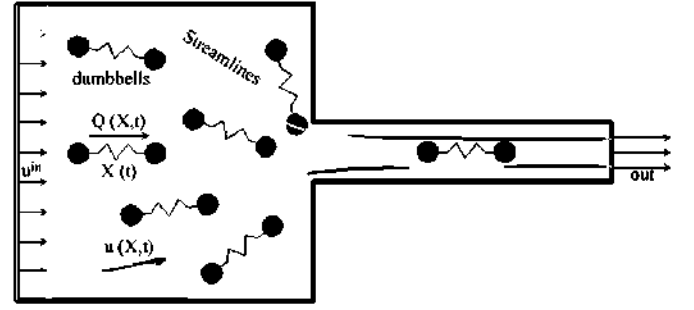


Fig. 1. The 10:1 planar contraction geometry for the 'dumbbells' FENE model.

a constant viscosity coefficient, is the solvent contribution to the stress; and  $\boldsymbol{\tau}_p$  is the polymer contribution to the total stress that results from the anisotropic orientation and stretch of the polymer chains. This contribution is the dynamic link between the micro- and macro-scales, and is calculated by Kramers' relation [17] from the configurations obtained as solutions of the micro-scale model. Note that, since  $\eta_s$  is constant, we have by virtue of the divergence free condition that  $\nabla \cdot \boldsymbol{\tau}_s = \eta_s \Delta \mathbf{u}$ .

### 2.2. Micro-scale equations: the FENE dumbbell model

The polymer solution is modeled as a flowing suspension of non-interacting dumbbells that are transported by the flow. At time  $t$ , each dumbbell is characterized by the position of its center of mass  $\mathbf{X}(t)$  and the elongation vector  $\mathbf{Q}(t)$ , see Fig. 1. By the action of the macroscopic flow, the dumbbells experience three forces: the first one is the drag force due to the differences in velocity between the dumbbell and the surrounding flow particles, the second force is the elastic force due to the spring stiffness, and the third force is the so-called Brownian force due to the thermal agitation that is modeled as a Brownian motion. Using Newton's second law, we have that the following stochastic differential equations govern the dynamics of a dumbbell:

$$d\mathbf{X}(t) = \mathbf{u}(\mathbf{X}(t), t) dt \quad (4)$$

and

$$d\mathbf{Q} = \left[ \kappa(\mathbf{X}(t), t) \cdot \mathbf{Q}(t) - \frac{2}{\zeta} \mathbf{F}(\mathbf{Q}(t)) \right] dt + \sqrt{\frac{4k_B \Theta}{\zeta}} d\mathbf{W}(t), \quad (5)$$

where  $\zeta$  is a friction coefficient;  $k_B$  is the Boltzmann constant;  $\Theta$  stands here for the absolute temperature;  $\kappa = (\nabla \mathbf{u})^T$  is the transpose of the velocity gradient;  $\mathbf{F}(\mathbf{Q})$  denotes the elastic force acting on the spring due to elongation; and  $\mathbf{W}(t)$  is a vector of independent Wiener processes that accounts for the Brownian contribution to the dumbbell dynamics. Note that Eq. (5) is the Lagrangian formulation of the stochastic differential equation for each dumbbell. For the FENE model, the elastic force  $\mathbf{F}(\mathbf{Q}(t))$  is expressed as

$$\mathbf{F}(\mathbf{Q}(t)) = \frac{H\mathbf{Q}(t)}{1 - \|\mathbf{Q}(t)\|^2 / \|\mathbf{Q}_0\|^2}, \quad (6)$$

where  $H$  is the Hookean constant for the spring force,  $\mathbf{Q}_0$  is the maximal extension of the elongation vector  $\mathbf{Q}$ , and  $\|\cdot\|$  denotes the modulus of a vector. To solve Eq. (5) it is customary to introduce the non-dimensional elongation vector  $\mathbf{Q}^*$  defined as  $\mathbf{Q}^*(t) \equiv \sqrt{k_B \Theta / H} \mathbf{Q}(t)$ . In doing so (dropping for simplicity the superscript\*), Eq. (5) becomes

$$d\mathbf{Q} = \left[ \kappa(\mathbf{X}(t), t) \cdot \mathbf{Q}(t) - \frac{1}{2\lambda} \frac{\mathbf{Q}(t)}{1 - \|\mathbf{Q}(t)\|^2 / b} \right] dt + \sqrt{\frac{1}{\lambda}} d\mathbf{W}(t), \quad (7)$$

where  $b = H\|\mathbf{Q}_0\|^2 / (k_B \Theta)$  and  $\lambda = \zeta / (4H)$ . So that the equations of the dumbbells we have to solve are Eqs. (4) and (7) plus the initial

conditions for the location of the center of mass and the configuration vectors of the dumbbells. The polymer stress tensor is given by Kramers' formula as

$$\boldsymbol{\tau}_p = -nk_B\Theta\mathbf{I} + n\langle\mathbf{F}(\mathbf{Q}) \otimes \mathbf{Q}\rangle, \quad (8)$$

where  $n$  is the number density of dumbbells,  $\mathbf{I}$  denotes the identity tensor, and  $\langle \cdot \rangle$  is the ensemble average in  $\mathbf{Q}$  space.

### 2.2.1. Initial conditions for the stochastic Eqs. (5) and (7)

The initial condition for the connector vector  $\mathbf{Q}$  is drawn from the equilibrium distribution ( $\boldsymbol{\kappa} = \mathbf{0}$ ) represented by the probability density function

$$\psi(\|\mathbf{Q}\|) = \begin{cases} \frac{(1 - \|\mathbf{Q}\|^2/b)^{b/2}}{2\pi b^{3/2} B(3/2, (b+2)/2)} & \text{if } \|\mathbf{Q}\|^2 < b, \\ 0 & \text{otherwise,} \end{cases} \quad (9)$$

where  $B(3/2, b+2/2)$  is the so-called Euler beta function. Since  $\psi(\|\mathbf{Q}\|)$  is symmetric, then  $\mathbf{Q}$  is isotropic and its direction is obtained by random sampling over the surface of a sphere of radius  $r$ ; the modulus of  $\mathbf{Q}$  is calculated by the rejection method [18] to generate random deviates for a probability density function  $f(r) = 4\pi r^2 \psi(r)$ . The initial dumbbells are uniformly randomly distributed over the domain. The von Neumann acceptance-rejection method was employed for the extraction of random numbers satisfying the distribution  $f(r)$  with an upper constant  $M = 0.65$ . Computation of the random unit (director) vectors needed in the equilibrium distribution was carried out by means of the method proposed by Knop [19], and all the uniformly distributed random variables were generated by the "mt19937" algorithm of Matsumoto and Nishimura [20], available from the GNU Scientific Library [21].

## 3. Non-dimensional equations

To better understand the role played by the different mechanisms taking part in the phenomenology modeled by Eqs. (1)–(8), it is convenient to introduce non-dimensional variables and formulate the micro-macro scale equations in a non-dimensional form. For the macro-scale equations, we consider the following non-dimensional variables (denoted with the superscript<sup>\*</sup>):  $(\mathbf{x}^*, \mathbf{y}^*) = L^{-1}(\mathbf{x}, \mathbf{y})$ ;  $\mathbf{u}^* = U^{-1}\mathbf{u}$ ;  $t^* = T_c^{-1}t$ ;  $p^* = p/(\rho U^2)$ , where  $L$  and  $U$  are the characteristic scales for space and velocity, respectively, and  $T_c = L/U$  is the characteristic macroscopic convective time scale. Next, for the microscopic equations we define the non-dimensional variables  $\mathbf{Q}^* = \sqrt{k_B\Theta/H}\mathbf{Q}$ ;  $t^* = T_r^{-1}t = (4H/\zeta)t = (1/\lambda)t$ , where  $T_r$  is the mesoscopic relaxation time of the spring. Once Eqs. (1), (4) and (5) are non-dimensionalized, they have the form (dropping for simplicity the superscript<sup>\*</sup>)

$$\begin{cases} \text{Re} \left( \frac{D\mathbf{u}}{Dt} + \nabla p \right) = \alpha \Delta \mathbf{u} + \left( \frac{1-\alpha}{Wi} \right) \nabla \cdot \boldsymbol{\tau}_p, \\ \nabla \cdot \mathbf{u} = 0, \end{cases} \quad (10)$$

in  $D \times (0, T]$  for the macro-scale, whereas for the micro-scale

$$\boldsymbol{\tau}_p = \langle \mathbf{F}(\mathbf{Q}) \otimes \mathbf{Q} \rangle - \mathbf{I}. \quad (11)$$

$$\begin{cases} d\mathbf{X}(t) = \mathbf{u}(\mathbf{X}(t), t)dt, \\ d\mathbf{Q} = \left[ \boldsymbol{\kappa}(\mathbf{X}(t), t) \cdot \mathbf{Q}(t) - \frac{1}{2Wi} \frac{\mathbf{Q}(t)}{1 - \|\mathbf{Q}(t)\|^2/b} \right] dt + \frac{1}{\sqrt{Wi}} d\mathbf{W}(t), \end{cases} \quad (12)$$

where  $\text{Re} = (\rho UL)/\eta_s$  is the usual Reynolds number of the Navier–Stokes equations; here  $\eta$  is the total viscosity coefficient considered as a sum of the coefficient  $\eta_s$  and a coefficient  $\eta_p$  due to the polymer,  $Wi = T_r/T_c = \zeta U/(4HL) = \lambda U/L$  is the so-called Weissenberg number; this number is a measure of importance of the

memory of the polymer with respect to the macroscopic convective time,  $\alpha = \eta_s/(\eta_s + \eta_p)$ . And  $b = H\|\mathbf{Q}_0\|^2/(k_B\Theta)$  is the finite extensibility parameter that represents the maximum extensibility allowed; in our FENE model, we shall set  $nk_B\Theta = (b+5)\eta_p/(b\lambda)$ .

## 4. Numerical formulation

We introduce in this section the numerical methods we have developed to integrate the micro-macro-scale equations. The numerical difficulties associated with the problem are the following:

- (1) For the macro-scale equations, we have on one side the typical difficulties posed by the conventional time dependent Navier–Stokes equations; specifically, the non-linear terms included in the material derivative  $D\mathbf{u}/Dt$ , and the constraint  $\nabla \cdot \mathbf{u} = 0$ ; and on the other side, the calculation of the polymer stress  $\boldsymbol{\tau}_p$ .
- (2) As for the micro-scale equations, the main difficulty is the accurate and stable integration of Itô equation for the elongation vectors  $\mathbf{Q}(t)$  along the trajectories of the dumbbells.

Although the order we arrange the calculations in our algorithm at each time step is to integrate first the micro-scale equations and then the macro-scale equations, we shall start by describing the time integration scheme of the macro-scale model.

### 4.1. Time integration of the macro-scale equations

We divide the time interval  $I = (0, T]$  into  $N$  sub-intervals  $I_n = (t_n, t_{n+1}]$ ,  $0 \leq n \leq N-1$  of equal length  $\Delta t$ , and note that, for  $(\mathbf{x}, t)$  in  $D \times (0, T)$ , the material derivative  $D\mathbf{u}/Dt$  can be expressed as  $\frac{D\mathbf{u}(\mathbf{x}, t)}{Dt} = \frac{\partial \mathbf{u}(\mathbf{X}(\mathbf{x}, t; \tau), \tau)}{\partial \tau} \Big|_{\tau=t}$  where  $\mathbf{X}(\mathbf{x}, t; \tau)$  are the characteristic curves or (fluid trajectories), which, assuming that the velocity vector  $\mathbf{u}(\mathbf{x}, t)$  is continuous in time and Lipschitz continuous in space, are the unique solution of the system of equations

$$\begin{cases} \frac{d\mathbf{X}(\mathbf{x}, t; \tau)}{d\tau} = \mathbf{u}(\mathbf{X}(\mathbf{x}, t; \tau), \tau), \\ \mathbf{X}(\mathbf{x}, t; t) = \mathbf{x}. \end{cases} \quad (13)$$

Equivalently, we can write Eq. (13) as

$$\mathbf{X}(\mathbf{x}, t; s) = \mathbf{x} - \int_{\mathbf{x}}^t \mathbf{u}(\mathbf{X}(\mathbf{x}, s; \tau), \tau) d\tau \quad (14)$$

for all  $t, s \in I$  and for all  $\mathbf{x} \in D$ . Then, at each time instant  $t_{n+1}$ , we approximate along the characteristics  $\mathbf{X}(\mathbf{x}, t_{n+1}; t)$  the material derivative by the BDF2 formula; specifically, using the notation  $\mathbf{g}(\mathbf{x}, t_n) = \mathbf{g}^n(\mathbf{x})$ , or occasionally and if confusion does not arise,  $\mathbf{g}^n$ , we have

$$\begin{aligned} \frac{D\mathbf{u}(\mathbf{x}, t)}{Dt} \Big|_{t=t_{n+1}} &= \frac{3\mathbf{u}^{n+1} - 4\mathbf{u}^n(\mathbf{X}(\mathbf{x}, t_{n+1}; t_n))}{2\Delta t} \\ &\quad + \frac{\mathbf{u}^{n-1}(\mathbf{X}(\mathbf{x}, t_{n+1}; t_{n-1}))}{2\Delta t} + O(\Delta t^2), \end{aligned} \quad (15)$$

where  $\mathbf{X}(\mathbf{x}, t_{n+1}; t_n)$  and  $\mathbf{X}(\mathbf{x}, t_{n+1}; t_{n-1})$  denote the positions at time instants  $t_n$  and  $t_{n-1}$ , respectively, of a particle that at time  $t_{n+1}$  will be at  $\mathbf{x}$ . Therefore, they are solutions of Eq. (13) when  $t = t_{n+1}$  and  $\tau = t_n$  and  $\tau = t_{n-1}$ , respectively. It is clear in Eq. (15) that for  $\mathbf{x} \in D$  one needs to calculate  $\mathbf{X}(\mathbf{x}, t_{n+1}; t_n)$  and  $\mathbf{X}(\mathbf{x}, t_{n+1}; t_{n-1})$  in order to evaluate  $\mathbf{u}^{n+1}(\mathbf{x})$ . Postponing the description of the method to calculate such positions up to the next section, we have the following time integration scheme for the macro-scale equations, which is formally  $O(\Delta t^2)$ :

Given  $\mathbf{u}^0(\mathbf{x})$  and  $\mathbf{u}^1(\mathbf{x})$ , for all  $\mathbf{x} \in D$  and for all  $n = 1, \dots, N-1$

- (1) Calculate  $\mathbf{X}(\mathbf{x}, t_{n+1}; t_n)$  and  $\mathbf{X}(\mathbf{x}, t_{n+1}; t_{n-1})$  by solving numerically Eq. (13) or, equivalently, Eq. (14).
- (2) Calculate

$$\mathbf{u}^n(\mathbf{X}(\mathbf{x}, t_{n+1}; t_n)) \quad \text{and} \quad \mathbf{u}^{n-1}(\mathbf{X}(\mathbf{x}, t_{n+1}; t_{n-1}))$$

and set

$$\bar{\mathbf{u}}^n(\mathbf{x}) = \mathbf{u}^n(\mathbf{X}(\mathbf{x}, t_{n+1}; t_n)) \quad \text{and} \quad \bar{\mathbf{u}}^{n-1}(\mathbf{x}) = \mathbf{u}^{n-1}(\mathbf{X}(\mathbf{x}, t_{n+1}; t_{n-1})). \quad (16)$$

- (3) Solve the generalized Stokes problem

$$\begin{cases} 3\rho\mathbf{u}^{n+1} - 2\Delta t\eta_s\Delta\mathbf{u}^{n+1} + 2\Delta t\nabla p^{n+1} \\ = \rho(4\bar{\mathbf{u}}^n(\mathbf{x}) - \bar{\mathbf{u}}^{n-1}(\mathbf{x})) + 2\Delta t\nabla \cdot \boldsymbol{\tau}_p^{n+1}, \\ \nabla \cdot \mathbf{u}^{n+1} = 0, \\ \mathbf{u}^{n+1} = \mathbf{0} \quad \text{on } \Gamma^s, \\ -p^{n+1}\mathbf{n} + \eta_s \frac{\partial \mathbf{u}^{n+1}}{\partial \mathbf{n}} = \mathbf{g}^{n+1} \quad \text{on } \Gamma^o, \\ -\mathbf{n} \cdot \mathbf{u}^{n+1} = a^{n+1} \quad \text{on } \Gamma^i. \end{cases} \quad (17)$$

We have termed problem (17) the generalized Stokes problem because it is mathematically and numerically equivalent to a generalized Stokes problem with the forcing terms provided by the polymer stress tensor and the known value  $\rho(4\bar{\mathbf{u}}^n(\mathbf{x}) - \bar{\mathbf{u}}^{n-1}(\mathbf{x}))$ . Next, we describe the time integration scheme for the micro-scale equations.

**Remark 1.** Since the BDF2 scheme needs  $\mathbf{u}^n$  and  $\mathbf{u}^{n-1}$  to update  $\mathbf{u}^{n+1}$ , we calculate  $\mathbf{u}^1$  with a single step scheme, such as the Crank–Nicolson or Euler implicit schemes.

#### 4.2. Time integration of the micro-scale equations

Since the formulation of the stochastic equation (7) for the elongation vector  $\mathbf{Q}(t)$  is along the trajectories of the center of mass  $\mathbf{X}(t)$  of the dumbbells, then we have to calculate such trajectories by solving for each interval  $I_n$ ,  $0 \leq n \leq N-1$ , the equations

$$\begin{cases} \frac{d\mathbf{X}(t)}{dt} = \mathbf{u}(\mathbf{X}(t), t) \quad t_n \leq t \leq t_{n+1}, \\ \mathbf{X}(t_n) \in D \quad \text{is a datum.} \end{cases} \quad (18)$$

or equivalently

$$\mathbf{X}(t) = \mathbf{X}(t_n) + \int_{t_n}^t \mathbf{u}(\mathbf{X}(\tau), \tau) d\tau \quad t_n \leq t \leq t_{n+1}. \quad (19)$$

The system (18), which is formally the same as the system (13) of the characteristic curves, has a unique solution. Next, we integrate Eq. (7) along the trajectories  $\mathbf{X}(t)$  applying the second order (weak sense) semi-implicit scheme of [22], the stability of which has been analyzed in [23]. Thus, using the notation  $\mathbf{X}^n = \mathbf{X}(t_n)$ ,  $\mathbf{Q}^n = \mathbf{Q}^n(\mathbf{X}^n)$  and  $\boldsymbol{\kappa}^n = \boldsymbol{\kappa}(\mathbf{X}^n)$ , we calculate  $\mathbf{Q}^{n+1}$  by the following splitting procedure:

$$\tilde{\mathbf{Q}}^{n+1} = \mathbf{Q}^n + \Delta t \left[ \boldsymbol{\kappa}^n \cdot \mathbf{Q}^n - \frac{1}{2\lambda} \frac{\mathbf{Q}^n}{1 - \|\mathbf{Q}^n\|^2/b} \right] + \sqrt{\frac{1}{\lambda}} \Delta \mathbf{W}, \quad (20)$$

and

$$\begin{aligned} \mathbf{Q}^{n+1} \left( 1 + \frac{1}{4\lambda} \frac{\Delta t}{1 - \|\mathbf{Q}^{n+1}\|^2/b} \right) &= \mathbf{Q}^n + \frac{\Delta t}{2} \boldsymbol{\kappa}^{n+1} \cdot \tilde{\mathbf{Q}}^{n+1} \\ &+ \frac{\Delta t}{2} \boldsymbol{\kappa}^n \cdot \mathbf{Q}^n - \frac{\Delta t}{4\lambda} \frac{\mathbf{Q}^n}{1 - \|\mathbf{Q}^n\|^2/b} + \sqrt{\frac{1}{\lambda}} \Delta \mathbf{W}. \end{aligned} \quad (21)$$

To solve for  $\mathbf{Q}^{n+1}$  we take modulus on both sides of Eq. (21) and obtain the cubic equation

$$\|\mathbf{Q}^{n+1}\|^3 - R\|\mathbf{Q}^{n+1}\|^2 - b \left( 1 + \frac{\Delta t}{4\lambda} \right) \|\mathbf{Q}^{n+1}\| + bR = 0, \quad (22)$$

where  $R$  is the modulus of the right hand side of Eq. (21). This cubic equation has one root in the interval  $(0, \sqrt{b})$ ; thus, when solving Eq. (22) we choose the root which is in that interval and then go back to Eq. (21) to solve for the components of  $\mathbf{Q}^{n+1}$ . At this point, we must note that in Eq. (22) we have  $\boldsymbol{\kappa}^{n+1} = (\nabla \mathbf{u}^{n+1}(\mathbf{X}^{n+1}))^T$ , but  $\mathbf{u}^{n+1}(\cdot)$  is not known yet; so that, we extrapolate this value by the second order extrapolation formula  $\mathbf{u}^{n+1}(\cdot) = 2\mathbf{u}^n(\cdot) - \mathbf{u}^{n-1}(\cdot)$ . Hence, we calculate  $\boldsymbol{\kappa}^{n+1}$  as

$$\boldsymbol{\kappa}^{n+1} = (2\nabla \mathbf{u}^n(\mathbf{X}^{n+1}) - \nabla \mathbf{u}^{n-1}(\mathbf{X}^{n+1}))^T. \quad (23)$$

Once we know  $\mathbf{Q}^{n+1}$ , we proceed to calculate  $\boldsymbol{\tau}_p^{n+1}$  applying (8). The Wiener process  $\Delta \mathbf{W}$  is the one proposed by [22] for schemes of second order, namely

$$\Delta \mathbf{W} = \sqrt{\Delta t} \left( \mathbf{Y} - \frac{1}{2} \right) \left[ c_1 \left( \mathbf{Y} - \frac{1}{2} \right)^2 + c_2 \right], \quad (24)$$

where  $\mathbf{Y}$  is a stochastic variable uniformly distributed in  $[0, 1] \times [0, 1] \times [0, 1]$ , and the constants  $c_1$  and  $c_2$ , satisfying all the conditions for the moments of  $\mathbf{Y}$ , have the values  $c_1 = 14.14855378$  and  $c_2 = 1.21569221$ . The time integration scheme is the following:

Given (for each dumbbell)  $\mathbf{Q}^0$  and  $\mathbf{X}^0$ , and  $\mathbf{u}^0$  (see initial conditions), then for each dumbbell and for all  $n = 1, \dots, N-1$

- (1) Calculate  $\boldsymbol{\kappa}^n = (\nabla \mathbf{u}^n(\mathbf{X}^n))^T$ .
- (2) Calculate  $\mathbf{X}^{n+1}$  by solving numerically Eq. (18) or equivalently Eq. (19), and  $\boldsymbol{\kappa}^{n+1}$  by formula (23).
- (3) Calculate the components of  $\tilde{\mathbf{Q}}^{n+1}$  making use of Eq. (20).
- (4) Calculate  $\|\mathbf{Q}^{n+1}\|$  by solving Eq. (22) and choose the root which is in  $(0, \sqrt{b})$ .
- (5) Calculate the components of  $\mathbf{Q}^{n+1}$  making use of Eq. (21).
- (6) Calculate  $\boldsymbol{\tau}_p^{n+1}$  applying formula (8).

**Remark 2.** One may argue that integrating the macro-scale equations with a second order scheme will not significantly improve the accuracy of the solution versus that obtained with a first order scheme, such as the Euler implicit scheme used by many researchers, since the main component of error committed in the calculation of the macro-scale variables is contributed by the approximation of  $\boldsymbol{\tau}_p^{n+1}$  via an ensemble average that is described below, and the statistical error of such an average is  $O(1/\sqrt{M})$ ,  $M$  being the number of dumbbells used in the average. Admitting that such an argument is true, we, nevertheless, use the BDF2 scheme for the following reasons: (1) to be consistent in terms of numerical convergence with the second order scheme employed to integrate the stochastic equations along the dumbbells trajectories; and (2) because from a computational point of view, the cost of the BDF2 scheme in terms of memory and CPU time is marginally higher than that of any first order scheme.

#### 4.3. Finite element formulation

Our next concern is the formulation of the time discretization schemes in a finite element context. To this end, we shall use unstructured meshes composed of quasi-uniformly regular triangular elements  $T_j$ ,  $1 \leq j \leq NE$ , where  $NE$  is the number of elements of the mesh. The set  $D_h = \{T_j, 1 \leq j \leq NE\}$  denotes a uniformly regular partition of  $D$  such that  $D \cup \Gamma = \cup_{1 \leq j \leq NE} T_j$ . The triangles  $T_j$  that are in the interior of  $D$  have straight sides, whereas those triangles which intersect the boundary  $\Gamma$  may have curved sides. One has to make additional assumptions on curved triangles.  $P_m(T_j)$  denotes

the set of polynomials (or polynomial-like functions) of degree  $\leq m$  defined on the triangle  $T_j$ . The type of finite elements we employ for space discretization of velocity and pressure is the so-called Taylor-Hood  $P_2/P_1$  element, that is, quadratic polynomials for velocity and linear polynomials for pressure on each triangle  $T_j$ ; this element is known to be stable in the sense that it satisfies the inf-sup condition. As for the polymer stress tensor,  $\boldsymbol{\tau}_p$ , we shall use  $P_1$ -element, that is, linear polynomials. So that, we are now in a position to define the finite element spaces for velocity, pressure and polymer stress tensor associated to the partition  $D_h$ .

Velocity:

$$\mathbf{V}_h = \left\{ \mathbf{v}_h \in C^0(\bar{D}) : \mathbf{v}_h|_{T_j} \in P_2(T_j), 1 \leq j \leq NE \right\},$$

and

$$\mathbf{V}_{h0} = \left\{ \mathbf{v}_h \in \mathbf{V}_h : \mathbf{v}_h|_{\Gamma^o} = \mathbf{0} \right\}.$$

Pressure:

$$Q_h = \{ q_h \in C^0(\bar{D}) : q_h|_{T_j} \in P_1(T_j), 1 \leq j \leq NE \}.$$

Polymer stress tensor:

$$\mathbf{S}_h = \{ \mathbf{s}_h \in C^0(\bar{D})^4 : \mathbf{s}_h|_{T_j} \in P_1(T_j)^4, 1 \leq j \leq NE \}.$$

Here,  $C^0(\bar{D})$  means that the functions are bounded and continuous up to the boundary;  $P_2(T_j)^2$  means that the two components of the velocity vector are piecewise polynomials of degree 2, whereas  $P_1(T_j)^4$  denotes that the four components of the stress tensor are piecewise linear polynomials. Let  $MV$  be the number of velocity nodes in the mesh, and let  $MP$  be the number of pressure (also polymer stress tensor) nodes in the mesh. Then, any  $\mathbf{v}_h \in \mathbf{V}_h$ ,  $q_h \in Q_h$  and  $\mathbf{s}_h \in \mathbf{S}_h$  can be represented as

$$\begin{aligned} \mathbf{v}_h &= \sum_{i=1}^{MV} \mathbf{V}_i \Phi_i, \\ q_h &= \sum_{k=1}^{MP} Q_k \Psi_k, \\ \mathbf{s}_h &= \sum_{k=1}^{MP} \mathbf{S}_k \Psi_k, \end{aligned} \quad (25)$$

where  $\{\Phi_i \mathbf{e}_1, \Phi_i \mathbf{e}_2\}$ ,  $\{\Psi_k\}$  and  $\{\Psi_k \mathbf{e}_1 \mathbf{e}_1, \Psi_k \mathbf{e}_1 \mathbf{e}_2, \Psi_k \mathbf{e}_2 \mathbf{e}_1, \Psi_k \mathbf{e}_2 \mathbf{e}_2\}$  denote the sets of global basis functions of the finite element spaces  $\mathbf{V}_{h0}$ ,  $Q_h$  and  $\mathbf{S}_h$ , respectively;  $\mathbf{V}_i = (V_{1i}, V_{2i})$ ,  $Q_k$  and  $\mathbf{S}_k = (S_{11k}, S_{12k}, S_{21k}, S_{22k})$  are the sets of nodal values of  $\mathbf{v}_h$ ,  $q_h$  and  $\mathbf{s}_h$ , respectively; and  $\mathbf{e}_1$  and  $\mathbf{e}_2$  denote the elements of the canonical basis of  $\mathbb{R}^2$ .

#### 4.3.1. Finite element formulation of the generalized Stokes problem (17)

Assuming that at the time instant  $t_n$ ,  $\mathbf{u}_h^{n-1}$  and  $\mathbf{u}_h^n \in \mathbf{V}_{h0}$ ,  $p_h^n \in Q_h$ ,  $\boldsymbol{\tau}_p^{n+1} \in \mathbf{S}_h$  are known, we wish to find  $\mathbf{u}_h^{n+1} \in \mathbf{V}_{h0}$  and  $p_h^{n+1} \in Q_h$  such that  $-\mathbf{n} \cdot \mathbf{u}_h^{n+1} = \alpha^{n+1}$  on  $\Gamma^i$ , and for any  $\mathbf{v}_h \in \mathbf{V}_{h0}$  and  $q_h \in Q_h$

$$\begin{cases} 3(\rho \mathbf{u}_h^{n+1}, \mathbf{v}_h)_D + 2\Delta t (\nabla \mathbf{u}_h^{n+1}, \nabla \mathbf{v}_h)_D \\ -2\Delta t (p_h^{n+1}, \nabla \cdot \mathbf{v}_h)_D = (\rho(4\bar{\mathbf{u}}_h^n - \bar{\mathbf{u}}_h^{n-1}), \mathbf{v}_h)_D \\ +2\Delta t (\mathbf{g}^{n+1}, \mathbf{v}_h)_{\Gamma^o} + 2\Delta t (\nabla \cdot \boldsymbol{\tau}_p^{n+1}, \mathbf{v}_h)_D, \\ (\nabla \cdot \mathbf{u}_h^{n+1}, q_h)_D = 0, \end{cases} \quad (26)$$

where the following notation has been used:  $(\mathbf{a}, \mathbf{b})_D = \int_D \mathbf{a} \cdot \mathbf{b} \, dx$ ,  $(\nabla \mathbf{a}, \nabla \mathbf{b})_D = \int_D \nabla \mathbf{a} : \nabla \mathbf{b} \, dx$  and  $(\mathbf{a}, \mathbf{b})_{\Gamma^o} = \int_{\Gamma^o} \mathbf{a} \cdot \mathbf{b} \, ds$ .

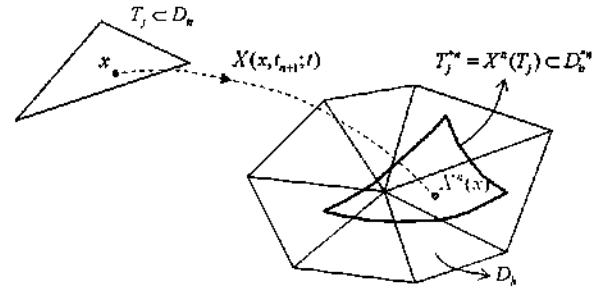


Fig. 2. Bijective transformation  $\mathbf{X}(\cdot, t_{n+1}, t_n)$  of element  $T_j$  and fixed partition  $D_h$  onto element  $T_j^{*l}$  and virtual partition  $D_h^{*l}$ .

**Remark 3.** In Eq. (26),  $\bar{\mathbf{u}}_h^n(\mathbf{x})$  and  $\bar{\mathbf{u}}_h^{n-1}(\mathbf{x}) \in \mathbf{V}_h$  are the finite element approximations to

$$\bar{\mathbf{u}}_h^n(\mathbf{x}) = \mathbf{u}^n(\mathbf{X}(\mathbf{x}, t_{n+1}, t_n)) \quad \text{and} \quad \bar{\mathbf{u}}_h^{n-1}(\mathbf{x}) = \mathbf{u}^{n-1}(\mathbf{X}(\mathbf{x}, t_{n+1}, t_{n-1})).$$

These approximations are calculated by the semi-Lagrangian method that is explained next.

**Remark 4.** In contrast with other micro-macro algorithms, which calculate  $(\mathbf{u}_h^{n+1}, p_h^{n+1})$  using the value of the polymer stress tensor at the previous time step  $t_n$ , making the whole scheme be  $\mathcal{O}(\Delta t)$ , we use  $\boldsymbol{\tau}_p^{n+1}$  in order to be formally consistent with the second order BDF2 discretization of the material derivative.

**Remark 5.** Notice that the bilinear form for the velocity  $\mathbf{u}_h^{n+1}$  is symmetric and positive definite, so that we solve Eq. (26) by the Uzawa-preconditioned conjugate gradient algorithm of [24].

#### 4.3.2. Semi-Lagrangian calculation of $\bar{\mathbf{u}}_h^l(\mathbf{x})$ when $l = n - 1$ and $n$

There are different approaches to approximate  $\bar{\mathbf{u}}^l(\mathbf{x}) = \mathbf{u}^l(\mathbf{X}(\mathbf{x}, t_{n+1}, t_l))$ . For instance, on one side, we have the so-called Lagrange-Galerkin method (also known as Characteristic-Galerkin method) introduced in [25] and [26], and further analyzed in [27] for the incompressible Navier-Stokes equations and [28] for the scalar linear advection equation; or the weak Lagrange-Galerkin of Benque et al. [29]. These methods are specifically formulated in a finite element framework since they perform a  $L^2$ -projection of  $\mathbf{u}^l(\mathbf{X}(\mathbf{x}, t_{n+1}, t_l))$  onto  $\mathbf{V}_h$ , and they have been used in viscoelastic flow by [13,15] and [16], just to cite a few. On the other side, in a finite difference context, we have the so-called semi-Lagrangian methods that were first introduced in the numerical weather prediction community by Robert [30], and since then they have become a standard scheme in numerical weather prediction models. Roughly speaking, the idea of the semi-Lagrangian schemes when they are applied in combination with finite differences (or finite volume methods) for space discretization consists of computing the departure points (or feet of the characteristics)  $\mathbf{X}(\mathbf{x}, t_{n+1}, t_l)$  for the grid points  $\{\mathbf{x}_i\}$  of a fixed square uniform grid and then calculating  $\bar{\mathbf{u}}^l(\mathbf{x}_i)$  by interpolation of degree  $\geq 2$ . However, recently Bermejo and Carpio [31] have devised a semi-Lagrangian scheme more consistent with the finite element methodology based on the Galerkin projection, and it is this semi-Lagrangian scheme that we use to calculate  $\bar{\mathbf{u}}_h^l(\mathbf{x})$ . Briefly speaking, the idea of the new semi-Lagrangian Galerkin projection scheme, hereafter, SLG scheme unless otherwise stated, consists of first generating a virtual partition  $D_h^{*l}$ , which is the image of the fixed partition  $D_h$ , by convecting backwards in time the elements  $T_j$  of the fixed mesh; specifically, let  $\bar{D}^{*l} = \{\mathbf{y} \in \mathbb{R}^2 : \mathbf{y} = \mathbf{X}(\mathbf{x}, t_{n+1}, t_l), \forall \mathbf{x} \in \bar{D}\}$ , then we define  $\bar{D}_j^{*l} = \{T_j^{*l} \subset \bar{D}^{*l} : T_j^{*l} = \mathbf{X}(T_j), 1 \leq j \leq NE\}$ , where  $\mathbf{X}^l(T_j)$  is a shorthand notation for  $\mathbf{X}(\mathbf{x}, t_{n+1}, t_l)$ , for any  $\mathbf{x}$  in  $T_j$ . Therefore,  $T_j^{*l}$  is the image of  $T_j$  by the action of the bijective transformation  $\mathbf{X}(\cdot, t_{n+1}, t_l)$ . See Fig. 2, where it is shown that, for  $l = n$  and  $\Delta t$  suf-

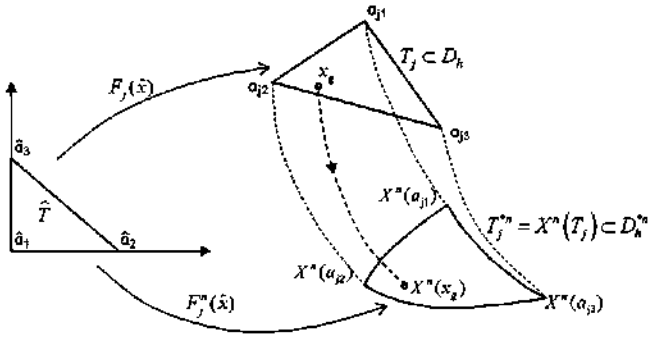


Fig. 3. Relation among triangles  $\hat{T}$ ,  $T_j$  and  $T_j^{*n}$  through the bijective transformation  $\mathbf{X}(\cdot, t_{n+1}; t_n)$ , bijective mapping  $F_j^n(\hat{\mathbf{x}})$ , and invertible affine transformation  $F_j(\hat{\mathbf{x}})$ .

ficiently small, the triangle  $T_j^{*n}$  may intersect several elements  $T_k$  of the fixed partition  $D_h$ . Moreover, Fig. 3 shows graphically (1) the relation of  $T_j$  with  $T_j^{*n}$  through the bijective transformation  $\mathbf{X}(\mathbf{x}, t_{n+1}; t_n)$ ; (2) the relation of  $T_j^{*n}$  with the triangle of reference  $\hat{T}$ , whose vertices are  $(0, 0)$ ,  $(1, 0)$  and  $(0, 1)$ , via the bijective mapping  $F_j^n(\hat{\mathbf{x}})$ ; and (3) the relation of  $\hat{T}$  with  $T_j$  through the invertible affine transformation  $F_j(\hat{\mathbf{x}})$ . Secondly, we consider the finite element space  $\mathbf{V}_h^i$  associated to the partition  $D_h^i$ :

$$\mathbf{V}_h^i = \{\mathbf{v}_h^i \in C^0(\bar{D}^i)^2 : \forall T_j^i \in D_h^i, \mathbf{v}_h^i|_{T_j^i} \in P_2(T_j^i)^2, \quad 1 \leq j \leq NE\},$$

where for any  $\mathbf{x} \in T_j$  and  $\mathbf{y} = \mathbf{X}(\mathbf{x}, t_{n+1}; t_i)$ , we can define

$$P_2(T_j^i) = \{p^*(\mathbf{y}) : p^*(\mathbf{y}) = p(\mathbf{x}) \text{ with } p(\mathbf{x}) \in P_2(T_j)\}.$$

The definitions of  $\mathbf{V}_h^i$  and  $P_2(T_j^i)$  imply that if  $\{(\Phi_i \mathbf{e}_1, \Phi_i \mathbf{e}_2)\}_{i=1}^{MV}$  is the set of global basis functions of  $\mathbf{V}_h$ , then we can define a set  $\{(\Phi_i^i \mathbf{e}_1, \Phi_i^i \mathbf{e}_2)\}_{i=1}^{MV}$  of global basis functions for  $\mathbf{V}_h^i$  as follows: for  $\mathbf{x} \in T_j$  and  $\mathbf{y} = \mathbf{X}(\mathbf{x}, t_{n+1}; t_i) \in T_j^i$ ,  $(\Phi_i^i(\mathbf{y}))|_{T_j^i} = (\Phi_i(\mathbf{x}))|_{T_j}$  and  $(\Phi_i^i(\mathbf{y}_k))|_{T_j^i} = (\Phi_i(\mathbf{x}_k))|_{T_j} = \delta_{ik}$ , where  $\mathbf{x}_k$  are the mesh points for velocity. So that, to any function  $\mathbf{v}_h^i \in \mathbf{V}_h^i$ ,

$$\mathbf{v}_h^i(\mathbf{y}) = \sum_{i=0}^{MV} \mathbf{v}_i^i \Phi_i^i(\mathbf{y})$$

there corresponds a function  $\tilde{\mathbf{v}}_h^i(\mathbf{x}) \in \mathbf{V}_h$  such that

$$\tilde{\mathbf{v}}_h^i(\mathbf{x}) = \sum_{i=0}^{MV} \mathbf{v}_i^i \Phi_i(\mathbf{x}).$$

Next, we consider  $\mathbf{u}_h^i(\mathbf{X}(\mathbf{x}, t_{n+1}; t_i))$ ; this is a continuous function defined on  $D_h^i$  that does not belong to  $\mathbf{V}_h^i$ . We approximate it on this space by the function  $\mathbf{u}_h^i(\mathbf{y})$ ,  $\mathbf{y} = \mathbf{X}(\mathbf{x}, t_{n+1}; t_i)$ , which is the  $L^2$ -projection of  $\mathbf{u}_h^i(\mathbf{X}(\mathbf{x}, t_{n+1}; t_i))$  onto  $\mathbf{V}_h^i$ . In other words,  $(\mathbf{u}_h^i, \mathbf{v}_h^i)_{D_h^i} = (\mathbf{u}_h^i(\mathbf{X}(\mathbf{x}, t_{n+1}; t_i)), \mathbf{v}_h^i)_{D_h^i}$  or equivalently, given that for any integrable function  $G^*$  defined on  $D_h^i$ ,  $\int_{D_h^i} G^*(\mathbf{y}) d\mathbf{y} = \sum_{j=1}^{NE} \int_{T_j^i} G^*(\mathbf{y}) d\mathbf{y}$ , we can set

$$\sum_{j=1}^{NE} \int_{T_j^i} \mathbf{u}_h^i(\mathbf{y}) \cdot \mathbf{v}_h^i(\mathbf{y}) d\mathbf{y} = \sum_{j=1}^{NE} \int_{T_j^i} \mathbf{u}_h^i(\mathbf{y}) \cdot \mathbf{v}_h^i(\mathbf{y}) d\mathbf{y}, \quad \forall \mathbf{v}_h^i \in \mathbf{V}_h^i.$$

Noting that, in general,  $T_j^i$  are curved triangles, which are not computationally feasible to generate, then the SLG scheme approximates the integrals  $\int_{T_j^i} G^*(\mathbf{y}) d\mathbf{y}$  by  $\int_{\hat{T}_j^i} G^*(\mathbf{y}) d\mathbf{y}$ ,

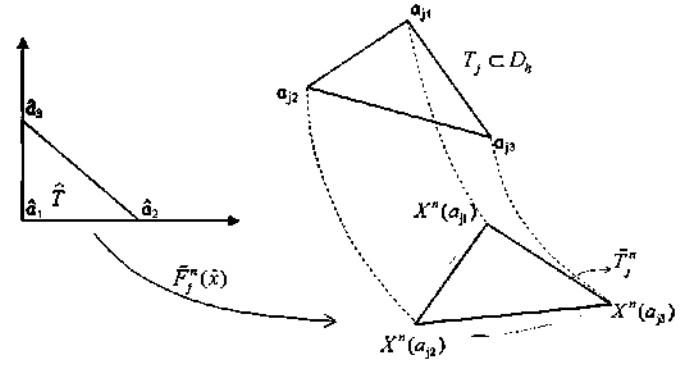


Fig. 4. Affine transformation  $\tilde{F}_j^n(\hat{\mathbf{x}})$  and linear approximation  $\tilde{T}_j^n$  to  $T_j^{*n}$ .

where  $\tilde{T}_j^n$  are straight side triangles whose vertices are:  $\mathbf{b}_{j1} = \mathbf{X}(\mathbf{a}_{j1}, t_{n+1}; t_i)$ ,  $\mathbf{b}_{j2} = \mathbf{X}(\mathbf{a}_{j2}, t_{n+1}; t_i)$  and  $\mathbf{b}_{j3} = \mathbf{X}(\mathbf{a}_{j3}, t_{n+1}; t_i)$ , with  $\{\mathbf{a}_{j1}, \mathbf{a}_{j2}, \mathbf{a}_{j3}\}$  being the vertices of  $T_j$ . See Fig. 4, where it is shown that  $\tilde{T}_j^n$  is a linear approximation to the curved triangle  $T_j^{*n}$ .

To evaluate  $\int_{\tilde{T}_j^n} \mathbf{u}_h^i(\mathbf{y}) \cdot \mathbf{v}_h^i(\mathbf{y}) d\mathbf{y}$ , we consider that  $\mathbf{u}_h^i(\mathbf{y})|_{\tilde{T}_j^n} \in P_2(\tilde{T}_j^n)^2$ , so that we can write  $\mathbf{u}_h^i(\mathbf{y})|_{\tilde{T}_j^n} = \sum_{k=1}^6 \mathbf{V}_k^i \tilde{\phi}_k(\mathbf{y})$ , where  $\{\tilde{\phi}_k(\mathbf{y})\}$  is the set of local basis function for the set of polynomials  $P_2(\tilde{T}_j^n)$  defined on  $\tilde{T}_j^n$ , and for  $1 \leq i \leq 6$  we set  $\mathbf{v}_h^i(\mathbf{y}) = (\tilde{\phi}_i(\mathbf{y}) \mathbf{e}_1, \tilde{\phi}_i(\mathbf{y}) \mathbf{e}_2)$ ; hence, we have

$$\begin{aligned} \int_{\tilde{T}_j^n} \mathbf{u}_h^i(\mathbf{y}) \cdot \mathbf{v}_h^i(\mathbf{y}) d\mathbf{y} &= \sum_{k=1}^6 (V_{1k}^i, V_{2k}^i) \int_{\tilde{T}_j^n} \tilde{\phi}_i(\mathbf{y}) \tilde{\phi}_k(\mathbf{y}) d\mathbf{y} \\ &= \sum_{k=1}^6 \tilde{m}_{ik}^i (V_{1k}^i, V_{2k}^i), \quad 1 \leq i \leq 6, \end{aligned}$$

where  $\tilde{m}_{ik}^i = \int_{\tilde{T}_j^n} \tilde{\phi}_i(\mathbf{y}) \tilde{\phi}_k(\mathbf{y}) d\mathbf{y}$  denotes the local mass matrix for the element  $\tilde{T}_j^n$ . Making use of the invertible affine transformation  $\tilde{F}_j^i : \hat{T} \rightarrow \tilde{T}_j^n$ , see Fig. 4, we can evaluate the coefficients  $\tilde{m}_{ik}^i$  as it is usual in finite element technology; that is

$$\tilde{m}_{ik}^i = \int_{\tilde{T}_j^n} \tilde{\phi}_i(\mathbf{y}) \tilde{\phi}_k(\mathbf{y}) d\mathbf{y} = \int_{\hat{T}} \tilde{\phi}_i(\hat{\mathbf{x}}) \tilde{\phi}_k(\hat{\mathbf{x}}) |J_{\tilde{F}_j^i}| d\hat{\mathbf{x}}, \quad (27)$$

where  $J_{\tilde{F}_j^i}$  denotes the Jacobian determinant of the transformation  $\tilde{F}_j^i$ , which is constant for each element  $\tilde{T}_j^n$ , and  $\{\tilde{\phi}_k(\hat{\mathbf{x}})\}$  is the set of basis functions for the set  $P_2(\hat{T})$  of polynomials defined on the reference triangle  $\hat{T}$ . To evaluate  $\int_{\tilde{T}_j^n} \mathbf{u}_h^i(\mathbf{y}) \cdot \mathbf{v}_h^i(\mathbf{y}) d\mathbf{y}$ , we set, for  $i \leq 6$ ,

$\mathbf{v}_h^i = (\tilde{\phi}_i(\mathbf{y}) \mathbf{e}_1, \tilde{\phi}_i(\mathbf{y}) \mathbf{e}_2)$ , and making use of the affine transformation  $\tilde{F}_j^i$  and quadrature rules we have that

$$\begin{aligned} \mathbf{r}_i^i &= (r_{i1}^i, r_{i2}^i) = \int_{\tilde{T}_j^n} (v_{1h}^i(\mathbf{y}), v_{2h}^i(\mathbf{y})) \tilde{\phi}_i(\mathbf{y}) d\mathbf{y} \\ &= \int_{\hat{T}} (v_{1h}^i(F_j^i(\hat{\mathbf{x}})), v_{2h}^i(F_j^i(\hat{\mathbf{x}}))) \tilde{\phi}_i(\hat{\mathbf{x}}) |J_{F_j^i}| d\hat{\mathbf{x}} \simeq \\ &= |J_{F_j^i}| \sum_{g=1}^{ngp} \varpi_g (v_{1h}^i(F_j^i(\hat{\mathbf{x}}_g)), v_{2h}^i(F_j^i(\hat{\mathbf{x}}_g))) \tilde{\phi}_i(\hat{\mathbf{x}}_g), \end{aligned} \quad (28)$$

where  $ngp$  denotes the number of quadrature points of the rule, and  $(\hat{\mathbf{x}}_g, \varpi_g)$  is the  $g$ th quadrature pair, denoting the quadrature point and its corresponding weight, respectively; for  $\tilde{\phi}_i(\hat{\mathbf{x}}) \in P_2(\hat{T})$  we use Hammer's rule of 7 points [32]. The key point in Eq. (28) is the evaluation of  $(v_{1h}^i(\mathbf{y}_g), v_{2h}^i(\mathbf{y}_g))$ ,  $\mathbf{y}_g = F_j^i(\hat{\mathbf{x}}_g)$ , because we must identify

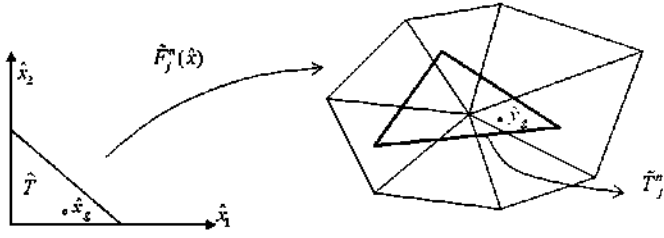


Fig. 5. The quadrature point  $\hat{\mathbf{x}}_g$  has as image in  $\tilde{T}_j^n$  the point  $\tilde{\mathbf{y}}_g$  which is in a fixed mesh element.

the element  $T_k$  of the fixed mesh that contains  $\mathbf{y}_g$  and then calculate  $(r_{1h}^l(\mathbf{y}_g), r_{2h}^l(\mathbf{y}_g))$ , making use of the element of reference  $\hat{T}$  as is usual in finite element technology (see Fig. 5). To perform this task, locate the element and calculate the interpolation in this element, we have developed a new search-locate algorithm for unstructured triangular meshes based on the one presented in [33], which is able to locate the element containing a point given by the Cartesian coordinates  $(x_1, x_2)$  and then calculate the reference coordinates  $(\hat{x}_1, \hat{x}_2)$  of this point (Fig. 6 illustrates the capabilities of this method to search and locate points in unstructured meshes on multiply connected domains). We must remark that the approximation of  $T_j^n$  by  $\tilde{T}_j^n$  introduces the significant difference that the calculation of the points  $\tilde{F}_j^n(\hat{\mathbf{x}}_g)$  is done very efficiently, and so is the identification of the elements  $T_k$  wherein they are located, because the search to find out  $T_k$  is local in the sense that if we know, by previous calculations, the element  $T_k$  that contains, say  $\mathbf{X}^n(\mathbf{a}_1)$ ; then, when both  $\Delta t$  and  $|\mathbf{u}|$  are sufficiently small,  $\tilde{F}_j^n(\hat{\mathbf{x}}_g)$  must be either in the same element  $T_k$  or in a neighbor element. Fig. 5 illustrates the situation. In contrast with Lagrange–Galerkin methods, for which one needs to solve the system (13) for each quadrature point in the element  $T_j$ , or for interpolatory semi-Lagrangian schemes of [34] and [35] where the system (13) is numerically solved for each mesh-point, in the SLG scheme one has to solve (13) only for the vertices of the triangles, what means a large saving in CPU time. Details on the numerical analysis of the SLG method as well as a comparison of its performance on typical pure convection problems and the Navier–Stokes equations with Lagrange–Galerkin and interpolatory semi-Lagrangian schemes can be found in [31]. An algorithmic description of the SLG projection scheme is as follows:

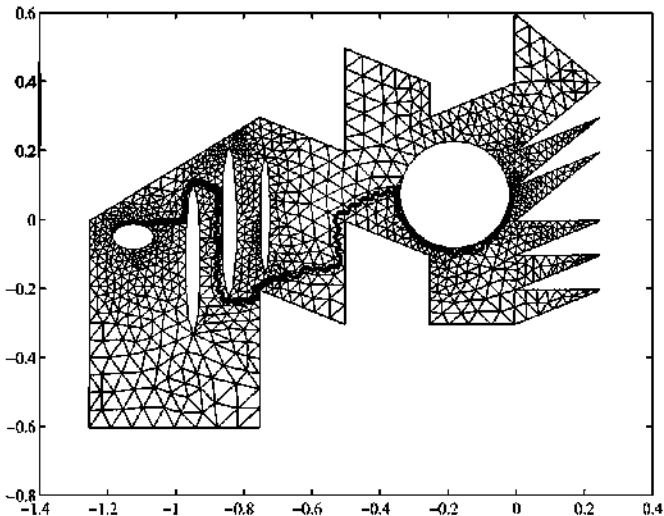


Fig. 6. The search-locate algorithm starts at point  $(-1.185, -0.03749)$  and after 164 steps finds out the element containing the point  $(-0.01816, 0.09047)$ . The figure shows the path followed by the algorithm.

Assuming that the velocities  $\mathbf{u}_h^l \in \mathbf{V}_{h0}$  are known, find  $\tilde{\mathbf{u}}_h^l \in \mathbf{V}_h$  through the following steps: For  $j = 1, 2, \dots, NE$

- (1) Calculate the element  $\tilde{T}_j^l$  by solving Eq. (13), with  $\mathbf{x} = \mathbf{a}_{jk}$ ,  $1 \leq k \leq 3$  being the vertices of the fixed mesh element  $T_j$ .
- (2) Calculate the elemental mass matrix  $\tilde{M}_j^l$  by Eq. (27) and assemble it in the global mass matrix  $\tilde{M}^l$ .
- (3) Calculate  $\mathbf{r}_1^l$  by Eq. (28) and assemble it in the vectors  $\mathbf{R}_1^l = (r_{11}^l, \dots, r_{1MV}^l)^T$  and  $\mathbf{R}_2^l = (r_{21}^l, \dots, r_{2MV}^l)^T$ .
- (4) Calculate the entries of the vectors

$$\tilde{\mathbf{V}}_1^l = (\tilde{V}_{11}^l, \dots, \tilde{V}_{1MV}^l)^T; \quad \tilde{\mathbf{V}}_2^l = (\tilde{V}_{21}^l, \dots, \tilde{V}_{2MV}^l)^T,$$

$$\text{by solving } \tilde{M}^l(\tilde{\mathbf{V}}_1^l, \tilde{\mathbf{V}}_2^l) = (\mathbf{R}_1^l, \mathbf{R}_2^l).$$

- (5) Set  $\tilde{\mathbf{u}}_h^l \in \mathbf{V}_h$  as

$$\tilde{\mathbf{u}}_h^l(\mathbf{x}) = \sum_{i=1}^{MV} (\tilde{\mathbf{V}}_1^l, \Phi_i(\mathbf{x}), \tilde{\mathbf{V}}_2^l, \Phi_i(\mathbf{x})). \quad (29)$$

#### 4.3.3. Calculation of the backward and forward trajectories of the flow

At each time instant  $t_{n+1}$ , we have to compute the departure points  $\mathbf{X}(\mathbf{x}, t_{n+1}; t_n)$  and  $\mathbf{X}(\mathbf{x}, t_{n+1}; t_{n-1})$  of the characteristic curves of the material derivative operator, as well as the points  $\mathbf{X}(t_{n+1})$  which are the positions of the center of mass of the dumbbells. Looking at Eqs. (13)–(14) for  $\mathbf{X}(\mathbf{x}, t_{n+1}; t_n)$  and  $\mathbf{X}(\mathbf{x}, t_{n+1}; t_{n-1})$ ; and (18)–(19) for  $\mathbf{X}(t_{n+1})$ , we note that the calculation of  $\mathbf{X}(\mathbf{x}, t_{n+1}; t_n)$ ,  $\mathbf{X}(\mathbf{x}, t_{n+1}; t_{n-1})$  consists of solving backwards in time for all  $\mathbf{x} \in \hat{D}$ , whereas the calculation of  $\mathbf{X}(t_{n+1})$  represents for each dumbbell the solution forward in time of the same system of ordinary differential equations; therefore, we shall apply the same numerical method to calculate the solutions  $\mathbf{X}(\mathbf{x}, t_{n+1}; t_n)$ ,  $\mathbf{X}(\mathbf{x}, t_{n+1}; t_{n-1})$  and  $\mathbf{X}(t_{n+1})$ . There are a number of numerical methods that can be used to perform these calculations, for instance, considering only explicit schemes, we have the families of Runge–Kutta methods of order  $\geq 2$ , or some predictor–corrector schemes of order  $\geq 2$ ; however, based on the experience of [36] and our own one, we find satisfactory the second order scheme described in [34] which combines the scheme 2 of [36] with our new search-locate algorithm for unstructured meshes.

Calculation of  $\mathbf{X}(\mathbf{x}, t_{n+1}; t_l)$  when  $l = n - 1$  or  $n$

Recalling Eq. (14) and approximating the integral by the midpoint rule up to  $O(\Delta t^3)$ , we can set

$$\mathbf{X}(\mathbf{x}, t_{n+1}; t_l) = \mathbf{x}_l - \Delta t \mathbf{u}_h(\mathbf{X}(\mathbf{x}_l, t_{n+1}; t_l + \vartheta_{nl}), t_l + \vartheta_{nl}), \quad (30)$$

where  $\vartheta_{nl} = (n + 1 - l)\Delta t/2$ , and the points  $\mathbf{x}_l$  are the vertices of the triangles of the fixed mesh. Since all the variables are calculated at time instants  $n\Delta t$ , we have to approximate both  $\mathbf{X}(\mathbf{x}_l, t_{n+1}; t_l + \vartheta_{nl})$  and  $\mathbf{u}_h(\cdot, t_l + \vartheta_{nl})$  in this formula. The approximation of  $\mathbf{u}_h(\cdot, t_l + \vartheta_{nl})$  is calculated via the second order extrapolation formula

$$\mathbf{u}_h(\cdot, t_l + \vartheta_{nl}) = \begin{cases} \frac{3}{2}\mathbf{u}_h(\cdot, t_n) - \frac{1}{2}\mathbf{u}_h(\cdot, t_{n-1}), & l = n, \\ \mathbf{u}_h(\cdot, t_n), & l = n - 1, \end{cases} \quad (31)$$

whereas in order to approximate  $\mathbf{X}(\mathbf{x}_l, t_{n+1}; t_l + \vartheta_{nl})$  we note that  $\mathbf{X}(\mathbf{x}_l, t_{n+1}; t_{n+1}) = \mathbf{x}_l$ , so that we can write

$$\mathbf{X}(\mathbf{x}_l, t_{n+1}; t_l + \vartheta_{nl}) = \begin{cases} \frac{1}{2}(\mathbf{x}_l + \mathbf{X}(\mathbf{x}_l, t_{n+1}; t_n)), & l = n, \\ \mathbf{X}(\mathbf{x}_l, t_{n+1}; t_n), & l = n - 1. \end{cases} \quad (32)$$

Hence, defining  $\mathbf{e}(\mathbf{x}_l, t_{n+1}; t_l) \equiv \mathbf{x}_l - \mathbf{X}(\mathbf{x}_l, t_{n+1}; t_l)$ , we have, by a Taylor expansion up to order  $O(\Delta t^2)$ , that  $\mathbf{X}(\mathbf{x}_l, t_{n+1}; t_l + \vartheta_{nl}) =$

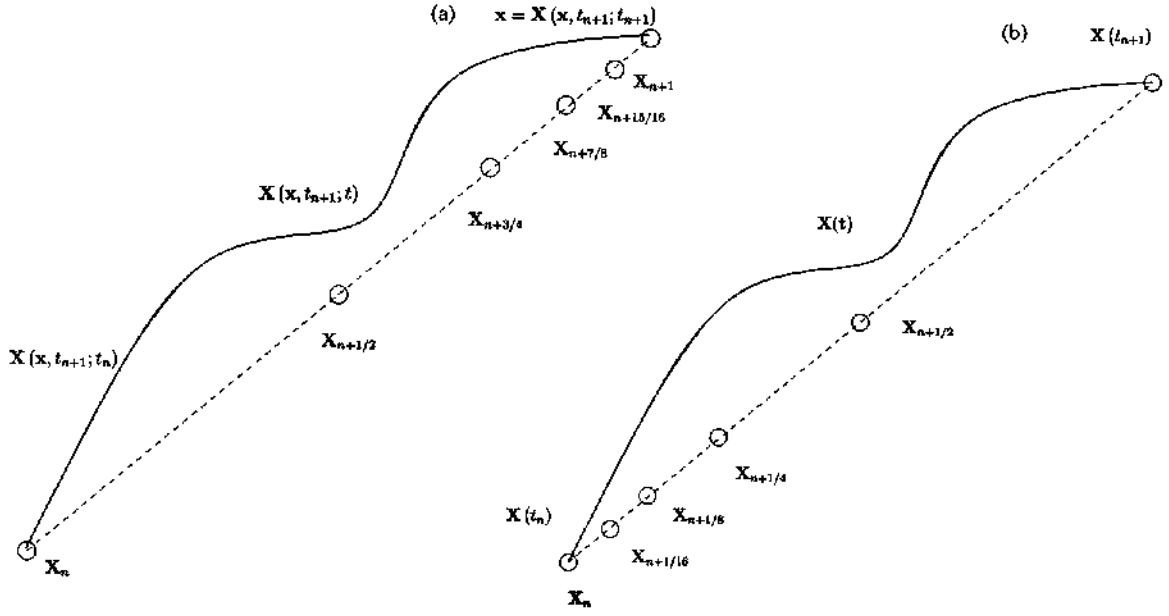


Fig. 7. Time adaptive scheme to compute  $\mathbf{X}(\cdot, t_{n+1}; t_l)$  (panel (a)), and  $\mathbf{X}(t_{n+1})$  (panel (b)).

$\mathbf{x}_i - \frac{1}{2}\mathbf{e}(\mathbf{x}_i, t_{n+1}; t_l)$ . Substitution in Eq. (30) yields the following equation to calculate  $\mathbf{e}(\mathbf{x}_i, t_{n+1}; t_l)$  up to order  $O(\Delta t^3)$ :

$$\mathbf{e}(\mathbf{x}_i, t_{n+1}; t_l) = \Delta t \mathbf{u}_h(\mathbf{x}_i - \frac{1}{2}\mathbf{e}(\mathbf{x}_i, t_{n+1}; t_l), t_l + \vartheta_{nl}). \quad (33)$$

This equation has a unique solution if [37]

$$\Delta t \max_{(\mathbf{x}, t_n) \in B_i, l} |\nabla \mathbf{u}_h^n(\mathbf{x})| < 2 \quad (34)$$

where  $B_i$  is a neighborhood of the points  $\mathbf{x}_i$  such that  $\mathbf{X}(\mathbf{x}_i, t_{n+1}; t_l) \in B_i$ . Thus, assuming that  $\Delta t$  and  $\mathbf{u}_h^n$  satisfy this condition, we make use of Eq. (31) to calculate the solution of Eq. (33) by a fixed point iterative procedure that will stop when either a tolerance  $Tol$  or the maximum number of iterations,  $KMAX$ , are reached. Both  $Tol$  and the integer  $KMAX$  are supplied by the user. Then, we have that for each  $i$ :

$$\mathbf{X}(\mathbf{x}_i, t_{n+1}; t_l) = \mathbf{x}_i - \mathbf{e}(\mathbf{x}_i, t_{n+1}; t_l). \quad (35)$$

Several remarks are now in order:

- (1) Let  $\mathbf{e}(\mathbf{x}_i, t_{n+1}; t_l)$  be the  $k$ -iteration of the fixed point iterative procedure as per Eq. (33), then we need to identify the element that contains the point  $\mathbf{x}_i - \frac{1}{2}\mathbf{e}^{(k-1)}(\mathbf{x}_i, t_{n+1}; t_l)$ , and then, by finite element interpolation, evaluate

$$\mathbf{u}_h(\mathbf{x}_i - \frac{1}{2}\mathbf{e}^{(k-1)}(\mathbf{x}_i, t_{n+1}; t_l), t_l + \vartheta_{nl})$$

on such an element. We do this by using the new search-locate algorithm.

- (2) Note that for some points  $\mathbf{x}_i$ , the criterion for the convergence of the iterative procedure may impose a restriction on  $\Delta t$  that should unnecessarily penalize the size of  $\Delta t$  for the overall calculation.
- (3) It can be shown theoretically that the trajectories  $\mathbf{X}(\mathbf{x}_i, t_{n+1}; t)$  cannot leave the domain  $D$  through a solid wall because on it the velocity is zero; however, due to numerical errors, for  $\Delta t$  not sufficiently small, some trajectories leave the domain; since the numerical errors depend on the size of  $\Delta t$ , then by taking it sufficiently small we ensure that no trajectory will cross over a solid boundary.

To overcome the last two problems in the calculation of  $\mathbf{X}(\mathbf{x}_i, t_{n+1}; t_l)$ , we shall implement an adaptive version of the formulae (30)–(35) (see Fig. 7). Thus, noting that for any integer  $m \geq 1$  we can write up to order  $O(\Delta t^3)$ :

$$\mathbf{X}(\mathbf{x}_i, t_{n+1}; t_{n+1} - 2\vartheta_{nlm}) = \mathbf{x}_i - 2\vartheta_{nlm} \mathbf{u}_h(\mathbf{X}(\mathbf{x}_i, t_{n+1}; t_{n+1} - \vartheta_{nlm}), t_{n+1} - \vartheta_{nlm}), \quad (36)$$

where  $\vartheta_{nlm} = \frac{n(l)\Delta t}{2^m}$ ,  $n(l) = 2^{n-l}$ , so that, when  $l = n - 1 \Rightarrow n(l) = 2$ , and when  $l = n \Rightarrow n(l) = 1$ . Then, setting

$$\mathbf{e}(\mathbf{x}_i, t_{n+1}; t_{n+1} - 2\vartheta_{nlm}) = \mathbf{x}_i - \mathbf{X}(\mathbf{x}_i, t_{n+1}; t_{n+1} - 2\vartheta_{nlm})$$

and arguing as above, we obtain

$$\mathbf{e}(\mathbf{x}_i, t_{n+1}; t_{n+1} - 2\vartheta_{nlm}) = 2\vartheta_{nlm} \mathbf{u}_h(\mathbf{x}_i - \frac{\mathbf{e}(\mathbf{x}_i, t_{n+1}; t_{n+1} - 2\vartheta_{nlm})}{2}, t_{n+1} - \vartheta_{nlm}). \quad (37)$$

Now, applying the fixed point iteration procedure we will find the unique solution of Eq. (37) if

$$\frac{n(l)\Delta t}{2^m} \max_{(\mathbf{x}, t_n) \in B_i, l} |\nabla \mathbf{u}_h^n(\mathbf{x})| < 1. \quad (38)$$

So that, for each  $\mathbf{x}_i$  one can control the convergence of the iterative procedure by increasing  $m$ . The important point here is that different  $\mathbf{x}_i$  have different values of  $\vartheta_{nlm}$ . This same method can also be applied to prevent the trajectories from leaving the domain through solid walls. We implement this time adaptive version in Algorithm 1.

*Forward calculation of the points  $\mathbf{X}_i(t_{n+1})$*

We now calculate the position of the center of mass  $\mathbf{X}_i(t_{n+1})$ , for all  $i = 1, \dots, N_d$  dumbbells in the domain, in an analogous manner to that used when computing  $\mathbf{X}(\mathbf{x}_i, t_{n+1}; t_l)$  for  $l = n - 1$  or  $n$ . Since both methods share a similar structure and implementation, we just provide the time adaptive procedure as laid out in Algorithm 2.

**Algorithm 1.** Scheme for the adaptive calculation of  $\mathbf{X}(\mathbf{x}_i, t_{n+1}; t_l)$  when  $l = n - 1$  or  $l = n$ .



Data:  $\mathbf{x}_i$ ,  $i$ -th vertex of the triangle  $T_j$ . Tol, KMAX,  $m = 1$ .  
Result:  $\mathbf{X}(\mathbf{x}_i, t_{n-1}; t_i)$ , for  $l = n - 1$  or  $l = n$ .

```

1 for  $j = 1$  to  $NE$  do
2   for  $i = 1$  to 3 do
3      $\varepsilon_i^{(0)} = 2\vartheta_{nlm}\mathbf{u}_h(\mathbf{x}_i, t_n)$ 
4     if  $\mathbf{x}_i - \frac{\varepsilon_i^{(0)}}{2} \notin \bar{D}$  /* Through solid boundaries */
5       then
6          $m = m + 1$ ;
7         do 3
8       end
9     for  $k = 1$  to KMAX do
10       $\varepsilon_i^{(k)} = 2\vartheta_{nlm}\mathbf{u}_h\left(\mathbf{x}_i - \frac{\varepsilon_i^{(k-1)}}{2}, t_{n+1} - \vartheta_{nlm}\right)$ 
11      if  $\frac{\|\varepsilon_i^{(k)} - \varepsilon_i^{(k-1)}\|}{\|\varepsilon_i^{(k-1)}\|} > Tol \wedge \mathbf{x}_i - \frac{\varepsilon_i^{(k)}}{2} \notin \bar{D}$  /* Through solid boundaries */
12        then
13           $m = m - 1$ ;
14          do 3
15        else
16           $\mathbf{X}(\mathbf{x}_i, t_{n-1}; t_{n+1} - 2\vartheta_{nlm}) = \mathbf{x}_i - \varepsilon_i^{(k)}$ 
17        end
18        if  $\mathbf{x}_i - \varepsilon_i^{(k)} \notin D$  /* Through solid boundaries */
19          then
20             $m = m - 1$ ;
21            do 3
22          end
23        end
24      for  $s = m - 1$  to 1 do
25         $\mathbf{X}(\mathbf{x}_i, t_{n-1}; t_{n+1} - 2\vartheta_{nlm}) -$ 
26         $\mathbf{x}_i - 2\vartheta_{nlm} \cdot \mathbf{u}_h(\mathbf{X}(\mathbf{x}_i, t_{n-1}; t_{n+1} - \vartheta_{nlm}), t_{n+1} - \vartheta_{nlm})$ 
27      end
28    end

```

**Algorithm 2.** Scheme for the adaptive calculation of  $\mathbf{X}_i(t_{n-1})$ .

Data:  $\mathbf{X}_i(t_n)$ , center of mass of  $i$ -th dumbbell at  $t_n$ . Tol, KMAX,  $m = 1$ .  
Result:  $\mathbf{X}_i(t_{n+1})$ .

```

1 for  $i = 1$  to  $N_d$  do
2    $\varepsilon_i^{(0)} = \frac{2\Delta t}{2^m}\mathbf{u}_h(\mathbf{X}_i(t_n), t_n)$ 
3   if  $\mathbf{X}_i(t_n) + \frac{\varepsilon_i^{(0)}}{2} \notin \bar{D}$  /* Through solid boundaries */
4     then
5        $m = m + 1$ ;
6       do 2
7     end
8   for  $k = 1$  to KMAX do
9      $\varepsilon_i^{(k)} = \frac{2\Delta t}{2^m}\mathbf{u}_h\left(\mathbf{X}_i(t_n) + \frac{\varepsilon_i^{(k-1)}}{2}, t_n + \frac{\Delta t}{2^m}\right)$ 
10    if  $\frac{\|\varepsilon_i^{(k)} - \varepsilon_i^{(k-1)}\|}{\|\varepsilon_i^{(k-1)}\|} > Tol \wedge \mathbf{X}_i(t_n) + \frac{\varepsilon_i^{(k)}}{2} \notin \bar{D}$  /* Through solid boundaries */
11      then
12         $m = m + 1$ ;
13        do 2
14      else
15         $\mathbf{X}_i\left(t_n + \frac{2\Delta t}{2^m}\right) = \mathbf{X}_i(t_n) + \varepsilon_i^{(k)}$ 
16      end
17      if  $\mathbf{X}_i(t_n) + \frac{\varepsilon_i^{(k)}}{2} \notin \bar{D}$  /* Through solid boundaries */
18        then
19           $m = m + 1$ ;
20          do 2
21        end
22      end
23    end
24    for  $s = m - 1$  to 1 do
25       $\mathbf{X}_i\left(t_n + \frac{2\Delta t}{2^s}\right) = \mathbf{X}_i(t_n) + \frac{2\Delta t}{2^s}\mathbf{u}_h\left(\mathbf{X}_i\left(t_n + \frac{\Delta t}{2^s}\right), t_n + \frac{\Delta t}{2^s}\right)$ 
26    end
27  end

```

**Remark 6.** Note that all dumbbells leaving the domain are re-entered in it, by sampling from a parabolic distribution in the case of the planar contraction 10:1 so as to obtain their spatial position. As for their configurations, they are extracted from the initial distribution indicated in Eq. (9).

**4.3.4. Computation of the transpose of the velocity gradient tensor  $\kappa$**

The transpose of the velocity gradient,  $\kappa = (\nabla \mathbf{u}_h)^T$ , is a term that must be calculated with care because it appears in the stochastic differential equation of the elongation vector  $\mathbf{Q}$ . As such, it must satisfy certain regularity conditions to guaranty the existence and uniqueness of the solution [22]. Since  $\mathbf{u}_h \in \mathbf{V}_{h0}$  is a piecewise quadratic vector valued function, then a direct calculation of  $(\nabla \mathbf{u}_h)^T$  yields a discontinuous piecewise linear function that does not belong to  $\mathbf{S}_h$ . In order to calculate  $\kappa$  as a Lipschitz continuous function in  $\mathbf{S}_h$  we recover  $\kappa$  as the  $L^2$ -projection of  $(\nabla \mathbf{u}_h)^T$  onto  $\mathbf{S}_h$ . Thus, setting for all time  $t_n$

$$\kappa^n(\mathbf{x}) = \sum_{i=1}^{MP} \kappa_i^n \Psi_i, \quad (39)$$

where  $\kappa_i^n = (\kappa_{ml}^n(\mathbf{x}_i))$ ,  $1 \leq m, l \leq 2$ , we have that for all  $j$ ,  $1 \leq j \leq MP$ ,

$$\int_D \kappa^n \Psi_j d\mathbf{x} = \int_D (\nabla \mathbf{u}_h)^T \Psi_j d\mathbf{x}. \quad (40)$$

Hence, we calculate the tensor values  $(\kappa_{ml}^n(\mathbf{x}_i))$  by inverting the so-called mass matrix. Noting that the mass matrix is symmetric and very well conditioned, this operation is trivially done by using a preconditioned conjugate gradient method.

**4.3.5. Computation of the polymer stress tensor  $\tau_p$**

The calculation of the polymer stress tensor  $\tau_p$  at each point  $(\mathbf{x}, t)$  is done according to formula (8). In the stochastic approach, the ensemble average is approximated by averaging over the individual realizations of the stochastic process  $\mathbf{Q}(t)$  which are solutions of Eqs. (20) and (21). In doing so, we must note that each realization is the solution of the stochastic equation along the trajectory of the center of mass of a particular dumbbell; so that, by calculating such a solution we can get at each time instant  $t_m$  a discrete set of values  $\{\mathbf{Q}_i^m(\mathbf{X}_i^m) \otimes \mathbf{F}(\mathbf{Q}_i^m(\mathbf{X}_i^m))\}_{1 \leq i \leq N_d^m}$ , with the points  $\{\mathbf{X}_i^m\}$  being located in the spatial domain  $D$ . With this information we must devise an scheme to calculate a polymeric stress tensor depending on both time and space, and belonging to the finite element space  $\mathbf{S}_h$ . In this respect, it is worth mentioning some of the schemes proposed in the past; for instance, the scheme of [4] consists of dividing each element (rectangle) into two triangular subelements within which the polymer stress tensor  $\tau_p$  is calculated applying formula (8) and taking the ensemble average over the dumbbells contained in such a subelement, assigning this value to the barycenter of the triangle, so that the polymer stress is a piecewise constant function in space. In the scheme proposed in [10] the polymer stress is calculated in each element by performing a linear least squares fit to the values of the stress associated to the particles carrying the dumbbells contained in each element, so that the polymer stress is now a piecewise linear discontinuous function.

We propose in this paper the following scheme. First, we divide each triangle of the mesh into three subelements by joining its barycenter with the mid-points of its edges, the area of each one of these subelements is equal to Area of  $T/3$ . Next, for each mesh-point  $\mathbf{x}_j$  that is a vertex of a triangle, we consider the region  $A_j$  formed by the union of the subelements of all triangles sharing  $\mathbf{x}_j$  as a vertex, see Fig. 8. Let  $s$  be the number of those subelements  $A_{jl}$ ,  $1 \leq l \leq s$ , note that  $A_j = \cup_l A_{jl}$  and  $D = \cup_j A_j$ , and therefore, Area  $(A_j) = \sum_{l=1}^s$

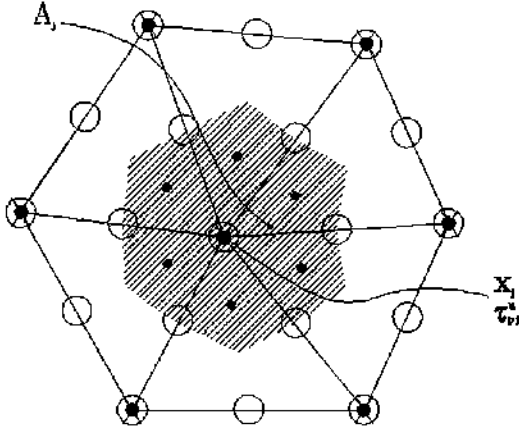


Fig. 8. Area  $A_j$  associated to the vertex  $x_i$  in the calculation of the stress  $\tau_p$ .

Area ( $A_{ji}$ ), and Area ( $D$ ) =  $\sum_{j=1}^{MP} \text{Area} (A_j)$ . Let  $N_l$  be the number of dumbbells that at time  $t_m$  are located in the subelement  $A_{ji}$ ; then, applying formula (8) we calculate

$$\tau_{p_i}^m = nk_B \Theta \left( \frac{1}{N_l} \sum_{i=1}^{N_l} \mathbf{Q}_i^m(\mathbf{X}_i^m) \otimes \mathbf{F}(\mathbf{Q}_i^m(\mathbf{X}_i^m)) - \mathbf{I} \right) \quad (41)$$

and set for all  $\mathbf{x}$  in  $A_{ji}$

$$\tau_p^m(\mathbf{x})|_{A_{ji}} = \tau_{p_i}^m.$$

This means that  $\tau_p^m(\mathbf{x})$  is a piecewise constant function that does not belong to  $S_h$ . We calculate  $\tau_{p_h}^m$  in  $S_h$  as the  $L^2$ -projection of  $\tau_p^m(\mathbf{x})$  onto  $S_h$ . Specifically, let

$$\tau_{p_h}^m(\mathbf{x}) = \sum_{j=1}^{MP} \tau_{p_j}^m \psi_j(\mathbf{x}), \quad (42)$$

where  $\tau_{p_j}^m = (\tau_{p_{qr}}^m(x_j))$ ,  $1 \leq q, r \leq 2$ , then for each  $j$

$$\int_D \tau_{p_h}^m \psi_j d\mathbf{x} = \int_D \tau_{p_m} \psi_j d\mathbf{x}, \quad (43)$$

Since  $\tau_p^m \psi_j$  is a polynomial of degree 1 in each triangle of the mesh, we can compute the integrals by means of a first order accurate quadrature rule such as the Gaussian rule of one point (barycenter of the triangles). This quadrature rule calculates exactly the right hand side of Eq. (43); as for the left hand side, the integral is equivalent to lump the mass matrix. Thus, the procedure yields

$$\tau_{p_j}^m = \frac{1}{\text{Area}(A_j)} \left( \sum_{i=1}^s \text{Area}(A_{ji}) \tau_{p_i}^m \right). \quad (44)$$

#### 4.4. An algorithmic presentation of the semi-Lagrangian micro-macro method

We have given a thorough description of our new semi-Lagrangian micro-macro method, separated into a time discretization part, a finite element formulation of the equations arising from the time discretization of the Navier–Stokes, and stochastic equations. From this description it is clear that at each time instant  $t_{n+1}$  we calculate the velocity vector  $\mathbf{u}_h^{n+1}$ , the pressure  $p_h^{n+1}$ , the elongation vectors  $\mathbf{Q}^{n+1}$ , the trajectories  $\mathbf{X}^{n+1}$  of the dumbbells,  $\tau_{p_h}^{n+1}$  and so on; however, in order to facilitate the comprehension and the existing interrelation of the different parts of the algorithm, we summarize its description in a step-by-step presentation.

- (1) Choose  $\Delta t$  and generate the mesh  $D_h$ .
- (2) Let  $N_d$  be the number of dumbbells uniformly and randomly distributed in the domain at time  $t_0$ . Assuming that  $\mathbf{u}_h^0, \{\mathbf{Q}_j^0\}_{j=1}^{N_d}$  and  $\{\mathbf{X}_j^0\}_{j=1}^{N_d}$  are known, perform the first time step to calculate  $\mathbf{u}_h^1, p_h^1, \{\mathbf{Q}_j^1\}_{j=1}^{N_d}, \{\mathbf{X}_j^1\}_{j=1}^{N_d}$  and  $\tau_{p_h}^1$ . Then
- (3) For  $n = 1, 2, \dots, N$ :
  - (3.1) For each dumbbell, apply Algorithm 2 to calculate its center of mass  $\mathbf{X}_j^{n+1}$ , the element  $T_k$  where it is located, and estimate  $\kappa^{n+1}(\mathbf{X}_j^{n+1})$  by the extrapolation formula (23)  $\kappa^{n+1}(\mathbf{X}_j^{n+1}) = 2\kappa^n(\mathbf{X}_j^{n+1}) - \kappa^{n-1}(\mathbf{X}_j^{n+1})$ .
  - (3.2) For each dumbbell, solve Eqs. (20)–(22) to obtain the elongation vectors  $\mathbf{Q}_j^{n+1}$ .
  - (3.3) Calculate the polymer stress tensor  $\tau_{p_h}^{n+1}$ .
  - (3.4) Apply the semi-Lagrangian projection algorithm to calculate  $\bar{\mathbf{u}}_h^n, \bar{\mathbf{u}}_h^{n-1}$ . Note that the semi-Lagrangian projection algorithm makes use of Algorithm 1 to calculate the departure points  $\{\mathbf{X}(\mathbf{a}_j, t_{n-l}), l = 1, 2\}$ , of the vertices ( $\mathbf{a}_j$ ) of the triangles.
  - (3.5) Solve the generalized Stokes problem (26) to calculate  $\mathbf{u}_h^{n+1}$  and  $p_h^{n+1}$ .
  - (3.6) Calculate  $\kappa^{n+1}$  by applying formulas (39) and (40).

**Remark 7.** Our method to calculate the solution of the micro-scale model is somewhat similar to the Lagrangian particle method (LPM) presented in [10], in the sense that both methods solve stochastic equations (in Lagrangian formulation) along trajectories; in our case, the trajectories correspond to the center of mass of the dumbbells, whereas in LPM the trajectories are described by the motion of particles, each one carrying a given number of dumbbells  $n_d$ . The difference between our method and LPM is, however, the following. Our method considers that a given number of dumbbells  $N_d$  is initially uniformly randomly distributed over the domain; whereas LPM considers a given number  $n_{part}$  of particles initially uniformly distributed in each element of the mesh. Since in a meaningful calculation,  $n_{part} < N_d/NE$ , then the number of trajectories to be calculated is presumably much higher in our method than in LPM, and therefore, our method may require longer time to solve the micro-scale equations. To this respect, we must say that in our method the calculations of the trajectories of the dumbbells and the location of their final positions over irregular meshes represents 20–25% of the time required to calculate the solution of the micro and macro-scale equations per time step on a single processor of a Dual Core AMD Opteron Processor 265 at 1.8 GHz, 1 MB cache, 4 GB RAM, compiled with no optimization flags; whereas the predictor-corrector algorithm spends about 40–50% of that time. Now, if one takes  $n_{part} = O(10)$ ,  $n_d = O(10^3)$  and  $N_d = (10^6)$  then, roughly speaking, our micro-macro algorithm may require about 20% more of CPU time that LPM with uncorrelated ensembles. In order to illustrate the performance of our semi-Lagrangian micro-macro method as for the CPU time is concerned, we say that the calculations on mesh  $M_1$  with  $N_d = 10^6$  and  $1.6 \cdot 10^6$  time steps need 110 days, or equivalently, an average of about 6 seconds per time step.

## 5. Results and discussion

The stability and convergence behavior of the new method were assessed by performing a series of calculations in the abrupt, planar 10:1 contraction for increasing values of the Weissenberg number  $Wi$ .

In all results reported below the calculation was started from a spatially uniform distribution of molecules, the configurations of which were initialized from the known equilibrium distribu-

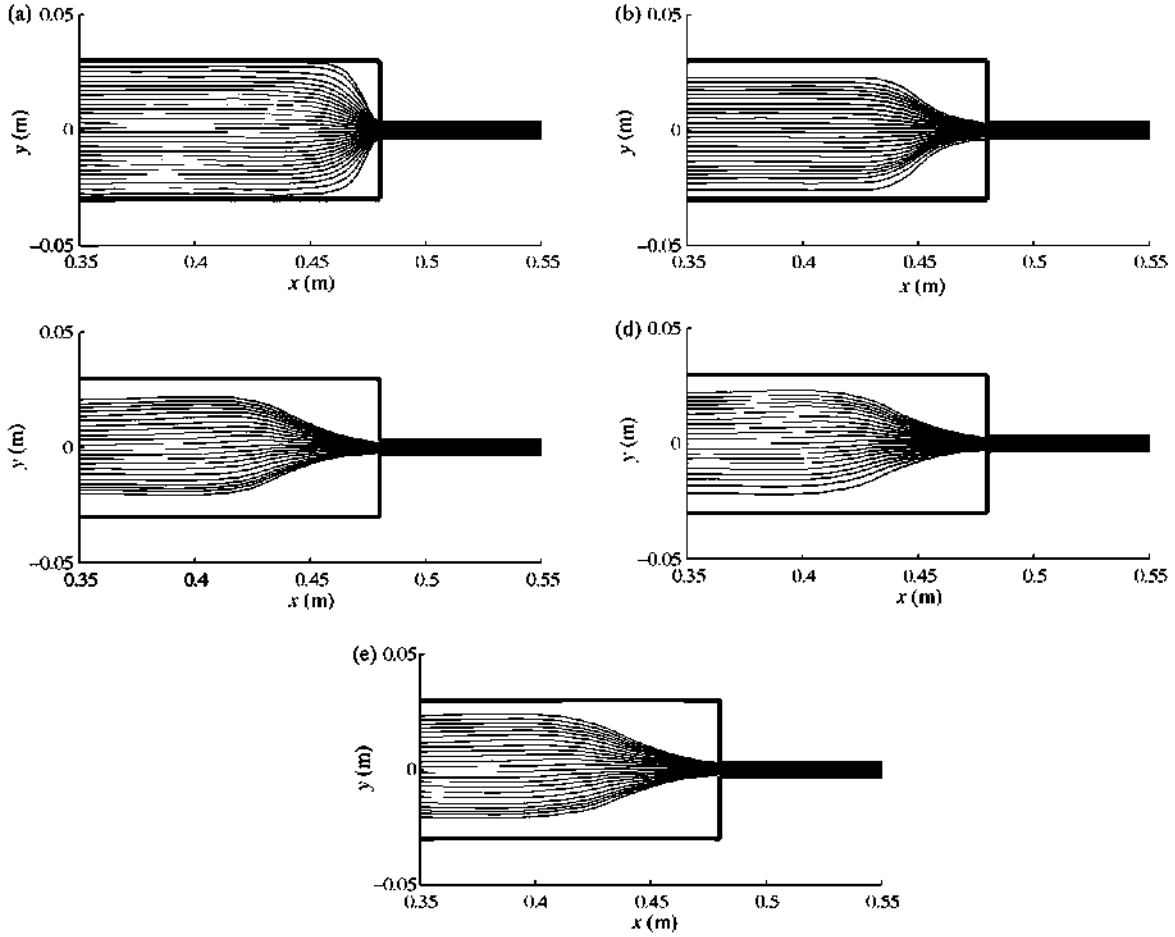


Fig. 9. Streamlines for  $Wi = 11$  (panel (a)),  $Wi = 66$  (panel (b)),  $Wi = 222$  (panel (c)),  $Wi = 333$  (panel (d)) and  $Wi = 444$  (panel (e)).

tion (9). A parabolic velocity profile was imposed at the inlet and a sufficient channel length ensured that a fully developed velocity profile was achieved three diameters upstream of the reentrant corner. Independent runs at increasingly larger values of  $Wi$  were performed and continued until  $10\lambda$ . Apart from the transient evolution, velocity, stress and molecular configuration fields at the steady state were obtained by averaging over the last five relaxation times. Outflow boundary conditions were free stress according to (3). Following [38]  $Wi$  is defined as

$$Wi = 2\lambda \frac{U}{L}, \quad (45)$$

where  $U = 6.6 \times 10^{-3}$  and  $L = 6 \times 10^{-3}$  are the mean velocity and channel width at the outflow, respectively. In all cases the following parameters and material properties were kept constant at these values:  $N_d = 10^6$ ;  $t_{end} = 10\lambda$ ;  $\eta_s = 10^{-2}$  Pa s;  $\rho = 10^3$  kg m $^{-3}$ ;  $b = 50$ ;  $\alpha = 0.2$ ;  $v_{x,max}^{in} = 10^{-3}$  m s $^{-1}$ ; where  $N_d$  is the size of the global ensemble of FENE dumbbells in the domain and  $v_{x,max}^{in}$  is the maximum in the parabolic velocity profile at the inlet. Mesh M1 is used for most calculations. The upstream/downstream channel lengths were set at 8 and 30 times the upstream/downstream channel widths, respectively.

In addition to that, two increasingly refined meshes M2, M3, and also increasingly larger molecular ensembles (between  $N_d = 10^6$  and  $N_d = 10^7$  molecules) were used for convergence tests. Table 1 shows  $MV$ ,  $MP$  and  $NE$  for each of the meshes. Time integration steps were varied between  $10^{-2}$  (mesh M1) and  $1.25 \times 10^{-3}$  (mesh M3). The symmetry of the domain with respect to its mid-plane was deliberately not exploited, and all calculations were carried out

Table 1

Velocity nodes  $MV$ , pressure nodes  $MP$  and number of elements  $NE$  for all three considered meshes M1, M2 and M3.

Mesh	$MV$	$MP$	$NE$
M1	4,511	1179	2,154
M2	17,367	4511	8,616
M3	69,737	7637	34,464

the whole domain. The reason for making the extra computational effort due to ignoring domain symmetry, was to assess the quality of the solutions obtained also with respect to symmetry, i.e. to verify that the method rigorously produced symmetric (apart from statistical fluctuations) solutions, and eventually to observe the onset of oscillatory behavior as  $Wi$  was increased.

The highly stable character of the method was confirmed by its ability to produce converged solutions up to values of  $Wi$  as high as  $Wi = 444$ , which were also stable with respect to mesh refinement. An actual  $Wi$  upper limit for convergence could not be found, since computation times increased greatly with  $Wi$  and were discontinued before any numerical instability could be found (see Table 2).

Table 2

Integration time step  $\Delta t$  as a function of the Weissenberg number  $Wi$ .

$Wi$	11	66	222	333	444
$\Delta t$ ( $\times 10^{-3}$ s)	10	5	2.5	1.25	1.25

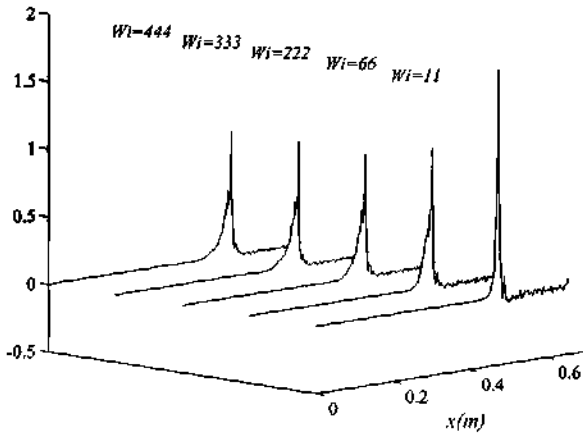


Fig. 10. Elongational component of the rate-of-strain tensor for  $Wi = 11, 66, 222, 333$  and  $444$ .

The stability of the method makes it possible to follow the growth of the vortex region at steady state with increasing  $Wi$ . From the streamline plots in Fig. 9, major changes in the streamline pattern, and appreciable growth of the recirculating regions are evident as  $Wi$  increases from  $Wi = 11$  to  $Wi = 222$ . However, as  $Wi$  goes beyond approximately 200, streamlines and vortex size remain almost constant. Velocity gradients (Fig. 10 shows the  $\dot{\gamma}_{11}$  extensional component) also saturate beyond  $Wi = 200$ .

In Fig. 11 the evolution of the vortex with  $Wi$  at steady state is characterized in a graphical way by plotting the separation line, and the migration of its stagnation point as  $Wi$  increases. Error bars indicating the statistical uncertainty in the position of the stagnation point are included in the plot for  $Wi = 4, 11, 66$ . For higher  $Wi$ , the error bar is smaller than the symbol. It can be observed that the stagnation point of the vortex (A) moves appreciably upstream and towards the center of the channel as  $Wi$  grows. The line separating the recirculating region from the flowing fluid also moves rapidly as  $Wi$  increases up to approximately 200, after which value it stalls up to the maximum value investigated ( $Wi = 444$ ). We also characterized vortex size quantitatively by the ratio  $X_C^* = 2x_C/L$  (Fig. 12), where  $x_C$  is the horizontal ( $x$ ) distance between the stagnation point on the wall and the re-entrant corner. In addition to its size, vortex intensity was also characterized by its flowrate, defined as the total volumetric flowrate being recirculated in the vortex at steady state.

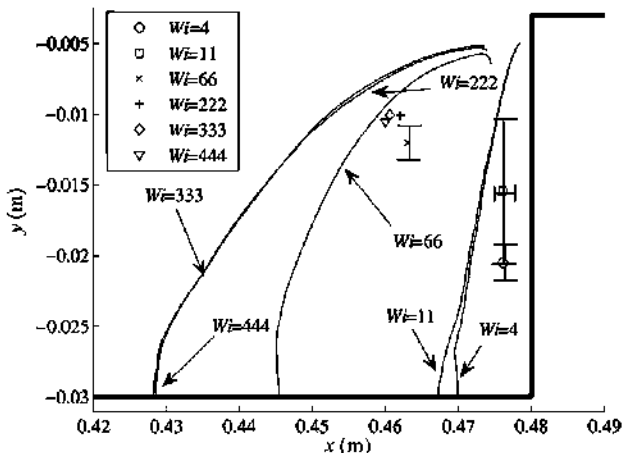


Fig. 11. Migration of vortex stagnation point for increasing values of  $Wi$ .

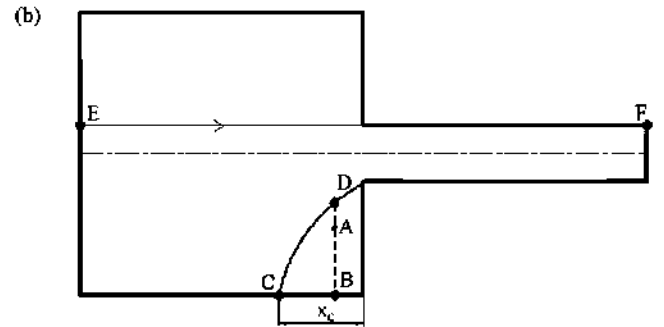
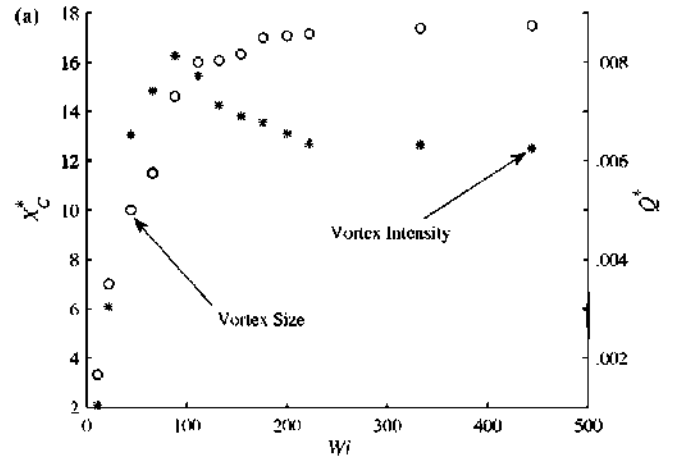


Fig. 12. Size and intensity of vortex as a function of  $Wi$ . (a) Left axis is vortex size, defined as the horizontal distance between the contraction and the detachment point of the separation line. Right axis is dimensionless vortex flowrate, as indicated by Eqs. (46) and (47). (b) Figure inset for the planar contraction.

This flowrate was computed as the integral

$$Q_{\text{vortex}} = \left| \int_A^B \mathbf{u} \cdot \mathbf{n} ds \right| \quad (46)$$

over the straight line from (internal vortex stagnation point) to (on the channel wall, immediately below A), as indicated in the inset of Fig. 12(b), where  $\mathbf{u}$  is the local velocity and  $\mathbf{n}$  is the local unit vector normal to the line A-B. Likewise, in Fig. 12(a) we have considered the dimensionless vortex intensity

$$Q^* = \frac{Q_{\text{vortex}}}{Q_{\text{inlet}}}, \quad (47)$$

with  $Q_{\text{inlet}} = 4 \times 10^{-3} \text{ m}^2 \text{ s}^{-1}$  being the total flow entering the channel. It is remarkable that, while vortex size grows monotonically and reaches almost asymptotic behavior after  $Wi \simeq 200$ , its intensity measured by (46), goes through a maximum at  $Wi \simeq 100$  and subsequently decreases to an asymptotic value of about 80% of its maximum value. In all cases, the recirculating flowrate in the vortex remains below 1% of the fluid flowrate traversing the contraction, although the extent occupied by the vortex grows appreciably with  $Wi$ .

The increase in vortex flowrate, which is a global, integral measure, is a direct consequence of major changes in the velocity profiles in the vortex. It was found however that the velocity field within the vortex maintains a high degree of geometric self-similarity over the whole range of  $Wi$  investigated, as can be seen in Fig. 13. In this figure, the horizontal velocity profile along the line B-A-D (as defined in Fig. 12) is plotted as a function of the ordinate, made dimensionless by dividing through the distance between the points B and D. Horizontal velocity  $u_x$  is also made dimensionless by dividing through  $u_x$  at point D. Although vortex size and intensity

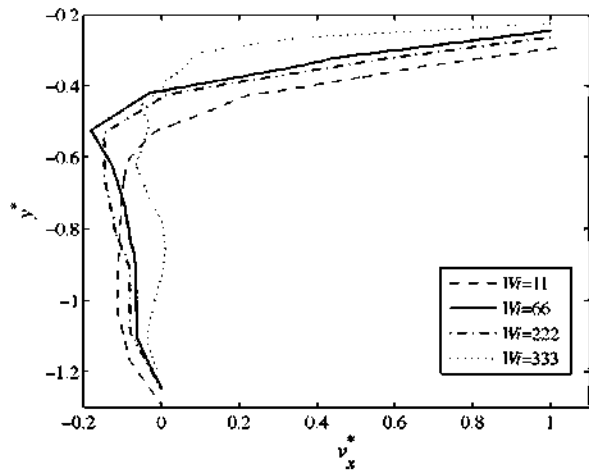


Fig. 13. Circulation velocity profile in vortex as a function of  $Wi$  along line A-B, where A is the stagnation point of the vortex (determined at steady state) and B lies on the domain boundary.

vary appreciably and in a very non-linear fashion, their velocity profiles virtually collapse on a single curve when suitably made dimensionless.

It also follows from the previous quantitative vortex description that, as vortex grows, the shape of the effective domain through which the non-stagnant fluid flows (i.e. total domain minus vortex) departs more and more from the shape of an abrupt contraction, and tends to resemble a rectangular-triangular (cylindrical-conical in axisymmetric contraction flow) channel.

In spite of this loss in cross sectional area close to the contraction, the total pressure drop (Fig. 14) decreases monotonically, i.e. since the volumetric flow rate is constant, the vortex has a mildly drag-reducing effect for the contraction flow. Drag reduction is caused by the smoother effective flow domain and by the shear thinning behavior of the FENE fluid, but partly also by the replacement of a no-slip boundary condition at the rigid wall by a velocity-continuity boundary condition at the vortex separation line.

As a consequence of the less abrupt effective contraction, the influence of  $Wi$  on the velocity profile along the symmetry axis is to damp the peak at the contraction proper (Fig. 15; in each panel of this figure, a single error bar has been plotted at  $x \approx 0.6$  as a measure of the size of statistical fluctuations).

The modulus of the FENE polymer contribution to the stress tensor along line E-F (inset Fig. 12(b)) at steady state is plotted in Fig. 16

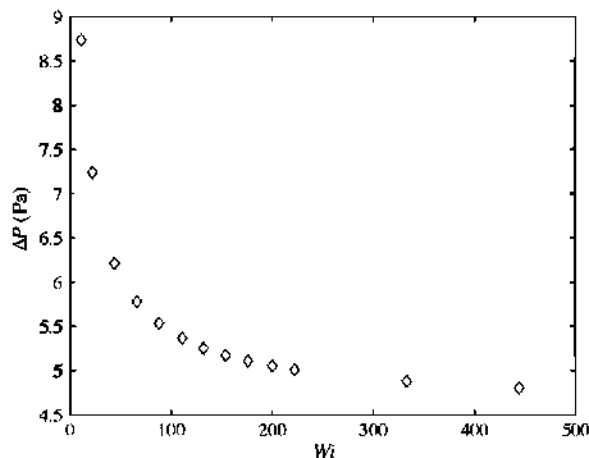


Fig. 14. Pressure drop across entire channel as a function of  $Wi$ .

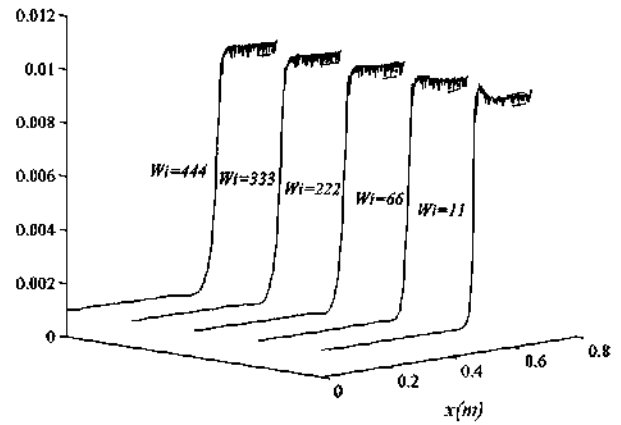


Fig. 15. Axial velocity profile for  $Wi = 11, 66, 222, 333$  and  $444$ .

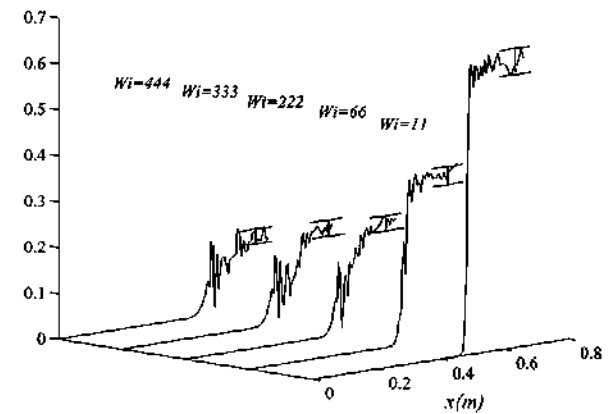


Fig. 16. Modulus of FENE polymer contribution to the stress tensor along line E-F for  $Wi = 11, 66, 222, 333$  and  $444$ .

for five selected values of  $Wi$ . Again, the strongest variation with  $Wi$  found up to approximately  $Wi = 200$ . The shear contribution (Fig. 17) shows similar behavior. It is noticeable that while  $\tau_{p12}$  downstream decreases with  $Wi$  due to shear-thinning, the large spike in  $\tau_{p12}$  at the contraction proper ( $x \approx 0.48$ ) caused by the large velocity gradient close to the re-entrant corner has an almost constant intensity of 0.06.

In Fig. 18, the contour lines of the modulus of the FENE polymer contribution  $|\tau_p|$  to the stress tensor are plotted in the neighbor-

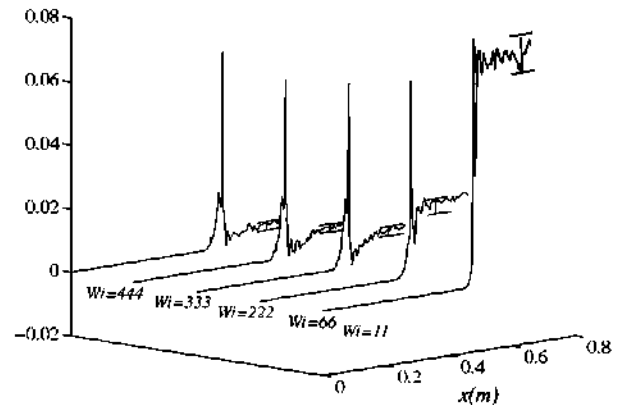


Fig. 17. Shear component of FENE polymer contribution to the stress tensor ( $\tau_p^{12}$ ) (Pa) along line E-F (see inset of Fig. 12(b)) for  $Wi = 11, 66, 222, 333$  and  $444$ .

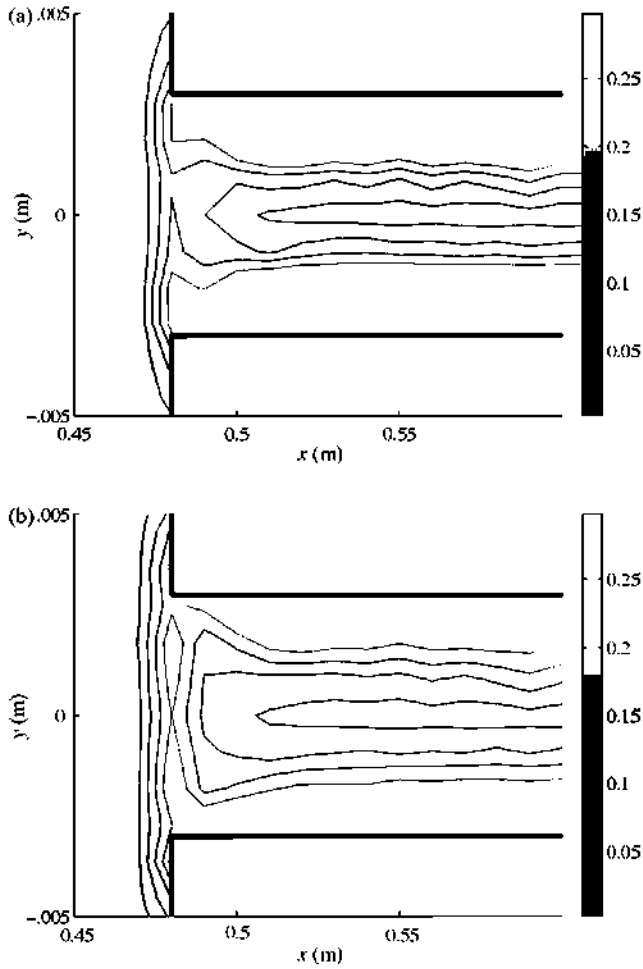


Fig. 18. Isolines of modulus of FENE polymer contribution to the stress tensor ( $\tau_{p12}$ ) (Pa) in the neighborhood of the contraction for  $Wi = 11$  (a) and  $Wi = 444$  (b).

hood of the contraction. An interesting qualitative change takes place as  $Wi$  increases from  $Wi = 66$  (panel (b)) to  $Wi = 222$  (panel (c)). Maximum values of polymer stress are found, as expected, at the contraction. At low  $Wi$ , two local maxima are found symmetrically located about the centerline, at  $y \approx \pm 0.002$ . These maxima are caused by FENE molecules flowing close the walls in the upstream region and thus undergoing large shear flow. However, as  $Wi$  increases, the two maxima merge into a single, centrally located maximum. This maximum in the modulus of the polymer stress comes now from the elongational contribution of FENE molecules flowing along the centerline and being subjected to strong elongational flow. This qualitative change at high  $Wi$  could not be observed in previous studies [4] in which only the two symmetric maxima were detectable before the numerical method failed to converge. The merger of the two local maxima into a single, central one is a consequence of the change in shape of the effective flowing domain, whose contraction (defined by the separation lines) becomes drastically less abrupt. This geometric change leads to a diminished shear strain rate intensity for molecules flowing closer to the wall, and to an increased extensional strain rate intensity for molecules flowing along the centerline.

Pressure contour line plots (Fig. 19) show the already observed [4] transition from approximately Newtonian ( $Wi = 11$ ) to a distinctly viscoelastic pressure field ( $Wi = 444$ ).

For the highest value of the Weissenberg number  $Wi = 444$ , we have represented the  $x$  and  $y$  components of the configuration vector  $\mathbf{Q}$  at four different positions of the 60% of height stream-

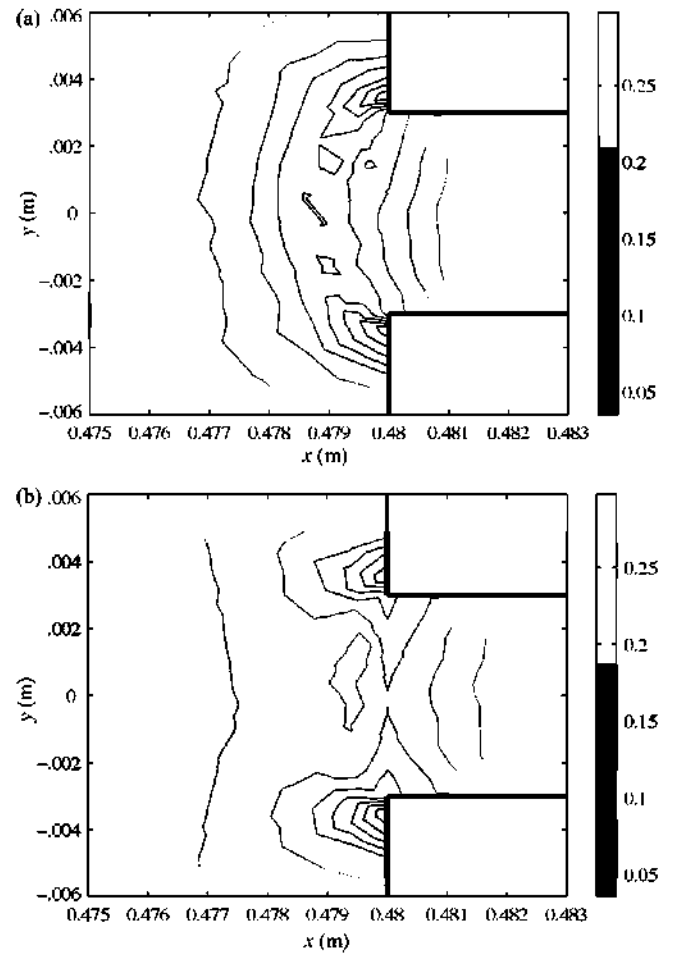


Fig. 19. Isolines of pressure in the neighborhood of the contraction for increasing values of  $Wi = 11$  (a) and  $Wi = 444$  (b).

line (initially at  $(0, 0.018)$ ) of a cloud of  $N = 5 \times 10^3$  dumbbells being evolved with the velocity field (see inset of Fig. 20(e)). The combined effects of shear and elongational stresses as the cloud progresses through the contraction can be noticed in Fig. 20(a-d). Just after passing the contraction, panel (c), a majority of the dumbbells in the cloud are extended close to their maximum length according to the FENE parameter  $b = 50$ , as can be seen in Fig. 21.

In this figure we show the probability density of the (square) end-to-end distance  $|\mathbf{Q}|^2$  at each selected position of the cloud of dumbbells, with the left vertical axis displaying the density for (a) and (b), and the right vertical axis being applied to (c) and (d), so as to highlight the difference in the scales. It is interesting to note how the distribution in Fig. 20(b) (just before reaching the contraction) evolves into the very sharply peaked distribution at point (c) (right after the contraction). This is a consequence of the very strong extensional character of the flow in the vicinity of the sudden contraction. It is also remarkable how the characteristic sigmoidal cloud of configuration points (Fig. 20(a)) evolves into the straight, almost horizontal cloud (panel (c)) due to the strong extensional flow. Axes scales in this latter panel differ by a factor of 10, so that the configurational cloud is virtually horizontal. Most dumbbell connector vectors are almost parallel to the symmetry axis of the exit channel. Further down the exit channel, connectors retain their high elongation, but the configurational cloud tends to regain the sigmoidal shape characteristic of a simple shear flow.

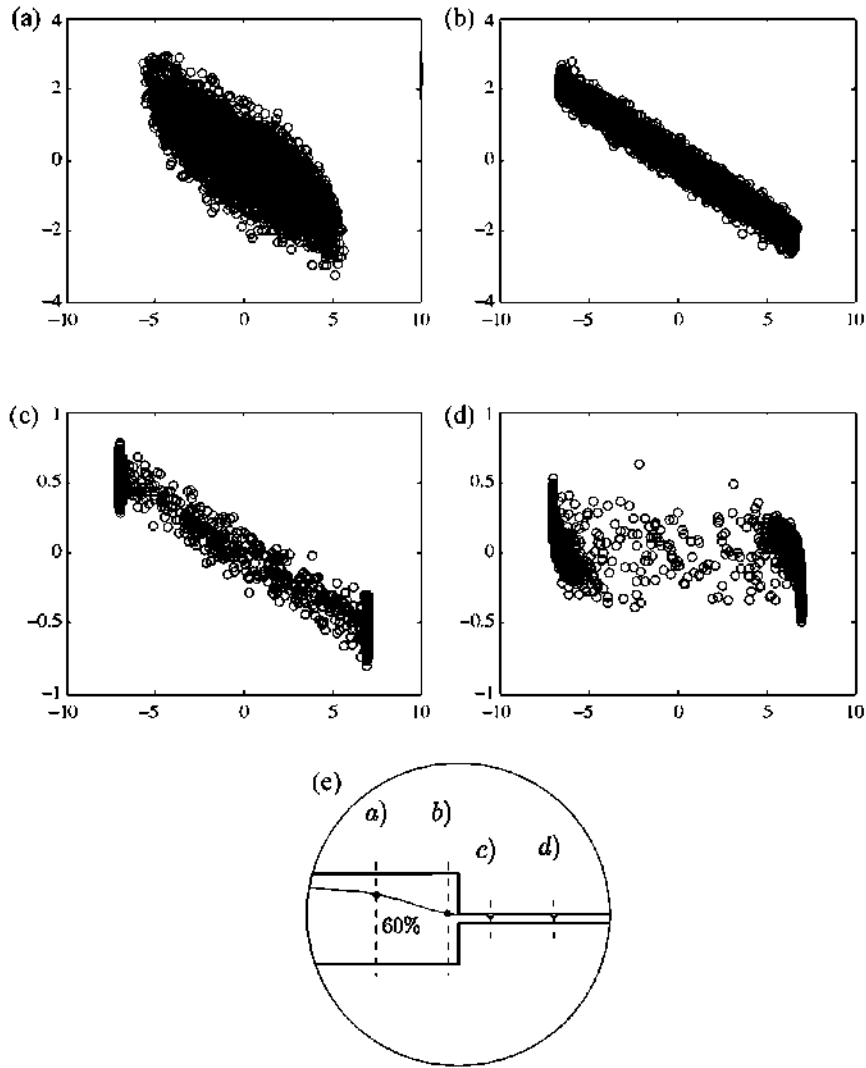


Fig. 20.  $Q_x$  (horizontal axis) and  $Q_y$  (vertical axis) at different positions (a) (0.4259, 0.01604), (b) (0.473, 0.00365), (c) (0.5011, 0.0023) and (d) (0.5432, 0.00229) of the 60% streamline. Inset of the 60% streamline and selected positions (e).

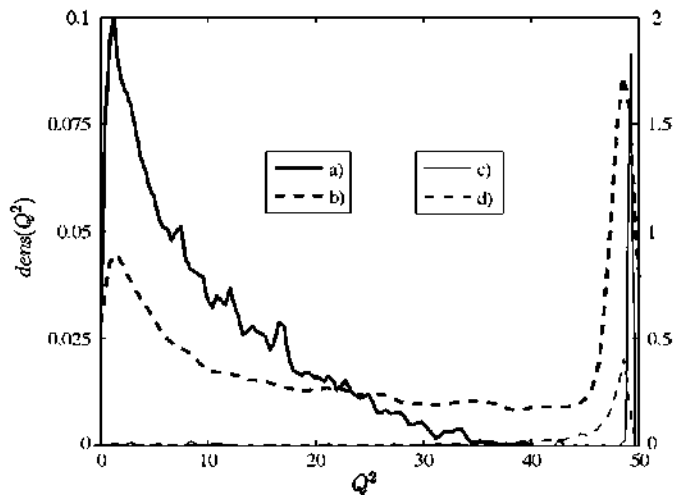


Fig. 21. Probability density of the square of the modulus of configuration vector  $Q$  at different positions: (a and b) (left vertical axis); (c and d) (right vertical axis). Notice different scales on both vertical axes.

## 6. Conclusions

In this paper, a new semi-Lagrangian micro-macro method for viscoelastic flow calculations has been presented. The method makes use of the semi-Lagrangian projection method to reduce the Navier-Stokes equations to a Stokes problem, employing quadratic polynomials for the velocity and linear for both pressure and polymer stress tensor; a new search-locate algorithm to cope with particles scattered over unstructured, unconventional meshes has been devised as well. For the stochastic equations of the micro-scale model, the second order predictor-corrector scheme presented in [22] is applied along the forward trajectories of the center of mass of the dumbbells. The polymer stress tensor is computed using a projection method that employs piecewise constant functions over an "influence" region around each mesh node.

The method is tested on a benchmark problem, namely, the planar contraction flow 10:1. The stable character of the method allows to compute at Weissenberg numbers as high as 444, showing remarkable elastic effects and a profound change in the rheological behavior of the fluid.

## Acknowledgments

Financial support by the EC through contracts G5RD-CT-2002-00720 and NMP3-CT-2005-016375, partial support by CICYT grant

MAT2006-04029, and support from the Barcelona Supercomputing Center and the high performance computational cluster Magerit at CesViMa UPM is acknowledged.

## References

- [1] M. Laso, H.C. Öttinger, Calculation of viscoelastic flow using molecular models: the CONNFESSIT approach, *J. Non-Newtonian Fluid Mech.* 47 (1993) 1–20.
- [2] K. Feigl, M. Laso, H.C. Öttinger, The CONNFESSIT approach for solving a two-dimensional viscoelastic fluid problem, *Macromolecules* 28 (1995) 3261–3274.
- [3] M. Laso, M. Picasso, H.C. Öttinger, 2-D time-dependent viscoelastic flow calculations using CONNFESSIT, *AIChE J.* 43 (4) (1997) 877–892.
- [4] M. Laso, M. Picasso, H.C. Öttinger, Calculation of flows with large elongational components: CONNFESSIT calculation of the flow of a FENE fluid in a planar 10:1 contraction, in: H. Kausch, H. Nguyen (Eds.), *Flexible Polymer Chains in Elongational Flow*, Springer, Berlin, 1999, pp. 101–136.
- [5] C.C. Hua, J.D. Schieber, Application of kinetic theory models in spatiotemporal flows for polymer solutions, liquid crystals and polymer melts using the CONNFESSIT approach, *Chem. Eng. Sci.* 51 (1996) 1473–1485.
- [6] J. Cormenzana, A. Ledda, M. Laso, B. Debbaut, Connfessit calculation of free surface viscoelastic flows, *J. Rheol.* 45 (1) (2001) 237–258.
- [7] E. Grande, M. Laso, M. Picasso, Calculation of variable-topology free surface flows using CONNFESSIT, *J. Non-Newtonian Fluid Mech.* 113 (2003) 127–145.
- [8] M.A. Hulsen, A.P.G. van Heel, B.H.A.A. van den Brule, Simulation of viscoelastic flows using Brownian configuration fields, *J. Non-Newtonian Fluid Mech.* 70 (1997) 79–101.
- [9] P.G. Gigras, B. Khomami, Adaptive configuration fields: a new multiscale simulation technique for reptation-based models with a stochastic strain measure and local variations of life span distribution, *J. Non-Newtonian Fluid Mech.* 108 (2002) 99–122.
- [10] P. Halin, G. Lielens, R. Keunings, V. Legat, The Lagrangian particle method for macroscopic and micro-macro viscoelastic flow computations, *J. Non-Newtonian Fluid Mech.* 79 (1998) 387–403.
- [11] P. Wapperom, R. Keunings, V. Legat, The backward-tracking Lagrangian particle method for transient viscoelastic flows, *J. Non-Newtonian Fluid Mech.* 91 (2000) 273–295.
- [12] R. Keunings, Micro-macro methods for the multiscale simulation of viscoelastic flow using molecular models of kinetic theory, in: D.M. Binding, K. Walters (Eds.), *Rheology Reviews*, British Society of Rheology, 2004, pp. 67–98.
- [13] 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 (1999) 215–246.
- [14] J.P. Aguayo, P.M. Phillips, T.N. Phillips, H.R. Tamaddom-Jahromi, B.A. Snigetev, M.F. Webster, The numerical prediction of planar viscoelastic contraction flows using the pom-pom model and higher-order finite volume schemes, *J. Comput. Phys.* 220 (2007) 586–611.
- [15] A. Machmoum, D. Esselaoui, Finite element approximation of viscoelastic fluid flow using characteristics method, *Comput. Methods Appl. Mech. Eng.* 190 (2001) 5603–5618.
- [16] M. Fortin, D. Esselaoui, A finite element procedure for viscoelastic flows, *Int. J. Numer. Methods Fluids* 7 (1987) 1035–1052.
- [17] R.B. Bird, O. Hassager, R.C. Armstrong, C.F. Curtiss, *Dynamics of Polymeric Liquids*, vol. 2, Kinetic Theory, 2nd ed., John Wiley and Sons Inc., New York, 1987.
- [18] J. Von Neumann, Various techniques used in connection with random digits, *Nat. Bureau Stand.* 12 (1951) 36–38.
- [19] R.E. Knop, Random vectors uniform in solid angle, *Commun. ACM* 13 (1970) 326–327.
- [20] M. Matsumoto, T. Nishimura, Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator, *ACM Trans. Modell. Comput. Simul.* 8 (1998) 3–30.
- [21] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, M. Booth, F. Rossi, *GNU Scientific Library Reference Manual*, 3rd ed., Network Theory Limited, 2009, <http://www.gnu.org/software/gsl/>.
- [22] H.C. Öttinger, *Stochastic Processes in Polymeric Fluids: Tools and Examples for Developing Simulation Algorithms*, Springer, Berlin, 1996.
- [23] W.P. Petersen, A general implicit splitting for stabilizing numerical simulations of Ito stochastic differential, *SIAM J. Numer. Anal.* 35 (1998) 1439–1451.
- [24] E.J. Dean, R. Glowinski, On some finite element methods for the numerical simulation of incompressible viscous flow, in: M.D. Gunzburger, R.A. Nicolaides (Eds.), *Incompressible Computational Fluid Dynamics*, Cambridge University Press, New York, 1993, pp. 109–150.
- [25] J. Douglas, T.F. Russell, Numerical methods for convection dominated diffusion problems based on combining the method of the characteristics with finite element or finite difference procedures, *SIAM J. Numer. Anal.* 19 (1982) 871–885.
- [26] O. Pironneau, On the transport diffusion algorithm and its applications to the Navier–Stokes equations, *Numer. Math.* 38 (1982) 309–332.
- [27] E. Süli, Convergence and nonlinear stability of the Lagrange–Galerkin method for the Navier–Stokes equations, *Numer. Math.* 53 (1988) 459–483.
- [28] K.W. Morton, A. Priestley, E. Süli, Stability of the Lagrange–Galerkin method with non-exact integration, *M<sup>2</sup>AN Math. Model. Numer. Anal.* 22 (1988) 625–653.
- [29] J.P. Benque, G. Labadie, J. Ronat, A new finite element method for the Navier–Stokes equations coupled with a temperature equation, pp. 295–301, in: T. Kawai (Ed.), *Proceedings of the 4th International Symposium on Finite Element Methods in Flow Problems*, North-Holland, Amsterdam, 1982.
- [30] A. Robert, A stable numerical integration scheme for the primitive meteorological equations, *Atmos. Ocean*, 19 (1981) 35–46.
- [31] R. Bermejo, J. Carpio, A semi-Lagrangian–Galerkin projection scheme for convection equations, *IMA J. Numer. Anal.*, doi:10.1093/imanum/dnn044.
- [32] A.H. Stroud, *Approximate Calculation of Multiple Integrals*, Prentice-Hall Inc, Englewood Cliffs, NJ, 1971.
- [33] A. Allievi, R. Bermejo, A generalized particle search-locate algorithm for arbitrary grids, *J. Comput. Phys.* 132 (1997) 157–166.
- [34] A. Allievi, R. Bermejo, Finite element modified method of characteristics for the Navier–Stokes equations, *Int. J. Numer. Methods Fluids* 32 (2000) 439–464.
- [35] M. Falcone, R. Ferretti, Convergence analysis for a class of higher-order semi-Lagrangian advection schemes, *SIAM J. Numer. Anal.* 61 (1998) 909–940.
- [36] C. Temperton, A.N. Staniforth, An efficient two-time level semi-Lagrangian semi-implicit integration scheme, *Q. J. R. Meteorol. Soc.* 113 (1987) 1025–1039.
- [37] L.M.G. Gutiérrez, R. Bermejo, A semi-Lagrangian level set method for incompressible Navier–Stokes equations with free surface, *Int. J. Numer. Methods Fluid*, 49 (10) (2005) 1111–1146.
- [38] T.N. Phillips, R.G. Owens, *Computational Rheology*, Imperial College Press, 2002.