Direct numerical simulation of complex viscoelastic flows via fast lattice-Boltzmann solution of the Fokker-Planck equation

Luca Bergamasco, Salvador Izquierdo, Amine Ammar

▶ **To cite this version:**

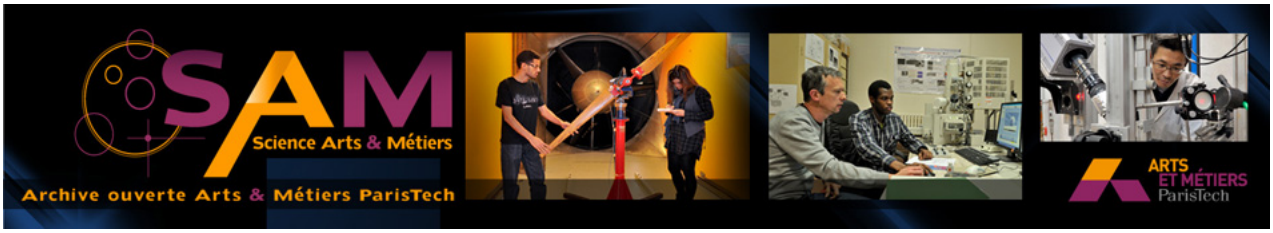
Luca Bergamasco, Salvador Izquierdo, Amine Ammar. Direct numerical simulation of complex viscoelastic flows via fast lattice-Boltzmann solution of the Fokker-Planck equation. Journal of Non-Newtonian Fluid Mechanics, 2013, 201, pp.29-38. 10.1016/j.jnnfm.2013.07.004 . hal-01061178

HAL Id: hal-01061178
https://hal.science/hal-01061178v1

Submitted on 5 Sep 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Science Arts & Métiers (SAM)

is an open access repository that collects the work of Arts et Métiers ParisTech researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: http://sam.ensam.eu
Handle ID: .http://hdl.handle.net/10985/8462

To cite this version :

Luca BERGAMASCO, Salvador IZQUIERDO, Amine AMMAR - Direct numerical simulation of complex viscoelastic flows via fast lattice-Boltzmann solution of the Fokker–Planck equation - Journal of Non-Newtonian Fluid Mechanics - Vol. 201, p.29-38 - 2013

Any correspondence concerning this service should be sent to the repository
Administrator : archiveouverte@ensam.eu

Direct numerical simulation of complex viscoelastic flows via fast lattice-Boltzmann solution of the Fokker–Planck equation

L. Bergamasco [a], S. Izquierdo [a,*], A. Ammar [b]

[a] Instituto Tecnológico de Aragón (ITA), María de Luna 8, 50018 Zaragoza, Spain
[b] Arts et Métiers ParisTech, Boulevard du Ronceray, BP 93525, F-49035 Angers cedex 01, France

Keywords:
Multi-scale
Finite volume method
Lattice Boltzmann method
FENE kinetic model
GPU computing

A B S T R A C T

Micro–macro simulations of polymeric solutions rely on the coupling between macroscopic conservation equations for the fluid flow and stochastic differential equations for kinetic viscoelastic models at the microscopic scale. In the present work we introduce a novel micro–macro numerical approach, where the macroscopic equations are solved by a finite-volume method and the microscopic equation by a lattice-Boltzmann one. The kinetic model is given by molecular analogy with a finitely extensible non-linear elastic (FENE) dumbbell and is deterministically solved through an equivalent Fokker–Planck equation. The key features of the proposed approach are: (i) a proper scaling and coupling between the micro lattice-Boltzmann solution and the macro finite-volume one; (ii) a fast microscopic solver thanks to an implementation for Graphic Processing Unit (GPU) and the local adaptivity of the lattice-Boltzmann mesh; (iii) an operator-splitting algorithm for the convection of the macroscopic viscoelastic stresses instead of the whole probability density of the dumbbell configuration. This latter feature allows the application of the proposed method to non-homogeneous flow conditions with low memory-storage requirements. The model optimization is achieved through an extensive analysis of the lattice-Boltzmann solution, which finally provides control on the numerical error and on the computational time. The resulting micro–macro model is validated against the benchmark problem of a viscoelastic flow past a confined cylinder and the results obtained confirm the validity of the approach.

1. Introduction

One of the most commonly adopted practices for the simulation of dilute polymeric suspensions relies on macroscopic constitutive equations for the polymeric extra stress, derived from molecular models and solved via well-established numerical methods [1]. The advantage of this approach is the low computational cost associated, the drawback is that some kinetic models does not have a closed-form continuous counterpart. With regards to the finitely extensible non-linear elastic (FENE) model for example, a rheological law can only be derived under closure approximations, i.e. FENE-P, FENE-LS [2]. The resulting models are then able to phenomenologically describe the basic flow features but the underlying theoretical assumptions can hinder the retrieval of relevant viscoelastic phenomena.

In a more general modeling strategy, the kinetic origin of the molecular models is retained [3]. Methods using this approach are generally described as micro–macro models, due to the separated solution of the micro and macroscales. Continuity and momentum equations are solved using continuous equations (macro-scale) and kinetic equations are solved by stochastic or deterministic methods (micro-scale) [4]. In this framework, one of the most popular methodologies is the CONNFFESSIT approach, where a finite element solution of the macroscopic equations is combined with stochastic simulations for the dumbbell configuration [5]. One of the major issues concerned with this approach is the high computational expense and the embedded statistical noise, which can be filtered using variance reduction techniques [6]. Another similar and commonly used approach is the Brownian configuration field method [7]. This method already embeds efficient variance reduction, as long as individual molecules are clustered in continuous configuration fields according to their initial configuration and applied force, but the computational cost of the stochastic simulation is anyway a limit.

An alternative approach for noise reduction and faster computations consists in the solution of an equivalent Fokker–Planck equation for the probability density of the dumbbell configuration. However, a literature review reveals that due to the dimensionality of the problem and the lack of efficient numerical methods to solve the Fokker–Planck equation, little progress has been done in this framework [4] and no method prevail. Relevant recent work about the direct solution of the Fokker–Planck equation for complex

* Corresponding author.
E-mail addresses: lbergamasco@ita.es (L. Bergamasco), sizquierdo@ita.es (S. Izquierdo), amine.ammar@ensam.eu (A. Ammar).

flows relies on a Galerkin spectral element technique for 2D [8] and its extension to 3D [9]. Another group of promising methods are those that approximate the solution of the Fokker–Planck equation reducing the dimensionality of the problem. This order-reduction can be done *a priori*, like in the lattice-Fokker–Planck method [10], *on line* like in the proper generalized decomposition [11] or *a posteriori* like in the proper orthogonal decomposition [12]. All these techniques aim to systematically reduce the degrees of freedom and therefore the computational expense.

In this work we focus on direct deterministic numerical methods, therefore no approximation occurs beyond mesh resolution. The proposed approach relies on a previous work by Ammar [13] about a lattice Boltzmann solution of the Fokker–Planck equation for homogeneous flows. Recently this method has been also theoretically analyzed [14] and applied for the solution of a population balance equation [15] and for the Fokker–Planck equation [16]. However, none of the previous works [13–16] deals with the coupling of the kinetic solution with macroscopic fields, thus we investigate efficient ways to exploit it in multi-scale simulations.

In the proposed micro–macro model, the macroscopic equations are solved by a finite-volume method using the commercial solver ANSYS Fluent® v14.0, while the microscopic equation is solved by a lattice-Boltzmann method. The Fokker–Planck equation is solved using an operator-splitting procedure that allows to solve the configurational part by a GPU implementation of the lattice Boltzmann method and the physical convection by a finite volume method. The operator-splitting indeed allows us to transport only viscoelastic stresses instead of the whole distribution function defined in the configuration space. Consequently, algorithms with low-memory requirements can be formulated.

The outline of the paper is as follows: the governing equations for the polymeric suspension and a derivation of the stochastic equation for the FENE dumbbell model are firstly presented; successively, the equivalent Fokker–Planck equation is introduced (Section 2). In Section 3, the solution and coupling strategy is detailed together with the numerical methods. Section 4 comprises the numerical analysis of the sub-grid solution, the validation of the coupled model and its optimization. The details of the GPU implementations and the relative coupling with the macroscopic solver are reported in Appendix C. A brief summary of the results obtained and an outlook on further developments concludes the paper (Section 5).

2. Theoretical model

2.1. Hydrodynamic system

Let us consider a polymeric solution as a blend between a Newtonian and a viscoelastic fluid. Assuming the flow to be incompressible and isothermal, mass and momentum conservation reads:

$$\nabla_x \cdot \mathbf{v} = 0; \tag{1}$$

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho \mathbf{v} \cdot (\nabla_x \mathbf{v}) = -\nabla_x p + \nabla_x \cdot \boldsymbol{\sigma}; \tag{2}$$

where ρ is the density, p the pressure, $\mathbf{v}$ the velocity vector and the subscript x denotes operators in the physical space. The total stress tensor $\boldsymbol{\sigma}$, embeds contributions from both the Newtonian solvent $\boldsymbol{\sigma}_s$ and the polymeric solute $\boldsymbol{\sigma}_p$, therefore $\boldsymbol{\sigma} = \boldsymbol{\sigma}_s + \boldsymbol{\sigma}_p$. Denoting by μ_s the dynamic viscosity of the solvent, $\boldsymbol{\sigma}_s$ is given as:

$$\boldsymbol{\sigma}_s = \mu_s(\nabla_x \mathbf{v} + (\nabla_x \mathbf{v})^\dagger) = \mu_s \dot{\boldsymbol{\gamma}}; \tag{3}$$

being $\dot{\boldsymbol{\gamma}}$ the rate of strain tensor. In order to close the hydrodynamic system, an additional material model must be solved for the viscoelastic contribution $\boldsymbol{\sigma}_p$.

2.2. Viscoelastic model

In the simplest micro-mechanical approach for polymer rheology, molecular chains are modeled by two beads and a spring connector, that is by a non-rigid dumbbell immersed in a fluid. A general kinetic model can then be derived considering the equations of motion of the beads in the dumbbell, namely the equilibrium of inertial, frictional, Brownian and connector forces [17]. For a jth bead located in $\mathbf{r}_i$, the equilibrium yields the so called *Langevin* equation:

$$m_j \frac{d}{dt}\left(\frac{d\mathbf{r}_j}{dt} - \mathbf{v}(\mathbf{r}_j)\right) = \zeta_j\left(\frac{d\mathbf{r}_j}{dt} - \mathbf{v}(\mathbf{r}_j)\right) + \sigma\frac{d\mathbf{W}_j}{dt} + \mathbf{F}^c_j; \tag{4}$$

with m being the mass of the bead, ζ a drag coefficient, σ a coefficient for the standard Wiener process $\mathbf{W}$ and $\mathbf{F}^c$ the connector force. Indicating with k_B the Boltzmann constant and T the absolute temperature, $\sigma = \sqrt{2k_B\zeta T}$ from the principle of equipartition of energy [1]. Assuming high friction regime and thus over-dumped Brownian dynamics [18], the inertial term on the left-hand side can be dropped and, indicating with $\boldsymbol{\xi} = \mathbf{r}_2 - \mathbf{r}_1$ the end-to-end vector of a dumbbell, yields the following (Itô) stochastic differential equation:

$$\frac{d}{dt}\boldsymbol{\xi} = \boldsymbol{\kappa}\cdot\boldsymbol{\xi} - \frac{2}{\zeta}\mathbf{F}^c(\boldsymbol{\xi}) + \sqrt{\frac{4k_BT}{\zeta}}\frac{d}{dt}\mathbf{W}; \tag{5}$$

where $\mathbf{W}$ is a standard Brownian motion $(\mathbf{W}_2 - \mathbf{W}_1)/\sqrt{2}$ and the symbol $\boldsymbol{\kappa}$ has been adopted for the transpose of the velocity gradient tensor $(\nabla_x \mathbf{v})^\dagger$. The peculiarity of the dumbbell model lies in the expression of the connector force law $\mathbf{F}^c(\boldsymbol{\xi})$. In this work we are concerned with the finitely extensible non-linear elastic model, therefore indicating with h the spring constant and ξ_0 a finite extensibility parameter, the connector force reads:

$$\mathbf{F}^c(\boldsymbol{\xi}) = \frac{h}{1 - \|\boldsymbol{\xi}\|^2/\xi_0^2}\boldsymbol{\xi}; \tag{6}$$

with $\|\cdot\|$ indicating vector norm. This entropic force law, originally proposed by Warner [19], exhibits linear behavior for small extensions and the finite length ξ_0 in the limit of an infinite force. In a stochastic approach, Eq. (5) should then be stochastically solved for the dumbbell configurations in the random process $\mathbf{W}$ with the spring force law (6).

Using stochastic analysis, the ordinary differential Eq. (5) can be associated with a partial differential equation for a probability density function (PDF), which can then be deterministically solved instead of a large number of realizations for the Brownian driver. In this case the resulting probability density function $\psi(\mathbf{x}, \boldsymbol{\xi}, t)$ satisfies the Fokker–Planck equation [20]:

$$\frac{\partial \psi}{\partial t} + \mathbf{v}\cdot(\nabla_x\psi) + \nabla_\xi \cdot \left[\left(\boldsymbol{\kappa}\cdot\boldsymbol{\xi} - \frac{2}{\zeta}\mathbf{F}^c(\boldsymbol{\xi})\right)\psi\right] = \frac{2k_BT}{\zeta}\nabla^2_\xi\psi; \tag{7}$$

which is also called *Smoluchowski* equation in polymer science. Index ξ on operators indicates that they act in the configuration space. Due to its dimensionality, the solution of Eq. (7) is non-trivial and we proceed as detailed in the next section.

3. Numerical methods

3.1. Solution strategy

In order to solve the Fokker–Planck equation directly, we consider a time-splitting-like procedure similar to that proposed by Lozinski and Chauvière [8]. Following this idea, the operators acting in the configuration space are separated from those acting in the physical space. In this way Eq. (7) can be firstly solved in the configuration space for an intermediate distribution function ψ^{n^*},

which is then used for the solution in the physical space. We adopt a mixed explicit/implicit framework:

$$\frac{\psi^{n^*}-\psi^n}{\Delta t_\xi} = -\nabla_\xi \cdot \left[\left(\boldsymbol{\kappa}\cdot\boldsymbol{\xi} - \frac{2}{\zeta}\mathbf{F}^c(\boldsymbol{\xi})\right)\psi^n\right] + \frac{2k_BT}{\zeta}\nabla^2_\xi\psi^n; \tag{8}$$

$$\frac{\psi^{n+1}-\psi^{n^*}}{\Delta t_x} + \mathbf{v}\cdot(\nabla_x\psi^{n+1}) = 0; \tag{9}$$

thus Eq. (7) reduces to an advection–diffusion equation in the configuration space (8) and an advection equation in physical space (9). Let us now firstly focus on Eq. (8): the space scaling is achieved considering a relaxation time $\theta = \zeta/4h$ and a dimensionless finite extensibility parameter $b = \xi_0^2 h/k_BT$, therefore $\boldsymbol{\xi}$ is made dimensionless with $\sqrt{k_BT/h}$, $\boldsymbol{\kappa}$ with θ^{-1} and time with θ, thus the resulting dimensionless equation reads:

$$\frac{\psi^{n^*}-\psi^n}{\Delta\tilde{t}_{\tilde{\xi}}} = -\nabla_{\tilde{\xi}}\cdot\left[\left(\tilde{\boldsymbol{\kappa}}\cdot\tilde{\boldsymbol{\xi}} - \frac{1}{2}H(\tilde{\xi})\tilde{\boldsymbol{\xi}}\right)\psi^n\right] + \tilde{\alpha}\nabla^2_{\tilde{\xi}}\psi^n; \tag{10}$$

with the dimensionless diffusion coefficient being $\tilde{\alpha} = 1/2$ (see Section 3.3). From now on, the convection vector of ψ in the configuration space (terms in round brackets on right-hand side of Eq. (10)) will be indicated with $\mathbf{u}$ for convenience. The reader should notice that the scaling of the velocity gradient tensor $\boldsymbol{\kappa}$ represents the link between the physical velocity field and the convection vector $\mathbf{u}$ through the relaxation time θ of the polymer. On the basis of this consideration, we define a microscopic (or local) Weissenberg number that will be used later, based on the second invariant of the rate of strain tensor as:

$$\tilde{\boldsymbol{\kappa}} = \boldsymbol{\kappa}/\theta^{-1} \rightarrow Wi_m := \theta\sqrt{\frac{1}{2}\dot{\boldsymbol{\gamma}}:\dot{\boldsymbol{\gamma}}}. \tag{11}$$

Using the same scaling parameters as above, the connector force law $\mathbf{F}^c(\boldsymbol{\xi})$ in Eq. (6) is also made dimensionless as:

$$H(\tilde{\xi}) = \frac{1}{1-\|\tilde{\boldsymbol{\xi}}\|^2/b}. \tag{12}$$

Eq. (10) with the connector force law (12) is therefore the final dimensionless equation to be solved in the configuration space. We assume the dumbbells to be always laying in the same plane, therefore the configuration space is two-dimensional and the dumbbell extensibility domain (support of the PDF) results in a disc of radius $\sqrt{b}$. Eq. (10) is solved for a solution of ψ^{n^*} for the local convection vector $\mathbf{u}$ at each point in a domain. The details about the numerical method together with its optimization will be extensively discussed later. The obtained intermediate ψ^{n^*} should be convected in physical space by Eq. (9) according to the second stage of the operator-splitting procedure. However, we note that the convection of the full PDF in Eulerian framework would require a prohibitively amount of data to be stored and transported. This issue can be overcome considering that the final target for the hydrodynamic system is the viscoelastic stress tensor. Therefore, we proceed by computing an intermediate stress tensor, which is convected in physical space in place of the relative distribution. Being $\langle\langle\cdot\rangle\rangle$ the ensemble averaging operator, the intermediate dimensionless viscoelastic stress tensor $\tilde{\boldsymbol{\sigma}}_p^{n^*}$ is calculated from ψ^{n^*} using the Kramers expression [3]:

$$\tilde{\boldsymbol{\sigma}}_p^{n^*} = \langle\langle H(\tilde{\xi})\tilde{\boldsymbol{\xi}}\otimes\tilde{\boldsymbol{\xi}}\rangle\rangle - \mathbf{I} = \int_{\|\tilde{\boldsymbol{\xi}}\|^2<b}\psi^{n^*}(H(\tilde{\xi})\tilde{\boldsymbol{\xi}}\otimes\tilde{\boldsymbol{\xi}})d\tilde{\boldsymbol{\xi}} - \mathbf{I}. \tag{13}$$

Applying the ensemble average to Eq. (9) yields:

$$\frac{\tilde{\boldsymbol{\sigma}}_p^{n+1}-\tilde{\boldsymbol{\sigma}}_p^{n^*}}{\Delta t_x} + \mathbf{v}\cdot\left(\nabla\tilde{\boldsymbol{\sigma}}_p^{n+1}\right) = 0. \tag{14}$$

In the iterative solution adopted, this procedure is formally equivalent to the convection of the PDF before computing stresses. Furthermore, the conservation of stresses is analogous to the conservation of the second order moment of the distribution, which is actually the target quantity. The advantage of this approach is that the second stage for the solution of the Fokker–Planck Eq. (9), reduces to the convective transport of three scalar quantities, one for each component of the symmetric stress tensor.

Finally, the dimensionless stress tensor is scaled-up to its corresponding in the physical space, to serve as volumetric source term in the momentum Eq. (2). Indicating with n_c the number of polymer chains per unit volume, an equivalent polymer viscosity can be defined as $\mu_p = \theta n_c k_B T$ and the extra stress is scaled as [21]:

$$\boldsymbol{\sigma}_p = \frac{\mu_p}{\theta}\left(\frac{b+4}{b}\right)\tilde{\boldsymbol{\sigma}}_p^{n+1}. \tag{15}$$

3.2. Finite volume method

The macroscopic governing Eqs. (1) and (2) and the transport of stresses (14) are solved by finite volume method (FVM). In this approach, transport equations are numerically solved on a discretized computational domain (mesh) and the conserved variables are calculated at cell centers. Partial differential equations are therefore converted to algebraic equations by integration about the cells (or control volumes), for example Eq. (2):

$$\int_{V_c}\rho\frac{\partial\mathbf{v}}{\partial t}dV + \oint\rho\mathbf{v}\cdot(\nabla_x\mathbf{v})d\mathbf{A} = \oint(-\nabla_x p + \nabla_x\cdot\sigma)d\mathbf{A}. \tag{16}$$

Eq. (16) is then applied to each control volume and its neighboring cells in the domain, resulting in a system of algebraic equations with sparse coefficient matrix to be solved. Fluxes at cell faces, which are required for convective terms, can then be interpolated using several numerical schemes: we adopt a third order quadratic upwind scheme (QUICK) for momentum (2) and transport of stresses (14) and a second order scheme for pressure interpolation. The Semi-Implicit Method for Pressure Linked Equations (SIMPLE) is chosen for the pressure-velocity coupling. For the sake of clarity, we remark that despite the hyperbolic nature of Eq. (14), the solution is sufficiently smooth to be solved with a third order scheme. The interested reader can refer for example to [22] for details on the methods.

3.3. Lattice Boltzmann method

The advection-diffusion equation for the FENE model (10) is solved by lattice Boltzmann method (LBM). This mesoscopic approach relies on the Boltzmann transport equation, whose discrete form in the Bhatnagar–Gross–Krook (BGK) approximation of the collision operator, reads as [23]:

$$f_i(\tilde{\boldsymbol{\xi}}+\mathbf{c}_i\delta\tilde{t},\tilde{t}+\delta\tilde{t}) - f_i(\tilde{\boldsymbol{\xi}},\tilde{t}) = -\frac{1}{\tau}(f_i(\tilde{\boldsymbol{\xi}},\tilde{t}) - f_i^{eq}(\tilde{\boldsymbol{\xi}},\tilde{t})); \tag{17}$$

with $\delta\tilde{t}$ being the time step, f_i the discrete particle distribution functions and $\mathbf{c}_i$ the associated microscopic velocity vectors. The equilibrium distribution function f_i^{eq} can be derived, for example, via second-order Taylor expansion in the Mach number of the Maxwell–Boltzmann equilibrium [24]:

$$f_i^{eq} = \left(1 + \frac{\mathbf{c}_i\mathbf{u}}{c_s^2} + \frac{(\mathbf{c}_i\mathbf{u})^2}{2c_s^4} - \frac{\|\mathbf{u}\|^2}{2c_s^2}\right)\omega_i\psi; \tag{18}$$

where c_s is the lattice speed of sound that, indicating with $\delta\tilde{\xi}$ the lattice spacing and thus $c = \delta\tilde{\xi}/\delta\tilde{t}$ the lattice speed, is defined as $c_s = c/\sqrt{3}$. The reader should notice that in this case we retain the tilde notation for space and time for analogy with the equation being solved (10), but rigorously we should consider dimensionless lattice units. Macroscopic quantities can be recovered from the moments of the distribution function:

$$\psi = \sum_i f_i = \sum_i f_i^{eq}; \tag{19}$$

$$\psi \mathbf{u} = \sum_i \mathbf{c}_i f_i^{eq}; \tag{20}$$

$$\psi(\mathbf{uu} + c_s^2 \mathbf{I}) = \sum_i \mathbf{c}_i \mathbf{c}_i f_i^{eq}; \tag{21}$$

which also allow to recover the macroscopic Eq. (10) by multi-scale expansion and thus the following expression for the relaxation time (see Appendix A for details):

$$\tau = \frac{\tilde{\alpha}}{\delta \tilde{t} c_s^2} + \frac{1}{2}. \tag{22}$$

Given the advective–diffusive nature of Eq. (10), the numerical solution can be carried out on two lattice topologies, D2Q9 and D2Q5 (Fig. 1). The lattice constants for both stencils can be found in Appendix B. The domain length l is imposed to be 20% larger then the domain of existence of the PDF, therefore indicating with N the number of nodes, the lattice spacing $\delta\tilde{\xi}$ is given by l/N.

3.4. Coupled numerical algorithm

The numerical solution of the coupled model has been carried out using the commercial CFD code ANSYS Fluent® v14.0. The lattice Boltzmann solution is called at cell centers as a sub-grid routine via compiled-C user defined function (UDF). The numerical procedure can be summarized as follows:

1. solution of the governing equations for **v** and p by finite volume method: Eqs. (1) and (2);
2. sub-grid lattice Boltzmann solution of the FENE kinetic equation: Eq. (10);
3. computation of the local viscoelastic stress tensor: Eq. (13);
4. convective transport of the viscoelastic stresses by finite volume method: Eq. (14);
5. addition of the extra-stress to the momentum equation: (Eq. 2).

The procedure is iteratively repeated until global convergence. The internal convergence criterion for the FVM iterations (step 1 and 4 of the above algorithm) is a 10^{-8} residual, while for global convergence (between step 5 and 1 of the next loop) is 10^{-4}. The standard test case of the viscoelastic flow around a confined cylinder (used for the validation, see Section 4.2) has a steady-state solution. Thus, a proper choice of the time steps, both in the configurational and physical space, is needed. The configurational-space time step Δt_ξ is the one needed to reach the steady state of the local viscoelastic stress, which is dynamically checked in each cell. It should be noticed that several internal time steps in lattice units $\delta\tilde{t}$ are needed to reach Δt_ξ. The physical-space time step Δt_x is chosen in order to obtain an equilibrium between accuracy and performance, as done in other operator-splitting approaches [9].

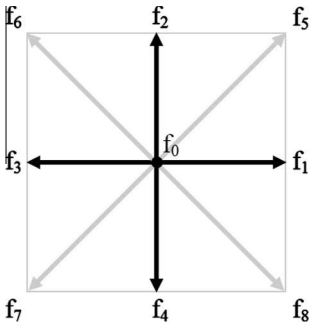


Fig. 1. Lattices and relative discrete distribution functions: five links for D2Q5 (black color) and four additional links for D2Q9 (gray diagonals).

4. Results and discussion

4.1. Sub-grid solution analysis

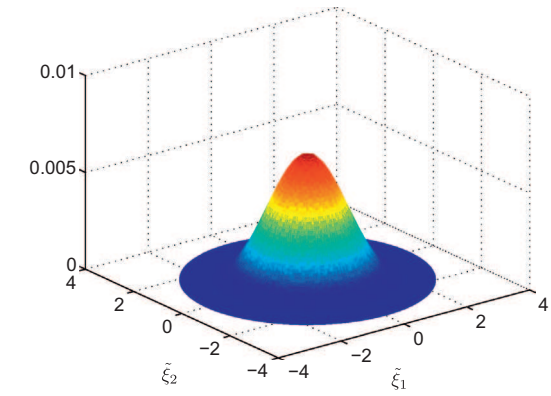
In this section, the indexes introduced in Section 3.1 are omitted for readability. We analyze the sub-grid solution of the Fokker–Planck equation by a lattice Boltzmann method in terms of: its relaxation towards equilibrium, evolution to steady-state solutions, numerical errors, stability range and computational time.

The relaxation of the probability density $\psi(\tilde{\xi}, \tilde{t})$ to equilibrium is tested considering that for null velocity gradient $\tilde{\boldsymbol{\kappa}}$, an analytical solution for Eq. (10) can be found in the form [13]:

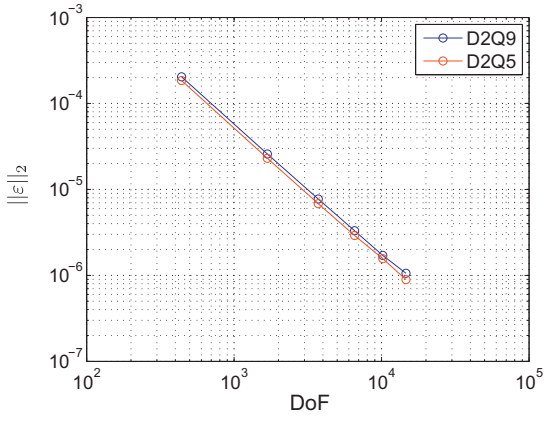
$$\psi_{eq} = \frac{H(\tilde{\xi})^{-b/2}}{\int H(\tilde{\xi})^{-b/2} d\tilde{\xi}}; \tag{23}$$

which for a dimensionless dumbbell extensibility $b = 10$ (constant throughout the paper), yields the equilibrium distribution shown in Fig. 2(a). Given an initial distribution function ψ_0 (constant in this case), satisfying the normality condition $\int \psi(\tilde{\xi}) d\tilde{\xi} = 1$, the relaxation rate and error convergence are analyzed by an ℓ_2 – *norm* with respect to the reference solution ψ_{eq} defined as:

$$\|\varepsilon\|_2 = \frac{1}{N} \sum_{k=1}^{N} \sqrt{\psi_{eq}^2 - \psi_t^2}; \tag{24}$$



(a)



(b)

Fig. 2. Shaded surface of the analytical equilibrium PDF (Eq. (23)) on a 1,681 DoF lattice (a) and ℓ_2-*norm* convergence of ψ with $\tau = 0.55$ (b).

being $\psi_{\tilde{t}}$ the distribution function at time $\tilde{t}$. The convergence criterion for relaxation is a 10^{-8} residual calculated as backward finite difference on the norm. The analysis for the two lattices shows that the error of the 8-neighbors lattice is slightly larger than that of the 4-neighbors one (see Fig. 2(b)).

With regards to the analysis of non-null $\tilde{\boldsymbol{\kappa}}$ gradient, we examine the time evolution of the shear stress $\tilde{\sigma}_{p_{xy}}$ for a start-up planar Couette flow $[0, \tilde{\kappa}_{xy}; 0, 0]$. The initial distribution function ψ_0 is in this case (as in the rest of the paper) given as Eq. (23). According to its definition (11), in this case the local Weissenberg number corresponds to the magnitude of the component $\tilde{\kappa}_{xy}$ itself. The resulting steady-state PDF for $Wi_m = 5$ is shown in Fig. 3(a), while the stress evolution for varying Wi_m is shown in Fig. 3(b). The error analysis has been carried out for $Wi_m = 1$ and $Wi_m = 5$ and is shown in Fig. 4(a). To allow a proper visualization of the comparison, the shear stress has been normalized using the value obtained with the highest number of nodes $\tilde{\sigma}_{p_{xy}}/\tilde{\sigma}^{ref}_{p_{xy}}$ for each case. Notice as for a higher Wi_m the solution requires a higher number of nodes to converge, in particular for the D2Q9 lattice. This behavior can be associated with the shape of the PDF at steady-state, that for lower Wi_m is closer to the initial condition. More details about error analysis of lattice Boltzmann methods for Fokker–Planck equations can be found in [13,16].

An analysis of the stability domains for the two tested lattices has been also carried out for Wi_m in the range 1–10. The results show that the stability range of the D2Q9 is larger than that of the D2Q5 lattice in the region of low Wi_m and high DoF

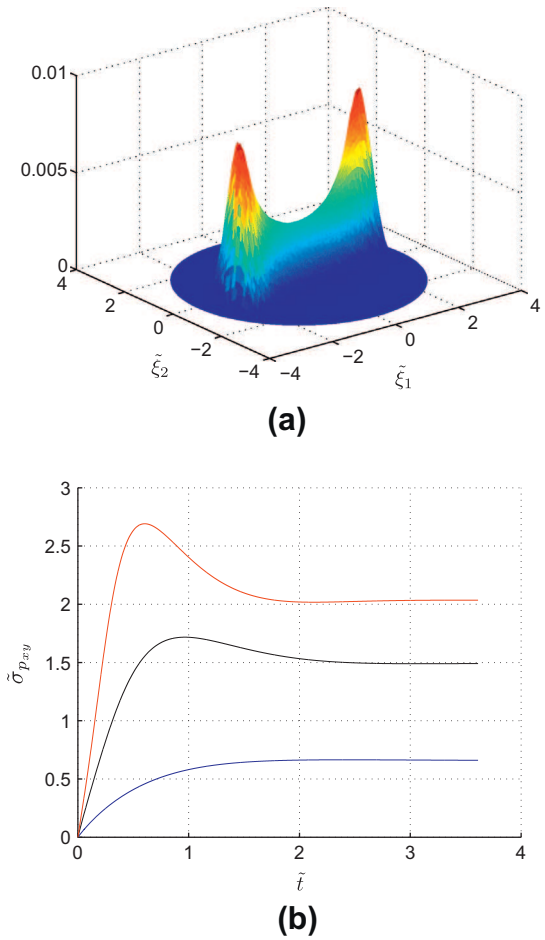


Fig. 3. Start-up plane Couette flow: shaded surface of the equilibrium PDF for $Wi_m = 5$ on a 1681 DoF lattice (a) and dimensionless shear stress evolution $\tilde{\sigma}_{p_{xy}}$ for different Wi_m on a D2Q9 lattice with 3721 DoF and $\tau = 0.55$ (b).

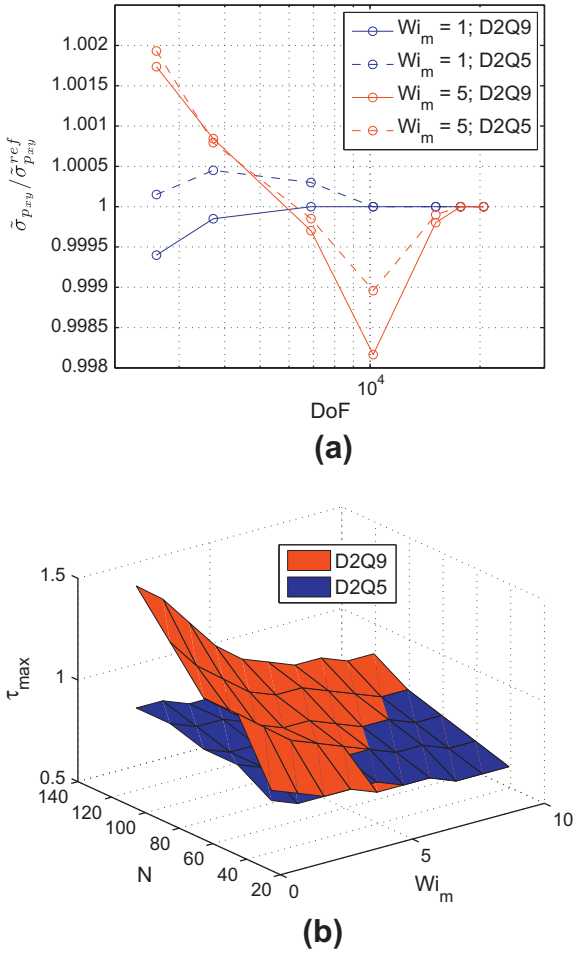


Fig. 4. Error convergence for $Wi_m = 1$ and 5 on the two lattices (a) and stability map for D2Q9 and D2Q5 lattices (b).

(Fig. 4(b)). Despite the increased stability, the error of the 8-neighbors lattice is also slightly larger than that of the 4-neighbors one (Fig. 4(a)).

Table 1 shows the comparison of the computational time required by the two lattices to converge to steady-state for the start-up plane Couette flow at $Wi_m = 5$, using the minimum relaxation time $\tau = 0.55$ and the maximum stable allowed on the basis of the stability map. The tested CPU is an Intel© Xeon© X5650 2.67 GHz. The D2Q5 lattice requires less computational time due to the reduced number of links and therefore of computational operations, however the speed-up for the D2Q9, when moving from τ_{min} to τ_{max}, is greater due to the larger stability range. The relative numerical error introduced increasing the relaxation time

Table 1
Comparison of the computational time [s] and relative numerical error [%] for the two lattices for $\tau = 0.55$ and $\tau = \tau_{max}$ (start-up plane Couette flow at $Wi_m = 5$).

Stencil	DoF=N^2	1681	3721	6561	10,201	14,641
D2Q9	$\tau = 0.55$	0.98	4.88	15.24	36.86	80.54
	$\tau = \tau_{max}$	0.33	0.87	2.48	4.63	8.43
	speed-up	3.0	5.6	6.1	8.0	9.6
	error	0.5357	−0.1933	−0.1538	−0.1690	−0.1092
D2Q5	$\tau = 0.55$	0.62	3.11	6.45	9.93	14.24
	$\tau = \tau_{max}$	0.2	0.63	1.93	3.92	7.03
	speed-up	3.1	4.9	3.4	2.5	2.0
	error	0.0764	−0.0594	−0.0297	−0.0347	−0.0198

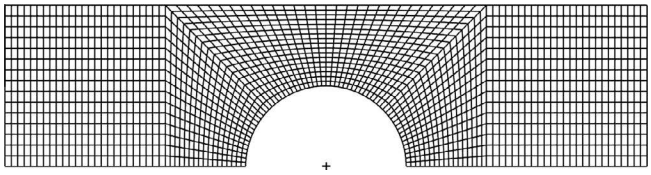


Fig. 5. Mesh layout close to the cylinder surface (2 m length and 1770 cells displayed).

for the two lattices is anyway always lower than 1% and the maximum speed-up achievable is nearly ten times for the D2Q9 lattice.

4.2. Coupled model

The coupled model is validated against a commonly adopted benchmark problem: two-dimensional viscoelastic flow around a cylinder confined between two parallel plates [1]. In order to save in computational time, only half of the domain is studied and symmetry conditions are applied on the lower boundaries. The domain extent is 4 m length (L), 0.5 m height (H) and the hole is 0.25 m radius (R) centered in the origin. The ratio of the radius of the cylinder to the half-width of the channel (blockage) has been chosen to be $\Lambda = 0.5$ and the ratio of the solvent to the total zero-shear-rate viscosity is $\beta = \mu_s/(\mu_s + \mu_p) = 0.59$ [9,21]. The mesh layout close to the cylinder surface is shown in Fig. 5, where only 2 m length and 1770 cells are displayed to allow a proper visualization of the mesh layout. The boundary conditions are: stream-wise periodicity between inlet and outlet; no-slip for momentum and homogeneous Neumann for convection of stresses on the hole and upper boundary (walls). Indicating with $\langle\cdot\rangle$ the volume-averaging operator, we define for this problem a macroscopic (or global) Weissenberg and Reynolds number based on the average velocity at inlet (or outlet):

$$Wi_M = \frac{\langle \mathbf{v} \rangle}{R}\theta = \langle \dot{\gamma} \rangle \theta; \quad Re_M = \frac{\rho \langle \mathbf{v} \rangle R}{\mu}. \tag{25}$$

The Reynolds number is kept constant to 10^{-3} for steady-state creeping flow. In order to test the FVM mesh independence, the solution has been carried out on three different grids, respectively of 15,000 (M1), 25,000 (M2) and 40,000 (M3) cells for $Wi_M = 0.6$. The number of nodes and relaxation time for the sub-grid solution are $N = 128$ and $\tau = 0.8$ (D2Q5). The obtained profiles of dimensionless viscoelastic stresses on the symmetry plane and on the cylinder surface are consistent with those obtained by Chauvière and Lozinski [9] with a Galerkin spectral element method for the 2D case (Fig. 6). We also report the contours of dimensionless dumbbell elongations in the domain (Fig. 7) and of the dimensionless viscoelastic stresses (Fig. 8). These latter are reported also for the case $Wi_M = 0.9$ in Fig. 9. As further validation we compare a drag coefficient defined as follows:

$$C_D = \frac{F_x}{4\pi R(\mu_s + \mu_p)\langle \mathbf{v} \rangle}; \tag{26}$$

with F_x being the total drag force on the cylinder surface (with polar angle ϑ):

$$F_x = 2\int_0^{\pi} \left[\left(-p + 2\mu_s \frac{\partial v_x}{\partial x} + \sigma_{pxx} \right) \cos\vartheta + \left(\mu_s \left(\frac{\partial v_y}{\partial x} + \frac{\partial v_x}{\partial y} \right) + \sigma_{pxy} \right) \sin\vartheta \right] R d\vartheta. \tag{27}$$

In order to compare the results with those obtained by Chauvière and Lozinski [9], the drag factor C_D is split into pressure C_D^p, viscous C_D^{μ} and viscoelastic contributions $C_D^{\sigma_p}$. The analysis (Table 2) show very good agreement for $Wi_M = 0.6$, the relative error on the total drag factor is below 1%. On the other hand, for $Wi_M = 0.9$ the error is around 2.5%. We noticed that despite a converged solution can be achieved with increasing Wi_M, the accuracy decreases. The cause of this decrease in accuracy can be sought in the discretization of the PDF in cartesian coordinates with the lattice-Boltzmann method and on the choice of the time-step. In this work we limit the maximum Wi_M to 0.9 and leave a deeper analysis of this issue for future work.

The sub-grid solution can be called from Fluent® via user defined function implementation and eventually parallelized on multiple processors. For computational efficiency, we use an accelerated version running on graphic card (GPU). The GPU is less flexible on the choice of the number of nodes, but provides remarkable acceleration: in this case the speed-up reaches nearly 50x with respect to the CPU (see Appendix C for details).

Let us now focus more in detail on the solution for $Wi_m = 0.6$. As derived from the analysis in Section 4.1, the error and numerical performance depends on the local Wi_m. The parameters for the sub-grid solution can then be adjusted according to the local Weissenberg number in the domain (Fig. 10). For this case, the local Weissenberg ranges between 0 and 9, with the highest frequencies between 0 and 2 and tail between 2 and 9. The parameters N and τ can then be chosen according to the following criteria:

1. *Minimize numerical error:* high lattice resolution and minimum relaxation time are to be used. This approach assures converged solution throughout the domain but non-homogeneous numerical error. The computational cost is high due to excessive number of nodes in $low - Wi_m$ regions.

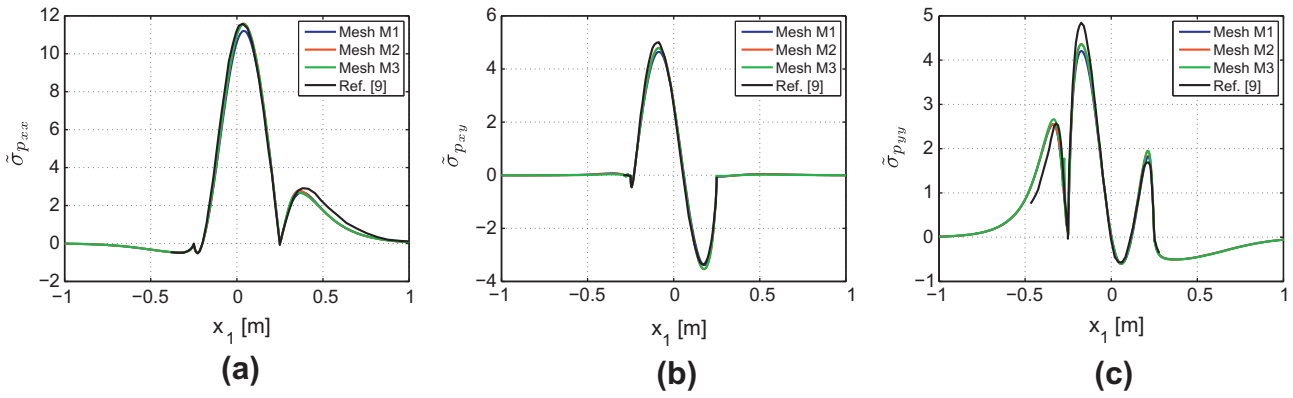


Fig. 6. Profiles of the dimensionless viscoelastic stresses on the symmetry plane and on the cylinder surface for $Wi_M = 0.6$. The results for the three tested FVM meshes (M1, M2 and M3) are compared with ref. [9]: (a) $\tilde{\sigma}_{p_{xx}}$; (b) $\tilde{\sigma}_{p_{xy}}$; and (c) $\tilde{\sigma}_{p_{yy}}$.

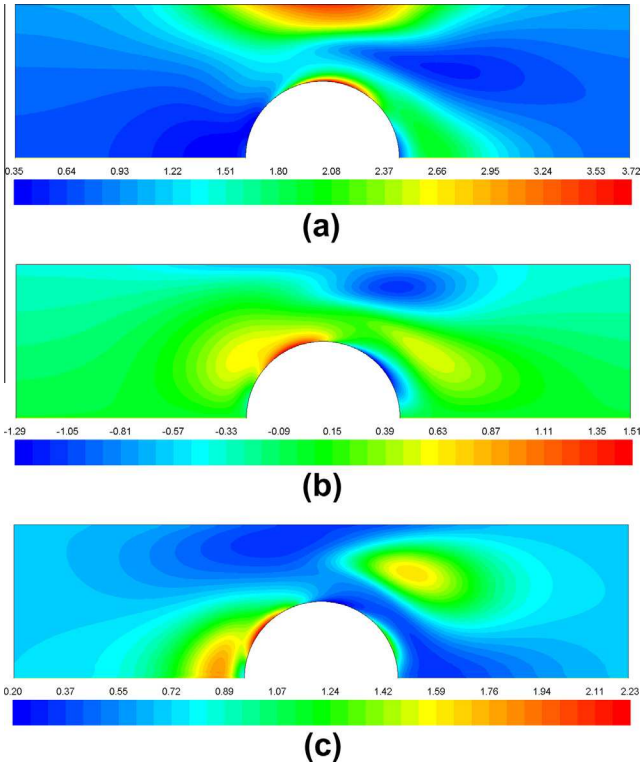


Fig. 7. Contours of dimensionless molecular elongations (configuration tensor) for $Wi_M = 0.6$: (a) $\langle\langle\tilde{\xi}_x\tilde{\xi}_x\rangle\rangle$; (b) $\langle\langle\tilde{\xi}_x\tilde{\xi}_y\rangle\rangle$; and (c) $\langle\langle\tilde{\xi}_y\tilde{\xi}_y\rangle\rangle$.

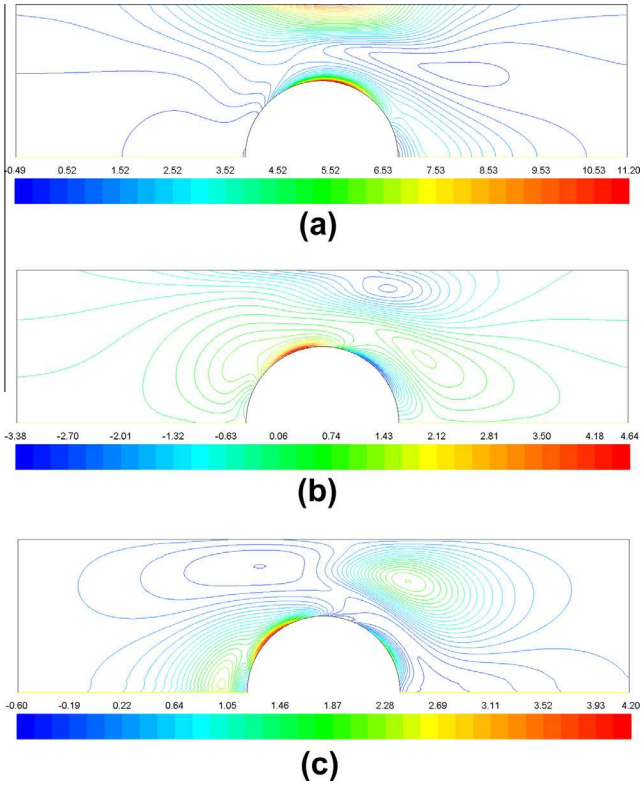


Fig. 8. Contours of dimensionless viscoelastic stresses for $Wi_M = 0.6$: (a) $\tilde{\boldsymbol{\sigma}}_{p_{xx}}$; (b) $\tilde{\boldsymbol{\sigma}}_{p_{xy}}$; and (c) $\tilde{\boldsymbol{\sigma}}_{p_{yy}}$.

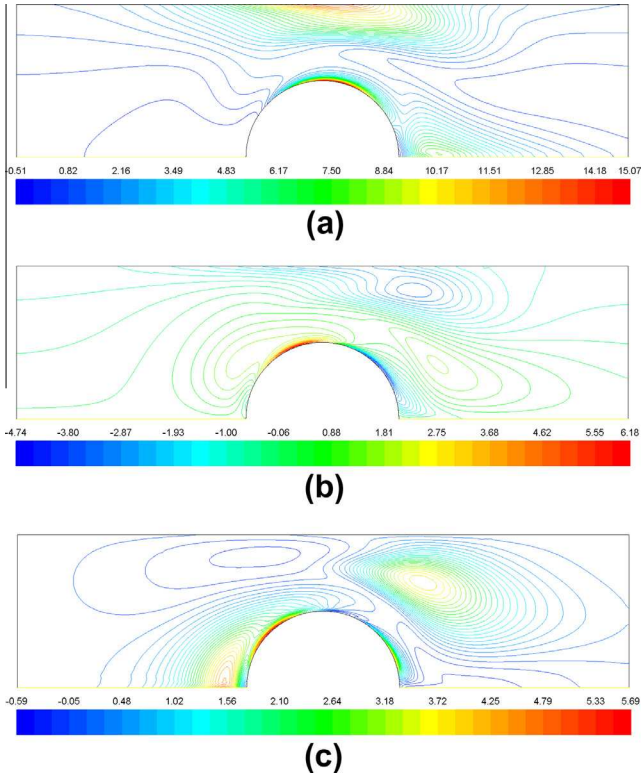


Fig. 9. Contours of dimensionless viscoelastic stresses for $Wi_M = 0.9$: (a) $\tilde{\boldsymbol{\sigma}}_{p_{xx}}$; (b) $\tilde{\boldsymbol{\sigma}}_{p_{xy}}$; and (c) $\tilde{\boldsymbol{\sigma}}_{p_{yy}}$.

2. *Maximize computational speed:* coarse lattice resolution and maximum stable relaxation time. The lattice parameters are chosen according to the maximum value of Wi_m in the domain, that is the coarsest allowed lattice and the maximum stable relaxation time. This approach does not guarantee constant nor converged numerical error. The maximum error depends on the choice of the lattice size (Fig. 4(a)).
3. *Strain-adaptive:* lattice resolution based on local Wi_m. The lattice parameters are dynamically adapted according to the local Wi_m (Fig. 10). Therefore, coarser lattices are used in $low - Wi_m$ regions and finer lattices in $high - Wi_m$ ones. This approach represents a trade-off between the two above discussed ones and allows to optimize the computational speed, providing control on the error. The number of different lattices to use can be chosen on the basis of an expanded analysis such as that in Fig. 4(a), according to the desirable degree of speed-up/error control.

An overview of the three approaches is reported in Table 3. The parameters for the comparison of the achievable speed-up are: $N_{min} = 41$, $N_{max} = 121$, $\tau_{min} = 0.55$ and τ_{max} the maximum local stable value for the locally-adaptive approach (Fig. 4(b)) and the maximum stable value for $Wi_m = 9$ for optimizing the computational speed (0.6 for D2Q9 and 0.55 for D2Q5). In order to compare the advantage of the locally-adaptive approach here we use two lattice sizes, namely $N = 81$ for $Wi_m = 1 \div 5$ and $N = 121$ for $Wi_m = 6 \div 9$. We remark that this choice is made to illustrate the methodology but the number of lattice resolutions is arbitrary.

5. Conclusions

A novel micro–macro model for dilute polymeric solutions has been presented. The proposed approach relies on a coupled numerical solution for the macro- and microscopic scales: a finite-volume

Table 2
Comparison of the calculated drag factor with the results obtained by Chauvière and Lozinski [9] for different Weissenberg number.

Wi_M	C_D^{μ}	C_D^{μ} [9]	C_D^{p}	C_D^{p} [9]	$C_D^{\sigma_p}$	$C_D^{\sigma_p}$ [9]	C_D	C_D [9]
0.6	6.2103	6.2248	1.9342	1.9980	0.6769	0.6696	8.8216	8.8925
0.9	5.8497	6.0175	1.9061	1.9953	0.5915	0.5393	8.3474	8.5521

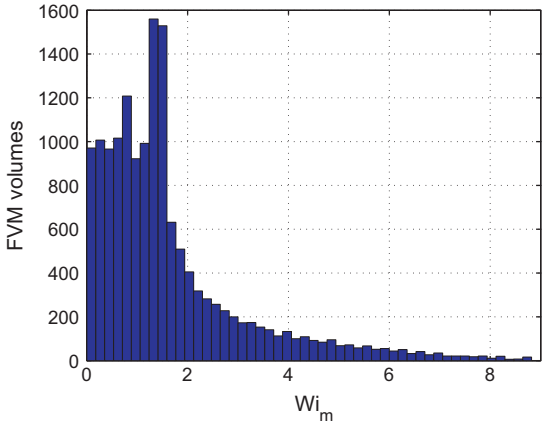


Fig. 10. Discrete distribution of Wi_m in the physical domain (mesh M1, number of bins 50, $Wi_M = 0.6$).

Table 3
Summary table of the three sub-grid solution approaches: ME (minimize error), SA (strain-adaptive) and MS (maximize speed). The comparison of the computational speed-up per FVM iteration refers to different approaches on the same stencil (results for mesh M1).

Stencil	Approach	N	τ	Error	Speed-up
D2Q9	(ME) *minimize error*	N_{max}	τ_{min}	variable	13.6 (ME/SA)
	(SA) *strain-adaptive*	$f(Wi_m)$	τ_{max}	controlled	17.9 (SA/MS)
	(MS) *maximize speed*	N_{min}	τ_{max}	variable	
D2Q5	(ME) *minimize error*	N_{max}	τ_{min}	variable	2.9 (ME/SA)
	(SA) *strain-adaptive*	$f(Wi_m)$	τ_{max}	controlled	23.7 (SA/MS)
	(MS) *maximize speed*	N_{min}	τ_{max}	variable	

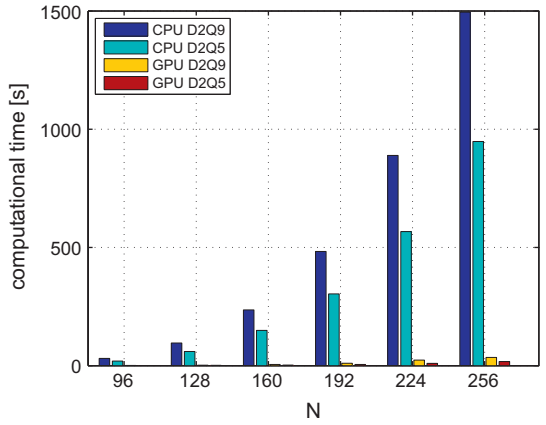


Fig. C.11. Comparison of the computational time for compiled C and the CUDA implementation using shared memory (D2Q9 and D2Q5 lattices with $\tau = 0.55$).

method for the fluid-flow equations and a lattice-Boltzmann method for the kinetic viscoelastic model. This micro–macro approach allows to properly simulate non-homogeneous viscoelastic flows. The convection of the configuration distribution function in physical space is taken into account by means of an operator-splitting algorithm. This leads to a convective-transport equation for viscoelastic stresses in the physical space, which is equivalent to the transport of the distribution function. The algorithm is optimized for its use in steady-state cases. The validity of the introduced model has been proven against the benchmark problem of two-dimensional flow past a confined cylinder. We have observed a decrease of accuracy as reaching $Wi_M = 1$. Regarding this issue, there are three sources of error that deserve further investigation: (i) the failure of the Chapman-Enskog expansion for $Wi_M > 1$ as reported by Singh et al. [16]; (ii) the time step selected for the operator-splitting algorithm, which is not unique and affects the accuracy and (iii) the Cartesian discretization of the configuration distribution function used when the Fokker–Planck equation is solved with the lattice Boltzmann method. These three sources of error could be avoided by a proper redefinition of the approach and this is left for future work. From a computational point of view, we have introduced and proven the validity of the coupling strategy when the micro-solver is implemented on a graphic card. This allows up to a 60x acceleration of the computational time. We have used a low-performance single graphic card, but the solution can also be distributed on multiple units, further reducing the computational time. We remark that in this work we proposed the coupling with a finite volume method solver, but the accelerated sub-grid solution can be easily called from other solvers (i.e. FEM–LBM or LBM–LBM solutions). Finally, the results obtained suggest that a direct numerical method together with proper hardware implementation, may deserve attention in the framework of numerical methods for complex fluids.

Table 4
Comparison of the computational time for C and CUDA (shared memory implementation).

N	96	128	160	192	224	256
D2Q9 CPU	30.4	95.5	235.8	483.0	889.0	1495.0
D2Q9 GPU	0.8	2.0	4.7	10.1	23.3	34.7
speed-up	38	48	50	48	38	43
D2Q5 CPU	19.0	60.0	149.0	303.5	567	948
D2Q5 GPU	0.4	1.3	2.5	5.0	9.6	7.1
speed-up	48	46	60	60	60	55

Acknowledgments

Referees are gratefully acknowledged for their help in improving the quality of the manuscript. LB and SI would also like to sincerely acknowledge Mr. Jorge Vea-Murguia for his support on the GPU facilities.

Appendix A. Asymptotic analysis

In this appendix we briefly report the procedure to recover the FENE equivalent Fokker–Planck Eq. (10) from the lattice-BGK Eq. (17) via asymptotic expansion (Chapman–Enskog procedure) [13]. Tilde notation is omitted for readability.

Let us consider the 2-nd order Taylor expansion of the post-collision term (first term on the left-hand side) in Eq. (17):

$$\Omega_i(f) \approx (\partial_t + \nabla\cdot\mathbf{c}_i)f_i + \frac{1}{2}(\partial_t^2 + 2\partial_t\nabla\cdot\mathbf{c}_i + \nabla\nabla:\mathbf{c}_i\mathbf{c}_i)f_i; \tag{A.1}$$

and the following expansions of the time derivative ∂_t and distribution function f_i in terms of a small formal number ϵ (spatial derivative is not expanded):

$$\partial_t = \epsilon\partial_{t1} + \epsilon^2\partial_{t2} + \mathcal{O}(\epsilon^3); \tag{A.2}$$

$$f_i = f_i^{eq} + \epsilon f_i^{(1)} + \epsilon^2 f_i^{(2)} + \mathcal{O}(\epsilon^3). \tag{A.3}$$

Applying (A.2) and (A.3) in Eq. (A.1) yields the scale-separated form (A.4) and (A.5). Combining to get rid of higher order derivatives yields (A.6).

$$(\partial_{t1} + \nabla\cdot\mathbf{c}_i)f_i^{eq} = -\frac{1}{\tau\delta t}f_i^{(1)}; \tag{A.4}$$

$$\partial_{t2}f_i^{eq} + (\partial_{t1} + \nabla\cdot\mathbf{c}_i)f_i^{(1)} + \frac{\delta t}{2}(\partial_{t1} + \nabla\cdot\mathbf{c}_i)^2 f_i^{eq} = -\frac{1}{\tau\delta t}f_i^{(2)}; \tag{A.5}$$

$$\partial_{t2}f_i^{eq} + \left(1 - \frac{1}{2\tau}\right)(\partial_{t1} + \nabla\cdot\mathbf{c}_i)f_i^{(1)} = -\frac{1}{\tau\delta t}f_i^{(2)}. \tag{A.6}$$

Using now the 0th order moment (19) and the condition (A.7) on the non-equilibrium distribution functions, yields (A.8) and (A.9):

$$\sum_i f_i^{noneq} = \sum_i f_i^{(1,2)} = 0; \tag{A.7}$$

$$\partial_{t1}\psi + \nabla\cdot(\psi\mathbf{u}) = 0; \tag{A.8}$$

$$\partial_{t2}\psi + \left(1 - \frac{1}{2\tau}\right)\nabla\cdot\sum_i \mathbf{c}_i f_i^{(1)} = 0; \tag{A.9}$$

Recovering $f_i^{(1)}$ from Eq. (A.4), the sum in (A.9) becomes (A.10), rearranging (A.11):

$$\begin{aligned}\sum_i \mathbf{c}_i f_i^{(1)} &= -\tau\delta t\sum_i \mathbf{c}_i(\partial_{t1} + \nabla\cdot\mathbf{c}_i)f_i^{eq} \\ &= -\tau\delta t\left(\partial_{t1}(\psi\mathbf{u}) + \nabla\cdot\left[\psi\left(\mathbf{u}\mathbf{u} + c_s^2\mathbf{I}\right)\right]\right);\end{aligned} \tag{A.10}$$

$$\sum_i \mathbf{c}_i f_i^{(1)} = -\tau\delta t(\mathbf{u}(\partial_{t1}\psi + \nabla\cdot(\psi\mathbf{u})) + c_s^2\nabla\psi). \tag{A.11}$$

Finally using (A.11) into (A.9) and reassembling scales, yields the final macroscopic equation:

$$\frac{\partial\psi}{\partial t} = -\nabla_\xi\cdot(\mathbf{u}\psi) + \delta t\left(\tau - \frac{1}{2}\right)c_s^2\nabla_\xi^2\psi; \tag{A.12}$$

that from the comparison with Eq. (7), gives the following expression for the lattice relaxation time:

$$\tau = \frac{\alpha}{\delta t c_s^2} + \frac{1}{2}. \tag{A.13}$$

Appendix B. Lattice constants

The discrete velocities $\mathbf{c}_i$ and weights ω_i for the D2Q9 lattice are:

$$\mathbf{c}_i = \begin{cases} (0,0) & i = 0 \\ (\pm c, 0), (0, \pm c) & i = 1,2,3,4 \\ (\pm c, \pm c) & i = 5,6,7,8 \end{cases}$$

$$\omega_i = \begin{cases} 4/9 & i = 0 \\ 1/9 & i = 1,2,3,4 \\ 1/36 & i = 5,6,7,8 \end{cases}$$

and for the D2Q5 lattice:

$$\mathbf{c}_i = \begin{cases} (0,0) & i = 0 \\ (\pm c, 0), (0, \pm c) & i = 1,2,3,4 \end{cases}$$

$$\omega_i = \begin{cases} 1/3 & i = 0 \\ 1/6 & i = 1,2,3,4 \end{cases}$$

Appendix C. GPU acceleration and coupling

In this section we present and discuss the methodology that progressively led us to the fastest implementation for Graphic Processing Unit (GPU). Going into the details of coding goes beyond the purpose of the present work, therefore we only provide a methodological description and the main results for each strategy.

The Compute Unified Device Architecture (CUDA) is a parallel computing platform and coding environment developed by NVIDIA© [25], which enables users to exploit the power of graphic processing units for scientific applications (see the CUDA Programming Guide [26]). Several open-source codes are already available to help users with the implementation.

Sailfish, for example, is an open-source code for computational fluid dynamics based on lattice Boltzmann method and optimized for NVIDIA© graphic cards [27]. The user sets the simulation up (in Python) and the CUDA code is created automatically and runs on the graphic card. Using *Sailfish* the speed-up of the sub-grid solution reaches nearly 60× with respect to a compiled-C code. However, calling the solution from Fluent® is not computationally efficient due to the *Sailfish* start-up time which is comparable with the simulation time, therefore an *ad-hoc* CUDA implementation has to be developed.

We developed a CUDA code based on *texture memories* [26]. However this implementation did not show to be optimal for our purposes, as long as the maximum speed-up achieved is around 25x with respect to the CPU.

In order to overcome the limit of the previous code, we developed a code based on *shared memory* [26]. Thanks to the utilization of this extremely fast memory, with this implementation the speed-up reaches nearly 60× with respect to the CPU, as shown in Fig. C.11 and Table 4. This solution is more flexible than *Sailfish* for our purpose and it can be efficiently called from Fluent®.

The developed code is finally compiled with the CUDA compiler (nvcc) and dynamically called from a Fluent® user defined function. The sub-grid simulation is driven by passage and retrieval of the required variables between the two compiled codes through a stream process.

References

[1] R.G. Owens, T.N. Phillips, Computational Rheology, Imperial College Press, 2002.
[2] R.B. Bird, R.C. Armstrong, O. Hassager, Dynamics of Polymeric Liquids (Fluid Mechanics), vol. 1, Wiley-Interscience, 1987.
[3] R.B. Bird, C.F. Curtiss, R.C. Armstrong, O. Hassager, Dynamics of Polymeric Liquids (Kinetic Theory), vol. 2, Wiley-Interscience, 1987.
[4] R. Keunings, Micro–macro methods for the multiscale simulation of viscoelastic flow using molecular models of kinetic theory, Rheology Reviews (British Society of Rheology) (2004) 67–98.
[5] M. Laso, H.C. Öttinger, Calculation of viscoelastic flow using molecular models: the CONNFFESSIT approach, Journal of Non-Newtonian Fluid Mechanics 47 (0) (1993) 1–20.
[6] B. Jourdain, C. Le Bris, T. Lelièvre, On a variance reduction technique for micro–macro simulations of polymeric fluids, Journal of Non-Newtonian Fluid Mechanics 122 (1–3) (2004) 91–106.
[7] M.A. Hulsen, A.P.G. van Heel, B.H.A.A. van den Brule, Simulation of viscoelastic flows using Brownian configuration fields, Journal of Non-Newtonian Fluid Mechanics 70 (1–2) (1997) 79–101.
[8] A. Lozinski, C. Chauvière, A fast solver for Fokker–Planck equation applied to viscoelastic flows calculations: 2D FENE model, Journal of Computational Physics 189 (2) (2003) 607–625.

[9] C. Chauvière, A. Lozinski, Simulation of complex viscoelastic flows using the Fokker–Planck equation: 3D FENE model, Journal of Non-Newtonian Fluid Mechanics 122 (1–3) (2004) 201–214.
[10] D. Moroni, B. Rotenberg, J.P. Hansen, S. Succi, S. Melchionna, Solving the Fokker–Planck kinetic equation on a lattice, Physical Review E 73 (2006) 066707.
[11] A. Ammar, B. Mokdad, F. Chinesta, R. Keunings, A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids, Journal of Non-Newtonian Fluid Mechanics 139 (3) (2006) 153–176.
[12] D.J. Knezevic, A.T. Patera, A certified reduced basis method for the Fokker–Planck equation of dilute polymeric fluids: FENE dumbbells in extensional flow, SIAM Journal on Scientific Computing 32 (2) (2010) 793–817.
[13] A. Ammar, Lattice Boltzmann method for polymer kinetic theory, Journal of Non-Newtonian Fluid Mechanics 165 (19–20) (2010) 1082–1092.
[14] F. Wu, W. Shi, F. Liu, A lattice Boltzmann model for the Fokker–Planck equation, Communications in Nonlinear Science and Numerical Simulation 17 (7) (2012) 2776–2790.
[15] A. Majumder, V. Kariwala, S. Ansumali, A. Rajendran, Lattice Boltzmann method for multi-dimensional population balance models in crystallization, Chemical Engineering Science 70 (0) (2012) 121–134.
[16] S. Singh, G. Subramanian, S. Ansumali, A lattice Boltzmann method for dilute polymer solutions, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 369 (1944) (2011) 2301–2310.
[17] T. Li, Mathematical analysis of multi-scale models of complex fluids, Communications in Mathematical Sciences 5 (1) (2007) 1–51.
[18] A. Onuki, Phase Transition Dynamics, Cambridge University Press, 2002.
[19] H.R. Warner, Kinetic theory and rheology of dilute suspensions of finitely extendible dumbbells, Industrial & Engineering Chemistry Fundamentals 11 (3) (1972) 379–387.
[20] J. Barrett, E. Süli, Existence of global weak solutions to some regularized kinetic models for dilute polymers, Multiscale Modeling & Simulation 6 (2) (2007) 506–546.
[21] R.O. Vargas, O. Manero, T.N. Phillips, Viscoelastic flow past confined objects using a micro–macro approach, Rheologica Acta 48 (2009) 373–395.
[22] H. Versteeg, W. Malalasekra, An Introduction to Computational Fluid Dynamics: The Finite Volume Method Approach, Prentice Hall, 1996.
[23] S. Succi, The Lattice Boltzmann Equation for Fluid Dynamics and Beyond, Oxford University Press, 2001.
[24] S. Chen, G.D. Doolen, Lattice Boltzmann method for fluid flows, Annual Review of Fluid Mechanics 30 (1) (1998) 329–364.
[25] Nvidia Developer Zone, 2012a. <https://www.developer.nvidia.com/category/zone/cuda-zone>.
[26] Cuda Toolkit Downloads, 2012b. <https://www.developer.nvidia.com/cuda-downloads>.
[27] Sailfish Reference Manual And Repository, 2012. <http://www.sailfish.us.edu.pl/>.