ELSEVIER

J. Non-Newtonian Fluid Mech. 141 (2007) 180–192

Journal of Non-Newtonian Fluid Mechanics

www.elsevier.com/locate/jnnfm

# An efficient algorithm for multiscale flow simulation of dilute polymeric solutions using bead-spring chains

Anantha P. Koppol [a], Radhakrishna Sureshkumar [a], Bamin Khomami [a,b,*]

[a] The Materials Research Laboratory, Department of Chemical Engineering, Washington University, St. Louis, MO 63130, USA
[b] Department of Chemical Engineering, University of Tennessee, Knoxville, TN 37996, USA

Received 2 May 2006; received in revised form 20 September 2006; accepted 31 October 2006

## Abstract

In this study, a computationally efficient algorithm for multiscale flow simulation of dilute polymer solutions using a bead-spring chain description of polymer molecules is presented. The algorithm combines a computationally efficient extension of the earlier BCF-based semi-implicit method (i.e., approximately four-fold speed up) for multiscale flow simulations using a bead-spring dumbbell description [M. Somasi, B. Khomami, Linear stability and dynamics of viscoelastic flows using time-dependent stochastic simulation techniques, J. Non-Newtonian Fluid Mech. 93 (2000) 339–362] with a highly CPU efficient predictor–corrector scheme for BD simulation of bead-spring chains [M. Somasi, B. Khomami, N. Woo, J. Hur, E. Shaqfeh, Brownian dynamics simulations of bead-rod and bead-spring chains: numerical algorithms and coarse-graining issues, J. Non-Newtonian Fluid Mech. 108 (2002) 227–255]. The fidelity and computational efficiency of the parallel implementation of the algorithm are demonstrated via three benchmark flow problems, namely, plane Couette flow, Poiseuille flow and 4:1:4 axisymmetric contraction–expansion flow. The algorithm shows linear speed up with the number of processors and more importantly with the number of chain segments. In addition, the proposed algorithm is approximately 50 times faster in comparison to the only existing fully implicit method [J. Ramirez, M. Laso, Size reduction methods for the implicit time-dependent simulation of micro-macro viscoelastic flow problems, J. Non-Newtonian Fluid Mech. 127 (2005) 41–49].

*Keywords:* Semi-implicit; Brownian configuration fields; Multiscale; FENE bead-spring chain

## 1. Introduction

Currently two main approaches, namely, calculation of non-Newtonian flows: finite elements and stochastic simulation techniques (CONNFFESSIT) [1] and Brownian configuration fields (BCF) [2] are available for multiscale flow simulation of dilute polymeric solutions. Both methodologies couple continuum-level conservation equations for mass and momentum that are solved in an Eulerian frame of reference with a stochastic differential equation for chain dynamics solved via Brownian dynamics (BD) to obtain the polymeric stress at the mesoscopic level. The CONNFFESSIT technique relies on the tracking of individual polymer molecules within the flow domain, hence, in complex kinematics flows it can lead to numerically induced spatial inhomogenity in polymer concentration resulting in substantial errors in the computation of polymeric stresses. However, these issues have been to a great extent addressed in a recent variation of the CONNFFESSIT technique, namely the Lagrangian particle method (LPM) [3]. Specifically, in this approach, a number of fluid particles that contain an ensemble of polymer molecules as opposed to individual chains are tracked; hence a significant reduction in the computational requirements is realized. In addition, the undesirable spatial depletion of polymer molecules has been addressed by judicious insertion of particles in the depleted regions via a backward tracking algorithm [4].

The aforementioned drawbacks of the CONNFFESSIT technique have also been addressed by the BCF technique. Specifically, the BCF approach relies on an ensemble of continuous fields of spatially correlated macromolecules as opposed to individual polymer molecules to describe the polymer dynamics under flow. In turn, the evolution of each field is computed using standard Eulerian techniques leading to tremendous spatial variance reduction, i.e., the polymeric stresses are spatially smooth.

Although in recent years significant progress in variance reduction methods [5,6] as well as algorithmic advances, i.e., development of semi-implicit [7,8] and fully implicit [9] tech-

* Corresponding author. Tel.: +1 865 974 2085; fax: +1 865 974 7076.
*E-mail address:* bkhomami@utk.edu (B. Khomami).

0377-0257/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.jnnfm.2006.10.003

niques for multiscale flow simulation of dilute polymer solutions have been made, the multiscale approach still remains computationally intensive in comparison to continuum-level flow simulations of viscoelastic fluids using single or multimode constitutive equations. Hence, to date all the large-scale complex kinematics multiscale flow simulations of dilute polymeric solutions have been restricted to very simple mesoscopic models for the polymer molecule, namely, single segment elastic dumbbell models [10,11].

In recent years, fluorescence microscopy of model macromolecules, namely DNA, in a variety of flow fields has provided an avenue for examining the fidelity of various mesoscopic level models describing polymer dynamics under flow. Specifically, it has been shown that multi-segment bead-rod and bead-spring descriptions can describe both single molecule dynamics such as molecular individualism, and unraveling dynamics, as well as the solution rheological properties such as viscosity and mean molecular extension (i.e., an indirect measure of normal stresses) with good accuracy [12,13]. In addition, it has been shown that single segment elastic dumbbell models as well as closed form constitutive equations obtained by invoking various closures such as the FENE-P, FENE-LS at this level of description can at best qualitatively describe the polymer dynamics and rheological properties of dilute polymer solutions [12,13].

These findings clearly underscore the fact that a multi-segment description of the macromolecule or reduced order coarse-grained models that contain information regarding the internal degrees of freedom of the macromolecule are required for quantitative modeling of dilute polymer solutions under flow. Motivated by this fact, Ramirez and Laso [14] have extended their recently developed fully implicit method based on the BCF approach [9] to multi-segment bead-spring models for describing the polymer dynamics. Although, their method due to its fully implicit nature allows for accurate and stable solutions with relatively large time-steps, it is highly computationally intensive and difficult to implement due to the use of a highly specialized size reduction technique for implicitly solving a large system of coupled non-linear equations, constituting the macroscopic conservation equations and the governing BCF equations.

In this paper, we present a computationally efficient semi-implicit method for large-scale multiscale flow simulations of dilute polymeric solutions using bead-spring chains. Essentially this method combines a computationally efficient semi-implicit predictor–corrector scheme for BD simulations of bead-spring chains [15] with a numerically efficient extension (i.e., a four-fold speed up) of a BCF-based semi-implicit multiscale flow simulation algorithm [7]. In what follows, the fidelity and computational efficiency of a parallel implementation of this new algorithm are demonstrated via three benchmark flow problems. In turn, the computational efficiency of this method is compared with the fully implicit method of Ramirez and Laso [14].

## 2. Governing equations

The mass and momentum conservation equations governing the creeping flow of an incompressible fluid in the absence of body forces are as follows:

$$-\vec{\nabla}P + \vec{\nabla}\cdot\underline{\underline{\tau}} = 0, \tag{1}$$

$$\vec{\nabla}\cdot\vec{u} = 0, \tag{2}$$

where $P$, $\underline{\underline{\tau}}$ and $\vec{u}$ are the dimensionless isotropic fluid pressure, deviatoric stress tensor and velocity vector, respectively. $P$ and $\underline{\underline{\tau}}$ are non-dimensionlized with $(\eta_0 u_c/L)$, and $\vec{u}$ is non-dimensionlized with $u_c$, where $L$ and $u_c$ are the chosen characteristic length and velocity scales, and $\eta_0$ is the zero shear viscosity of the fluid.

The extra stress ($\underline{\underline{\tau}}$) in Eqs. (1) and (2) is split into the stress contribution from the polymer and the solvent as follows:

$$\underline{\underline{\tau}} = \underline{\underline{\tau}}_p + \underline{\underline{\tau}}_s. \tag{3}$$

The solvent stress contribution $\underline{\underline{\tau}}_s$ is assumed to be Newtonian, therefore:

$$\underline{\underline{\tau}}_s = 2\beta\underline{\underline{\dot{\gamma}}}, \tag{4}$$

where $\beta$ is the ratio of the solvent to the zero shear viscosity, i.e., $\beta = \eta_s/\eta_0$. And $\underline{\underline{\dot{\gamma}}}$ is the rate of deformation tensor of the fluid, defined as

$$\underline{\underline{\dot{\gamma}}} = \frac{1}{2}(\vec{\nabla}\vec{u} + \vec{\nabla}\vec{u}^{\mathrm{T}}). \tag{5}$$

The macromolecules are modeled as freely draining, non-interacting bead-spring chains. The polymeric stress contribution, $\underline{\underline{\tau}}_p$is computed using the Kramer's expression [16] as follows:

$$\underline{\underline{\tau}}_p = \frac{1-\beta}{We^*}\sum_{i=1}^{N}(\langle\vec{F}_i\vec{Q}_i\rangle - \langle\vec{F}_i\vec{Q}_i\rangle_{\mathrm{eqbm}}), \tag{6}$$

where $N$ is the number of segments in the bead-spring model; $\langle\cdot\rangle$ the ensemble average; $\langle\vec{F}_i\vec{Q}_i\rangle_{\mathrm{eqbm}} = \underline{\underline{I}}$, the unit tensor; $\vec{Q}_i$ the dimensionless connector vector between the two beads of the $i$th segment scaled with the equilibrium length of a Hookean dumbbell $\sqrt{kT/H}$, with a spring constant $H$; $\vec{F}_i$ is the dimensionless entropic force vector for the $i$th segment. In this study, we have chosen to use the FENE force law:

$$\vec{F}_i = \frac{\vec{Q}_i}{1-(Q_i^2/b)}, \tag{7}$$

where $b$ is the square of the dimensionless segmental maximum extensibility, i.e., $b = HQ_0^2/kT$; $Q_0$ the maximum extensibility of each segment; $We^* = We((N+1)^2-1)/(3+15/b)$ [17], $We = \lambda_H u_c/L$ is the Weissenberg number and $\lambda_H$ is the Hookean relaxation time of the bead-spring chain.

In the BCF approach, the evolution equation for the $i$th connector vector $\vec{Q}_i$ is given by

$$\mathrm{d}\vec{Q}_i(x,t) = \left[-\vec{u}(x,t)\cdot\nabla\vec{Q}_i(x,t) + \underline{\underline{\kappa}}(x,t)\cdot\vec{Q}_i(x,t) + \frac{\vec{F}_i^{\mathrm{E}}}{4We}\right]\mathrm{d}t + \sqrt{\frac{1}{2We}}[\mathrm{d}\vec{W}_{i+1}(t) - \mathrm{d}\vec{W}_i(t)], \tag{8}$$

where

$$\vec{F}_i^{\mathrm{E}} = \begin{cases} -2\vec{F}_i + \vec{F}_{i+1}; & i = 1, \\ \vec{F}_{i-1} - 2\vec{F}_i + \vec{F}_{i+1}; & 1 < i < N, \\ \vec{F}_{i-1} - 2\vec{F}_i; & i = N. \end{cases} \tag{9}$$

In Eq. (8), the term $\mathrm{d}\vec{W}_i(t)$ is the Wiener process that accounts for the Brownian force experienced by the $i$th bead. It is mathematically represented as a Gaussian random vector with zero mean and a variance of d$t$. The term $\vec{u}(x,t)\cdot\vec{\nabla}\vec{Q}(x,t)$ accounts for the convection of a configuration field under the influence of the flow field. Note that the Wiener process is independent of $x$ indicating that the configuration field is subjected to a spatially uniform Brownian force at each time step. As a result, the configuration field vector, $\vec{Q}_i(x,t)$ evolves as a smooth and continuous function for spatially correlated velocity fields, thus making the BCF technique ideally suitable for implementation in a finite element context.

## 3. Computational technique

The DEVSS-G formulation [18–20] is used for discretization of the governing equations. In this formulation, a stabilization term is added to the momentum equation as shown below:

$$\vec{\nabla}^2\vec{u} - \vec{\nabla}P + \vec{\nabla}\cdot\left[\underline{\underline{\tau}}_{\mathrm{p}} - (1-\beta)(\underline{\underline{G}} + \underline{\underline{G}}^{\mathrm{T}})\right] = 0, \tag{10}$$

where $\underline{\underline{G}}$ is the discrete interpolant of the velocity gradient $(\vec{\nabla}\vec{u})$, and the term in the square brackets is the discrete form of the elastic stress tensor. The added stabilization term vanishes in the strong formulation but not in the weak form, hence it maintains the elliptic nature of the momentum equation. Hence, as $\underline{\underline{G}}$ "approaches" the exact solution the original momentum equation is restored.

The modified momentum equation (10), the continuity equation (2), and the velocity gradient interpolant, $\underline{\underline{G}}$ are discretized using the Galerkin technique. The weak form for these equations are given below:

$$[(\vec{\nabla}\vec{v}):(\vec{\nabla}\vec{u} + \vec{\nabla}\vec{u}^{\mathrm{T}} - P\underline{\underline{I}} + \Sigma)] = [\vec{v}:\vec{\sigma}]_{\Gamma}, \tag{11}$$

$$[q;\vec{\nabla}\cdot\vec{u}] = 0, \tag{12}$$

$$[g:(\underline{\underline{G}} - \vec{\nabla}\vec{u})] = 0, \tag{13}$$

where

$$\Sigma = \underline{\underline{\tau}}_{\mathrm{p}} - (1-\beta)(\underline{\underline{G}} + \underline{\underline{G}}^{\mathrm{T}}), \tag{14}$$

and $[a:b]$ (or $[a\,;b]$) and $[a:b]_\Gamma$ are the standard inner products of $(a\,,b)$ in the flow domain $\Omega$ and on the boundary $\Gamma$, respectively. $\sigma$ is the traction vector on the boundary.

The flow domain $\Omega$ is divided into quadrilateral finite elements in which the shape functions are defined. The hierarchic shape functions based on the Legendre polynomials as proposed by Szabo and Babuska [21] are used to approximate the variables. Specifically, the velocity components are approximated using second order polynomials ($p=2$), and the velocity gradient, stress, and pressure variables are approximated using bilinear shape functions ($p=1$). The choice of the shape functions to approximate the velocities, velocity gradients and pressure satisfy the Brezzi–Babuska condition [22]. The shape functions used to approximate the stresses are in accordance with that proposed in earlier studies [7,22].

The Galerkin projection is used to evaluate the polymeric stress via:

$$\left[\left(\underline{\underline{\tau}}_{\mathrm{p}} - \frac{1-\beta}{We^*}\sum_{i=1}^{N}(\langle\vec{F}_i\vec{Q}_i\rangle - \underline{\underline{I}})\right):\underline{\underline{a}}\right] = 0, \tag{15}$$

where $\underline{\underline{a}}$ is chosen from the bilinear continuous polynomial space ($p=1$).

The BCF equations (8) are discretized using the SUPG technique [22] in which the weighting functions are modified as $\underline{\underline{Z}} = \underline{\underline{\phi}} + (\vec{u}\cdot\vec{\nabla}\underline{\underline{\phi}})h/|\vec{u}|$ and $\underline{\underline{\phi}}$ represents the weighting function in the Galerkin formulation as shown below:

$$\left[\left(\mathrm{d}\vec{Q}_i - \left(-\vec{u}\cdot\vec{\nabla}\vec{Q}_i + \underline{\underline{\kappa}}\cdot\vec{Q}_i + \frac{1}{4We}\vec{F}_i^{\mathrm{E}}\right)\,\mathrm{d}t - \sqrt{\frac{1}{2We}}(\mathrm{d}\vec{W}_{i+1}(t) - \mathrm{d}\vec{W}_i(t))\right):\underline{\underline{Z}}\right] = 0, \tag{16}$$

where $\underline{\underline{Z}}$ is selected from the bilinear continuous polynomial space ($p=1$).

### 3.1. Algorithm development

The algorithm is developed employing a computationally efficient time integration method based on a semi-implicit predictor–corrector scheme, where the stress and velocity fields are alternatively updated in a decoupled fashion until a self-consistent solution is obtained at each time step. The resulting linear system of equations obtained at each time step are solved using a state-of-the-art frontal LU decomposition solver for finite element applications, which was also used in our earlier work [7] and was found to be computationally robust and efficient. The algorithmic details are presented below:

At time $t=0$, the configuration fields are initialized to their equilibrium values and the velocity field corresponding to the equilibrium polymeric stress is obtained. The evolution of the configuration fields and the velocity field is computed using the following steps.

The initial estimate for the velocity field variables of the $(n+1)$th time step used in both the BCF equation (8) as well as the SUPG test function ($\underline{\underline{Z}}$) is set to the previous time step ($n$), i.e., $\vec{u}^* = \vec{u}^n$ and $\underline{\underline{\kappa}}^* = \underline{\underline{\kappa}}^n$.

- *Step 1.* In the predictor step, an estimate for each configuration field $Q_i^*$, $i=1,\ldots,N$ is obtained from the solution at the previous time step ($n$) using the explicit Euler integration scheme as shown below:

$$\vec{Q}_i^* = \vec{Q}_i^n + \left[-\vec{u}^*\cdot\vec{\nabla}\vec{Q}_i^n + \underline{\underline{\kappa}}^*\cdot\vec{Q}_i^n + \frac{1}{4We}\vec{F}_i^{\mathrm{E},n}\right]\Delta t + \sqrt{\frac{1}{2We}}(\mathrm{d}\vec{W}_{i+1}^{n+1}(t) - \mathrm{d}\vec{W}_i^{n+1}(t)). \tag{17}$$

- *Step 2*. In the corrector step, the configuration fields for the current time step $Q_i^{n+1}, i=1,\ldots,N$ are determined employing a semi-implicit scheme shown below:

$$\vec{Q}_i^{n+1} + \left[\vec{u}^* \cdot \vec{\nabla}\vec{Q}_i^{n+1} - \underline{\underline{\kappa}}^* \cdot \vec{Q}_i^{n+1}\right]\Delta t = \vec{Q}_i^{n} + \frac{\Delta t}{4We}\tilde{\vec{F}}_i^{\mathrm{E},n+1} + \sqrt{\frac{1}{2We}}(\mathrm{d}\vec{W}_{i+1}^{n+1}(t) - \mathrm{d}\vec{W}_i^{n+1}(t)), \tag{18}$$

where

$$\tilde{\vec{F}}_i^{\mathrm{E},n+1} = \begin{cases} -2\tilde{\vec{F}}_i^{n+1} + \vec{F}_{i+1}^{n}; & i=1, \\ \vec{F}_{i-1}^{n+1} - 2\tilde{\vec{F}}_i^{n+1} + \vec{F}_{i+1}^{n}; & 1<i<N, \\ \vec{F}_{i-1}^{n+1} - 2\tilde{\vec{F}}_i^{n+1}; & i=N. \end{cases} \tag{19}$$

In the configuration field update (Eq. (18)), the $\vec{u}\cdot\vec{\nabla}\vec{Q}$ and $\underline{\underline{\kappa}}\cdot\vec{Q}$ terms, and the spring force of the $(i-1)$th segment are treated implicitly, the spring force of the $(i+1)$th segment is treated explicitly, while the spring force of the $i$th segment ($\tilde{\vec{F}}_i^{n+1}$) is obtained by solving the BCF equation (8) locally at the Gaussian quadrature points. Specifically, Eq. (8) is rearranged such that the spring force in the segment of interest is treated implicitly and the $\vec{u}\cdot\vec{\nabla}\vec{Q}$ and $\underline{\underline{\kappa}}\cdot\vec{Q}$ terms are treated explicitly as shown below:

$$\begin{aligned} \tilde{\vec{Q}}_i^{n+1} + \frac{\Delta t}{2We}\tilde{\vec{F}}_i^{n+1} &= \tilde{\vec{Q}}_i^{n} + \frac{1}{2}\left[-\tilde{\vec{u}}^*\cdot\vec{\nabla}(\tilde{\vec{Q}}_i^{*} + \tilde{\vec{Q}}_i^{n}) + \tilde{\underline{\underline{\kappa}}}^*\cdot(\tilde{\vec{Q}}_i^{*} + \tilde{\vec{Q}}_i^{n}) \right. \\ &\quad \left. + \frac{1}{2We}(\tilde{\vec{F}}_{i-1}^{n+1} + \tilde{\vec{F}}_{i+1}^{n})\right]\Delta t \\ &\quad + \sqrt{\frac{1}{2We}}(\mathrm{d}\vec{W}_{i+1}^{n+1}(t) - \mathrm{d}\vec{W}_i^{n+1}(t)), \end{aligned} \tag{20}$$

where the quantities with the superscript tilde are evaluated at the local Gaussian quadrature points.

Following the same approach as our earlier works [7,15,23] Eq. (20) is rearranged to obtain, a cubic equation for $|\tilde{\vec{Q}}_i^{n+1}|$:

$$|\tilde{\vec{Q}}_i^{n+1}|^3 - |\vec{R}||\tilde{\vec{Q}}_i^{n+1}|^2 - b\left(1+\frac{\Delta t}{2We}\right)|\tilde{\vec{Q}}_i^{n+1}| + b|\vec{R}| = 0, \tag{21}$$

where $|\vec{R}|$ is the magnitude of the right hand side vector of Eq. (20). The above cubic equation has a unique solution between 0 and $\sqrt{b}$, choosing this root will ensure $|\tilde{\vec{Q}}_i^{n+1}|$ is not greater than $\sqrt{b}$. $\tilde{\vec{F}}_i^{n+1}$ is then determined from Eq. (20), which is used in Eq. (18) to update the configuration field. Although $|\tilde{\vec{Q}}_i^{n+1}|$can never be larger than $\sqrt{b}$ at the Gaussian quadrature points, in regions of high stretch a minor violation of this condition could occur due to the projection of $\tilde{\vec{Q}}_i^{n+1}$ on to the bilinear basis functions. This issue can be resolved without loss of accuracy by locally substituting the violated configuration field with the one that predicts the highest permissible extension in the ensemble (with the assumption that the violated configuration field must be almost fully stretched).

Note that the semi-implicit treatment of the spring force (Eq. (19)) makes the left hand side of Eq. (18) a function of $\vec{u}^*$and $\underline{\underline{\kappa}}^*$, hence the mass matrix does not have to be recomputed in the corrector step. This modification results in significant computational saving (i.e., up to four-fold speed up) in comparison to the implicit treatment of the spring force [7], while retaining the fidelity of the computations. More importantly, this modification allows decoupling of the configuration field updates (Eqs. (17) and (18)) leading to independent solution for each segment at the Gaussian quadrature point solution (Eq. (20)). Hence, one would expect to obtain a nearly linear scale up of the computation time with the number of segments. Clearly in a fully implicit treatment a system of coupled non-linear equations has to be solved for which the computation time does not scale up linearly with the number of segments. Another advantage with the proposed approach is that the BCF equation (8) is solved in the weak form for any order of polynomials used to approximate the configuration fields and allows the use of a suitable upwinding scheme for dicretizing the BCF equation.

The corrected configuration fields $\vec{Q}_i^{n+1}$, obtained from Eq. (18) are used to update the polymeric stress as shown below:

$$\underline{\underline{\tau}}_{\mathrm{p}}^{n+1} = \frac{1-\beta}{We^*}\sum_{i=1}^{N}(\langle\vec{F}_i^{n+1}\vec{Q}_i^{n+1}\rangle - \underline{\underline{I}}). \tag{22}$$

The velocity field is then updated using $\underline{\underline{\tau}}_{\mathrm{p}}^{n+1}$ in Eqs. (11)–(14) to obtain $u^{n+1}$, $G^{n+1}$ and $P^{n+1}$.

The convergence of the configuration fields is estimated by computing the residual $\varepsilon$:

$$\varepsilon = \frac{\sum_{i=1}^{N}|1-|\vec{Q}_i^{n+1}|/|\vec{Q}_i^{*}||}{N}. \tag{23}$$

Note that the residual $\varepsilon$ is computed relative to the magnitude $\vec{Q}_i^*$, which is always greater than zero. If $\varepsilon$ is found to be greater than a specified tolerance (i.e., typically $10^{-3}$ to $10^{-6}$) $\vec{Q}_i^{n+1}$ is stored as $\vec{Q}_i^*$ and step 2 is repeated until the residual meets the tolerance, thus resulting in self-consistent stress and velocity fields at each time step.

In an attempt to further speed up our simulations, we have implemented a number of algorithmic modifications that lead to approximately 40% CPU reduction while maintaining the solution accuracy: (a) the residual $\varepsilon$ has been increased to $10^{-3}$ leading to reduction of the number of corrector steps; (b) the number of times the mass matrix for Eqs. (17) and (18) are updated is minimized by performing the updates only when the average relative change in $\vec{u}^*$and $\underline{\underline{\kappa}}^*$ exceeds 10%. Irrespective of the above rules, after every 100 time steps the mass matrices for Eqs. (17) and (18) are generated and the tolerance is reduced to $10^{-6}$. It should be noted that we store the transient data after every 100 time steps for post-processing.

The mesoscopic level simulations are inherently parallelizable, so we have implemented our algorithm on a parallel computing architecture, consisting of 64 SGI origin 2000 processors with 64 GB shared memory, 400 MHz clock speed, 8 MB L2 Cache, 800 MFLOP/s per processor, and fast Ethernet of bandwidth 1 GB/s for the inter-process communication. Message passing interface (MPI), which is widely accepted and available on all major platforms, has been used to efficiently parallelize the algorithm via optimizing the inter-process communication and memory requirements. The MPI is based on a master–slave framework, where the master processor initiates the parallelization and synchronizes the computations on all the processors. Further, the master processor, in our implementation, at each time step solves for the velocity field using the updated polymeric stress, communicates the updated velocity field to all the processors, and to avoid repetition generates and distributes the Gaussian random numbers to all the processors. The task of solving the BCF equations for the large number of configuration fields and updating of the polymeric stress is distributed among all the processors.

## 4. Parameter selection and fluid rheology

The rheological behavior of the fluid is predicted using equivalent FENE bead-spring models with the number of segments ($N$) varying from 1 to 6. The equivalence between the FENE bead-spring models has been established by relating the maximum extensibility (length scale) and time constant (time scale) of the FENE bead-spring chains with those of the FENE dumbbell model. For the FENE bead-spring chain with $N$ segments, the maximum extensibility of each spring, $b$ is fixed to $b_{\max}/N$, where $b_{\max}$ is the maximum extensibility of the FENE dumbbell model. The time constants of the FENE bead-spring models can be related via different approaches, such as, equating the longest relaxation time [15], matching the zero shear material functions $\eta_{\text{p}0}$ or $\psi_{10}$ [24], or the zero shear characteristic relaxation time scale ($\psi_{10}/2\eta_{\text{p}0}$) [15]. We have chosen to equate the zero shear characteristic relaxation time scales since this approach has been shown to provide a reasonable match of the steady and transient behavior of the FENE bead-spring chains in both shear and elongational flows [15]. Based on this approach, the time constants, that is, the Hookean relaxation times of the FENE dumbbell ($\lambda_{\text{H}}^*$) and the FENE bead-spring chains ($\lambda_{\text{H}}$) [17] are related as follows:

$$\lambda_{\text{H}} = \frac{\lambda_{\text{H}}^*}{d}, \quad \text{where } d = \left(\frac{b_{\max}+7}{15b_{\max}}\right)\left(\frac{b}{b+5}\right) \times \left((2(N+1)^2+7) - \frac{12((N+1)^2+1)}{(N+1)(b+7)}\right). \tag{24}$$

Furthermore, in order to facilitate a better comparison of the performance of the algorithm using different FENE bead-spring models we have also equated $\eta_{\text{p}0}$ of all the models by following an approach suggested by van Heel et al. [25], where the number density ($nkT$) is considered as an additional model parameter. This procedure allows us to match $\psi_{10}$ as well as $\eta_{\text{p}0}$ since the ratio of the two parameters is matched via Eq. (24).

Fig. 1(a) and (b) show the steady shear viscosity ($\eta_{\text{p}}$) and first normal stress coefficient ($\psi_1$) of the FENE bead-spring models with $b_{\max}=900$ and 225, respectively, for *We* number varying from 0 to 40. Fig. 1(c) shows the transient extensional viscosity ($\eta^{\text{E}}/\eta_0$) of the FENE bead-spring models with $b_{\max}=225$ up to 6 Hencky strain units at a representative *We* number of 20.0. The plots in Fig. 1 have been generated via Brownian dynamics simulation using the semi-implicit predictor–corrector scheme [14] with an ensemble size $N_t=1024$, and $\Delta t=10^{-3}$, and by averaging the steady state properties over 10,000 time steps. Also shown in Fig. 1 are the statistical error bars computed as $\sqrt{\sigma^2/N_t}$, where $\sigma^2$ is the variance. The FENE bead-spring models with $b_{\max}=900$ will be used in the plane Couette and Poiseuille flow benchmark problems, while the FENE bead-spring models with $b_{\max}=225$ will be used to simulate the 4:1:4 axisymmetric contraction–expansion flow. Since the zero shear relaxation times and the zero shear viscosities have been matched, as expected the viscometric shear properties in the limit of $\dot{\gamma}\to 0$ for the FENE bead-spring models with identical $b_{\max}$ values are also matched. In addition, as expected a more pronounced shear thinning of the viscometric shear properties with increasing number of segments is observed.

Table 1
Carreau model parameters that fit the steady shear viscosity of FENE bead-spring models ($b_{\max}=900$)

| *N* | *n* | *a* | *f* |
|---|---|---|---|
| 1 | 0.492 | 1.536 | 0.093 |
| 3 | 0.482 | 1.558 | 0.114 |
| 6 | 0.422 | 1.151 | 0.094 |

Fig. 1(a) also depicts the Carreau model fits for the shear viscosity, which will be used to demonstrate the fidelity of the algorithm by comparing the Carreau model-based steady state velocity profiles for the Poiseuille flow with those obtained using the self-consistent multiscale algorithm. The Carreau model [26] has been slightly modified by introducing a factor $f$ multiplying the *We* number (Eq. (25)) based on the premise that with an additional fitting parameter the shear viscosity data that is mostly in the transition regime can be captured more accurately. Specifically, the maximum relative error in the fits using the factor $f$ is 0.003, while it is 0.08 otherwise. Table 1 summarizes the Carreau model parameters that provide the best fit to the shear viscosities.

$$\frac{\eta_{\text{p}}}{\eta_0} = (1-\beta)[1+(f\ \ We)^a]^{(n-1)/a}. \tag{25}$$

## 5. Results and discussion

To demonstrate the fidelity and computational efficiency of our algorithm, the results of three benchmark flow problems, namely, the plane Couette, Poiseuille and 4:1:4 axisymmetric contraction–expansion are presented. In addition, the computational efficiency of the algorithm is directly compared with the fully implicit algorithm of Ramirez and Laso [14]. The results presented, unless specified otherwise are at $We=5$, $\beta=0.5$, for the FENE bead-spring models with $N=1$, 3, and 6 using two sets of Brownian configuration fields $N_{\text{f}}=1024$ and 2048, with the

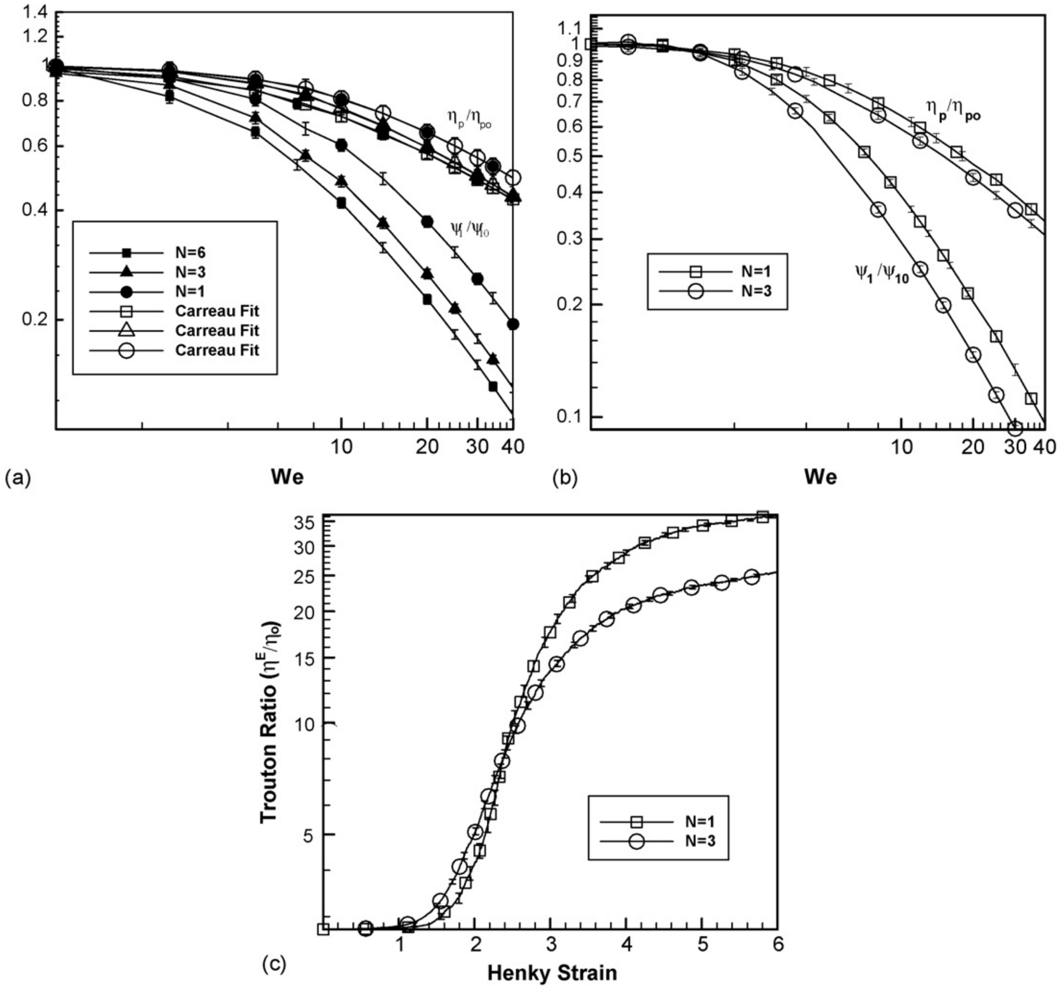

Fig. 1. Rheology of FENE bead-spring models. The figure includes the plots of steady shear viscosity and first normal stress coefficient vs. *We* for (a) $b_{\max} = 900$, and (b) $b_{\max} = 225$; transient extensional viscosity vs. Hencky strain for (c) $b_{\max} = 225$.

tolerance for the residue $\varepsilon = 10^{-3}$, and the time step $\Delta t = 10^{-3}$, which has been found to be sufficiently small for a wide range of *We* numbers in our earlier studies [7,15].

### 5.1. Plane Couette flow

The rationale behind selecting the plane Couette flow, namely, homogenous shear flow is the self-consistent multiscale simulation results can be directly compared with the Brownian dynamics simulation results. The flow domain $\Omega$, shown in Fig. 2(a), with the dimensionless height 1 unit and length $2\pi$ units has been divided into quadrilateral finite elements consisting of $5 \times 10$ elements of the same size shown in Fig. 2(c). The boundary conditions imposed are the standard no-slip boundary conditions along the walls, and the periodic boundary conditions at the entrance and exit to ensure that the flow is fully developed.

Fig. 3 depicts the transient normal and shear stress profiles, and the statistical error in the stress profiles. Table 2 summarizes the statistical error obtained at steady state, as expected the $\sqrt{1/N_{\mathrm{f}}}$ trend is observed. Fig. 3 also shows the transient normal and shear stress results from the Brownian dynamics (BD) simulations of the $N = 1$, 3, and 6 segment FENE bead-spring models subjected to shear flow at $We = 5$, $\beta = 0.5$, with an ensemble size $N_t = 2048$. The BD simulations have been performed employing the predictor–corrector scheme proposed in our earlier study [15]. As anticipated, the transient results from the BD simulations match the BCF results, thus, demonstrating the excellent accuracy of the self-consistent results obtained using our algorithm.

### 5.2. Poiseuille flow

The Poiseuille flow has been selected because the strain rate $(\dot{\gamma})$ varies along the cross-section providing a means to assess

Table 2
Statistical error of the polymeric stress components in the plane Couette flow

| $N$ | $N_{\mathrm{f}}$ | $\tau_{xx}$ | $\tau_{xy}$ |
|---|---|---|---|
| 1 | 1024 | 0.172 | 0.025 |
| 1 | 2048 | 0.120 | 0.018 |
| 3 | 1024 | 0.123 | 0.016 |
| 3 | 2048 | 0.091 | 0.012 |
| 6 | 1024 | 0.117 | 0.015 |
| 6 | 2048 | 0.088 | 0.011 |

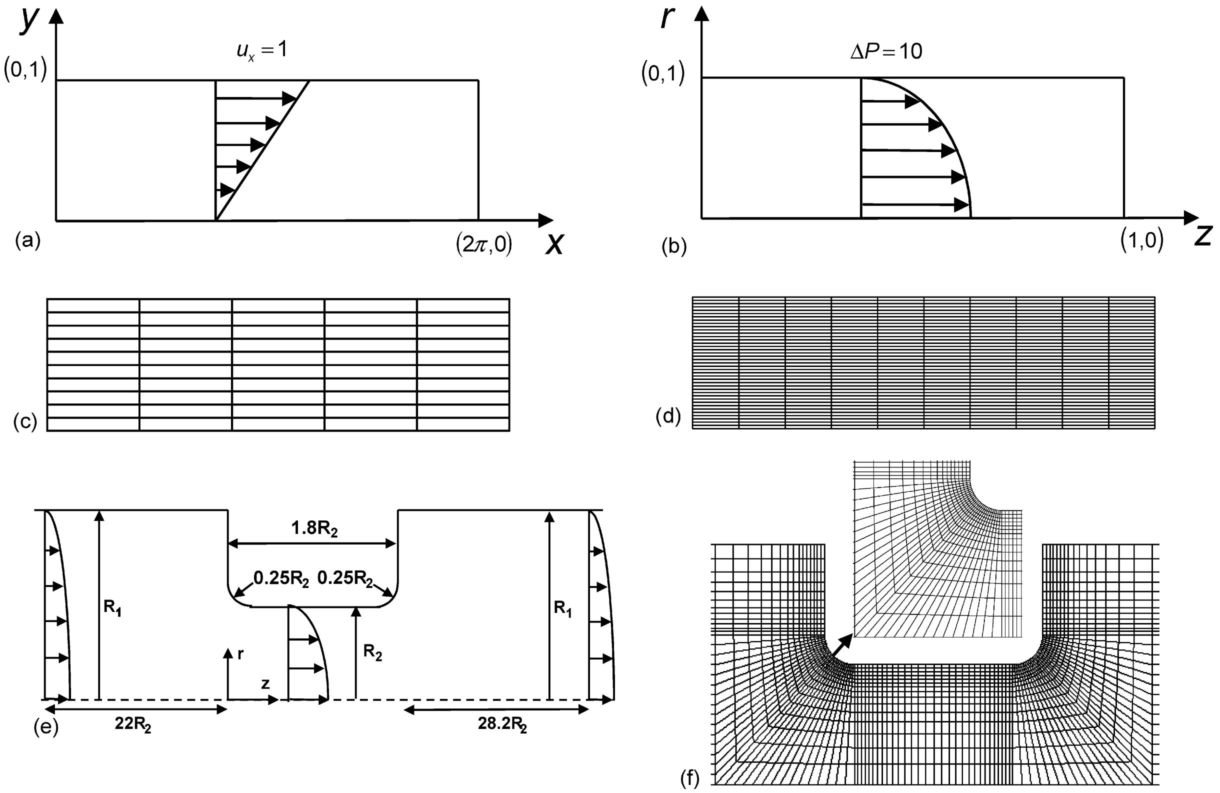

Fig. 2. Schematic diagram of: (a) plane Couette geometry; (b) Poiseuille flow geometry; (c) 50 element mesh; (d) 400 element mesh; (e) 4:1:4 axisymmetric contraction–expansion geometry; (f) mesh topology in the contraction region.

the fidelity of our algorithm in an inhomogenous shear flow by comparing the velocity and stress results with the Carreau model and BD predictions. The Poiseuille flow geometry, depicted in Fig. 2(b), with the dimensionless radius 1 unit and length 1 unit has been divided into quadrilateral finite elements consisting of 5 × 10 elements of the same size shown in Fig. 2(c). A dimensionless pressure drop $\Delta P = 10$ is imposed across the length of the pipe. The boundary conditions that have been imposed are the standard no-slip boundary condition along the wall ($r = 1$), the periodic boundary conditions at the entrance and exit to ensure that the flow is fully developed, and the symmetry boundary condition along the centerline ($r = 0$).

The steady state velocity profiles from the BCF simulations have been verified by a direct comparison with the Carreau model predictions (see Fig. 4). As evinced by the figure good agreement is obtained suggesting that the stress and velocity fields from the BCF simulations have been solved accurately. Fig. 4(d) shows the representative transient velocity profile using the three segment FENE bead-spring chain and $N_{\rm f} = 2048$. As expected, at time $t = 0$ the velocity profile is Newtonian since the initial polymeric stress is zero, and as the polymeric stress passes through the overshoot (expected for the range of strain rates observed near the wall) the velocity profile goes through a minimum to reach the final steady state.

The steady state stress profiles from the BCF simulations have been verified by comparing them with the results from the Brownian dynamics (BD) simulations at the local *We* number, $We_l = -We(\mathrm{d}u_z/\mathrm{d}r)$ at various radial positions shown in Fig. 5(a). Fig. 5 also shows the steady normal ($\tau_{rr}$) and shear ($\tau_{rz}$) stress profiles from the BCF and BD simulations as a function of the local *We* number with the appropriate statistical error bars. The comparison in Fig. 5 shows that the stress

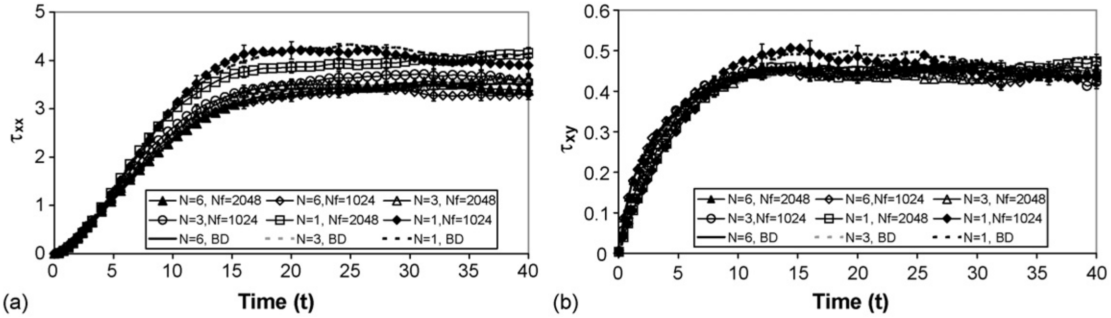

Fig. 3. Startup profiles of (a) $\tau_{xx}$ and (b) $\tau_{xy}$ for the plane Couette flow.

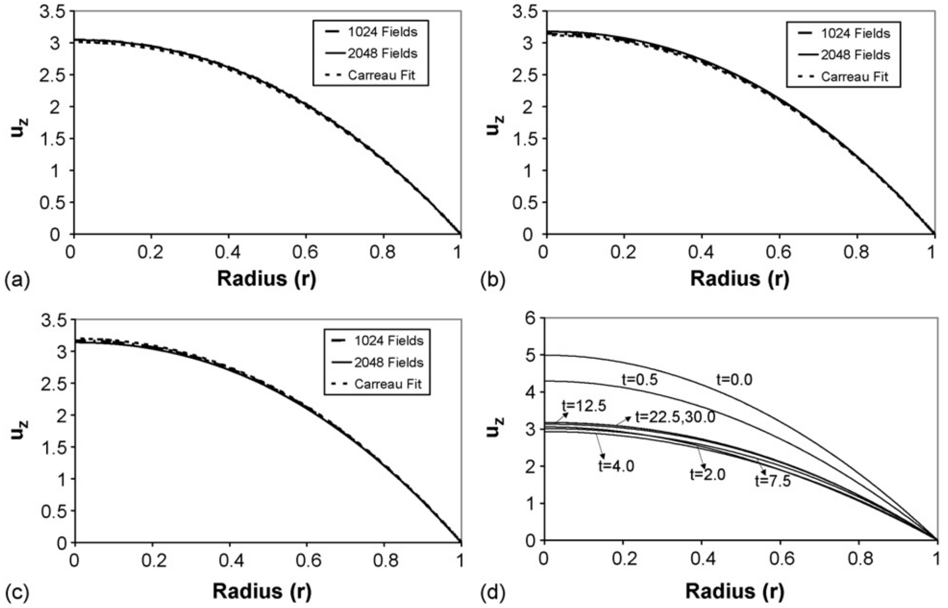

Fig. 4. Velocity profiles of the Poiseuille flow at steady state with (a) $N = 1$; (b) $N = 3$; (c) $N = 6$; during transience with (d) $N = 3$ and $N_f = 2048$.

profiles obtained from the BCF simulations are in good agreement with the corresponding BD results, thus, demonstrating the fidelity of the proposed algorithm for transient inhomogeneous flows. Furthermore, it should be noted that the stresses predicted by the different FENE models are closer to each other near the centerline (where $\dot{\gamma} \to 0$) because $\eta_{p0}$ and $\psi_{10}$ of the FENE models have been matched. As expected away from the centerline, the stresses predicted by the FENE bead-spring models with more number of segments are smaller. This is consistent with the trend observed as a function of $\dot{\gamma}$ shown in Fig. 1.

Fig. 6 shows the transient normal ($\tau_{rr}$) and shear ($\tau_{rz}$) stress profiles at different locations, $r = 1.0$, 0.6 and 0.2 across the radius. The stress profiles obtained using the two sets of Brownian configuration fields at the different radial locations and for the different number of segments are sufficiently close to each other, thus indicating the convergence of the results. The shear stress profiles on the wall ($r = 1$), as expected, satisfy the macroscopic force balance, which in terms of the dimensionless quantities is $\Delta P + 2\tau_{rz} + 2\beta(\mathrm{d}u_z/\mathrm{d}r) = 0$. The residue in the macroscopic force balance at steady state, relative to the net pressure force is on the order $10^{-4}$, indicating that the problem

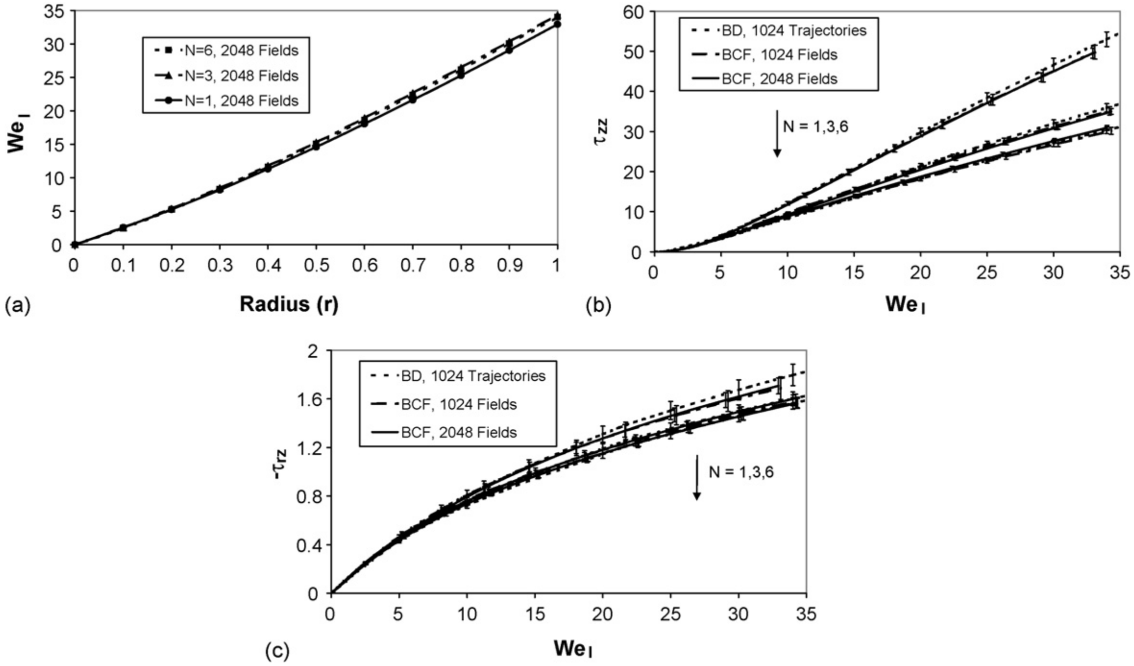

Fig. 5. Steady state profiles of: (a) $We_l$; (b) $\tau_{rr}$; (c) $\tau_{rz}$ for the Poiseuille flow.

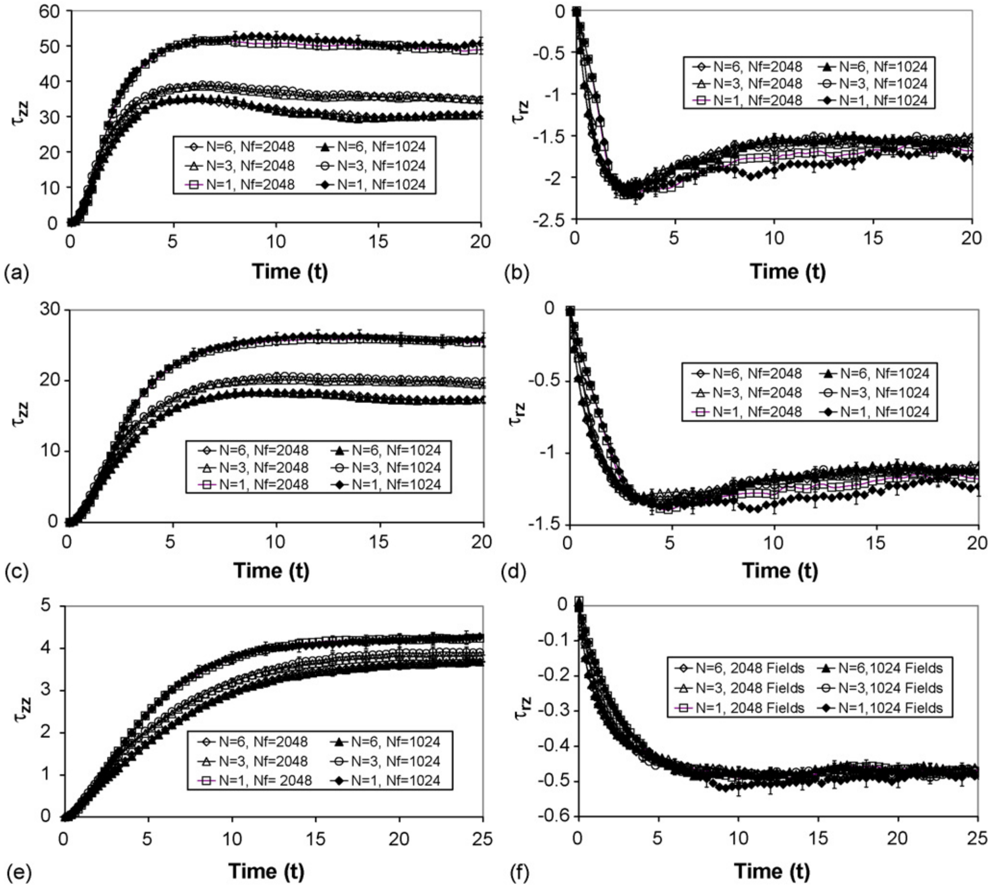

Fig. 6. Startup profiles of: (a) $\tau_{rr}$ at $r=1.0$; (b) $\tau_{rz}$ at $r=1.0$; (c) $\tau_{rr}$ at $r=0.6$; (d) $\tau_{rz}$ at $r=0.6$; (e) $\tau_{rr}$ at $r=0.2$; (f) $\tau_{rz}$ at $r=0.2$ for the Poiseuille flow.

has been solved self-consistently with adequate accuracy. The contribution of $\tau_{rz}$ to the macroscopic force balance is large, i.e., 33.76, 31.60, and 31.35% relative to the net pressure force for the $N=1$, 3, and 6 segment FENE bead-spring models, respectively, suggesting that the polymeric stress significantly modifies the Newtonian velocity profile in this problem.

### 5.3. 4:1:4 axisymmetric contraction–expansion flow

The 4:1:4 axisymmetric contraction–expansion flow has been chosen to demonstrate the fidelity of the algorithm in a complex kinematics flow. The schematic representation of the flow geometry is presented in Fig. 2(e). The contraction ratio ($R_1/R_2$) is selected to be 4, the re-entrant corner curvature ($R_c/R_2$) is fixed to 0.25, the contraction (throat) length ($L_c/R_2$) is set to 1.8, and the upstream and downstream sections of the flow geometry are chosen to be sufficiently long so that the fully developed flow can be assumed at the entrance and exit. The flow domain is divided into a finite element mesh consisting of 2736 quadrilateral elements, and the mesh topology near the contraction region is depicted in Fig. 2(f). The standard no-slip boundary conditions at the wall, symmetry boundary conditions along the centerline ($r=0$), periodic boundary conditions at the entrance and exit, and fully developed velocity boundary condition at the entrance are imposed. The characteristic velocity scale ($u_c$) is selected to be the maximum velocity at the entrance and the characteristic length scale ($L_c$) is selected to be the radius of the contraction tube $R_2$. The ratio of the solvent viscosity to the zero shear polymer viscosity ($\beta$) is set to 0.9231. The FENE bead-spring models with $b_{\max}=225$, and $N=1$ and 3 are used to characterize the polymer molecules.

BCF simulations were conducted with $N_f=1024$ adopting first-order continuation in the *We* number such that the pressure drop across the flow geometry (Fig. 2(e)) reaches steady state at each *We* number. Representative results at two *We* numbers, namely, 0.325 and 0.758, are presented. The steady state pressure drop at these *We* numbers is considerably different from the value (182.76) for the Newtonian flow (*We* = 0), specifically, at *We* = 0.325 it is 178.57 for $N=1$ and 180.25 for $N=3$, where as at *We* = 0.758 it is 186.36 and 188.35 for $N=1$ and 3, respectively. Fig. 7 shows the streamlines pattern at the two *We* numbers. At *We* = 0.325, the upstream vortex becomes slightly larger than the downstream vortex, and the pressure drop for both the FENE models ($N=1$ and 3) is less than the Newtonian pressure drop. On the contrary, at *We* = 0.758, for both the FENE models a significant growth of the upstream vortex size is observed and the pressure drop is also higher than the Newtonian value. Moreover, the upstream vortex size and the overall pressure drop is larger for $N=3$. This suggests that the internal degrees of freedom of the bead-spring model used to characterize the polymer molecules influence both the pressure drop and the vortex dynamics. A detailed analysis of the pressure drop and the vortex

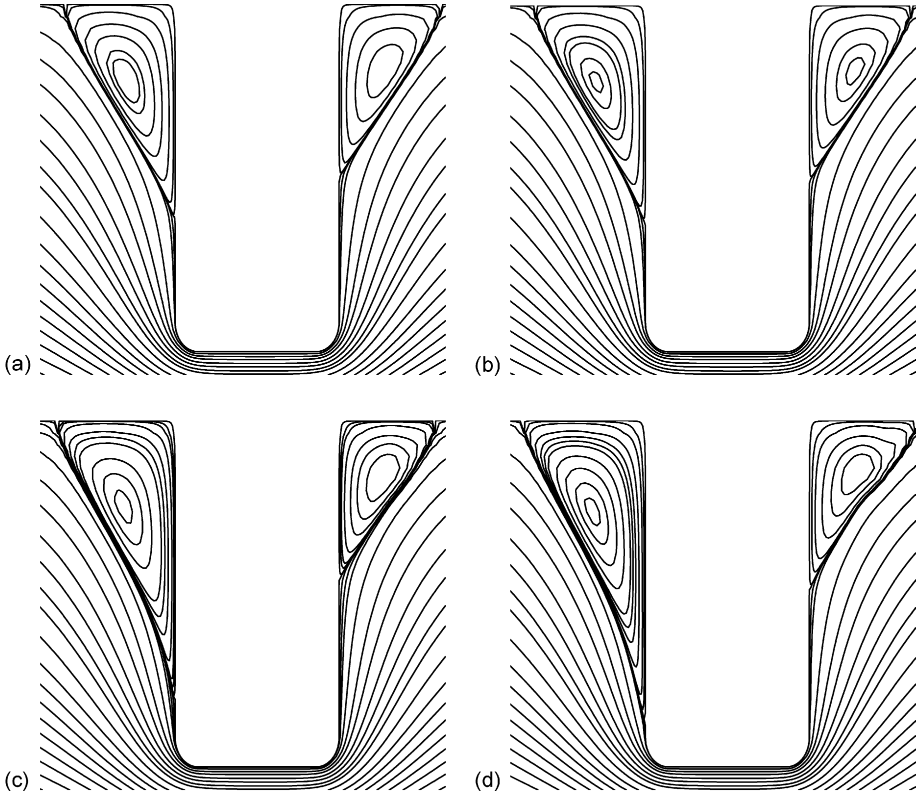

Fig. 7. Steady state streamlines pattern of the 4.1:4 contraction–expansion flow for FENE bead-spring models with (a) $N=1$ at $We=0.325$; (b) $N=3$ at $We=0.325$; (c) $N=1$ at $We=0.758$; (d) $N=3$ at $We=0.758$.

dynamics as a function of the *We* number, maximum extensibility, and internal degrees of freedom will be provided in an upcoming article.

Fig. 8 shows the steady state stress profiles ($\tau_{zz}$) in the contraction region along a streamline near the wall and along the center streamline ($r=0$), respectively, for $We=0.325$ and $We=0.758$. $\tau_{zz}$ has been selected to show the accuracy of the computations mainly because of its large magnitude and significant variation in the contraction region. The stress profiles along selected streamlines depicted in Fig. 8(e) and (f) have been obtained via Lagrangian integration utilizing the self-consistent local velocity vectors and a fourth-order Runge–Kutta method with a time step $\Delta t=10^{-3}$. The stress profiles follow the expected trend, that is, along the streamline near the wall the stress grows steeply due to the excessive stretching around the corners accompanied by the stress relaxation between the corners, and along the center streamlines the stress increases smoothly followed by a smooth decrease as the fluid accelerates and then decelerates. The comparison of the BCF results with those from the BD results, obtained using the predictor–corrector scheme [15] with the ensemble size $N_t=1024$ following the Lagrangian approach along the streamlines demonstrates the accuracy of the semi-implicit algorithm.

### 5.4. Computational efficiency

In this section, we present the CPU time comparison of the algorithm for the plane Couette and Poiseuille flow benchmark problems, described in Sections 5.1 and 5.2 using the original $5 \times 10$ element mesh (Fig. 2(c)), and a mesh consisting of $10 \times 40$ uniformly sized quadrilateral elements (Fig. 2(d)) for the Poiseuille flow problem. The CPU time study has been conducted on the 64 processor SGI Origin 2000 machine using the time step $\Delta t=10^{-3}$, and applying the set of rules defined in Section 3.1 to accelerate the computations. The CPU time is primarily effected by three parameters: the number of segments in the bead-spring chain ($N$), the number of Brownian configuration fields used ($N_\mathrm{f}$), and the number of processors chosen ($N_\mathrm{p}$) for the parallelization. We have studied the effect of all of these parameters on the CPU time and the results are shown in Fig. 9 in which the CPU times are scaled with those shown in Table 3. The CPU time that we report in this study is the CPU time needed to run 450 time steps, that is, the difference between the CPU times for the initial 500 and 50 time steps. Subtract-

Table 3
CPU time for different number of processors, fields and segments

| CPU | Parameters fixed | Elements | Flow | Time (min) |
|---|---|---|---|---|
| $\mathrm{CPU}_{N=1}$ | $N_\mathrm{f}=1024$, $N_\mathrm{p}=8$ | 400 | Poisuille | 55.6 |
| | | 50 | Poisuille | 5.3 |
| | | 50 | Plane Couette | 4.0 |
| $\mathrm{CPU}_{N_\mathrm{f}=512}$ | $N=3$, $N_\mathrm{p}=8$ | 400 | Poisuille | 87.5 |
| | | 50 | Poisuille | 8.4 |
| | | 50 | Plane Couette | 6.88 |
| $\mathrm{CPU}_{N_\mathrm{p}=2}$ | $N=3$, $N_\mathrm{f}=1024$ | 400 | Poisuille | 667.34 |
| | | 50 | Poisuille | 68.02 |
| | | 50 | Plane Couette | 49.63 |

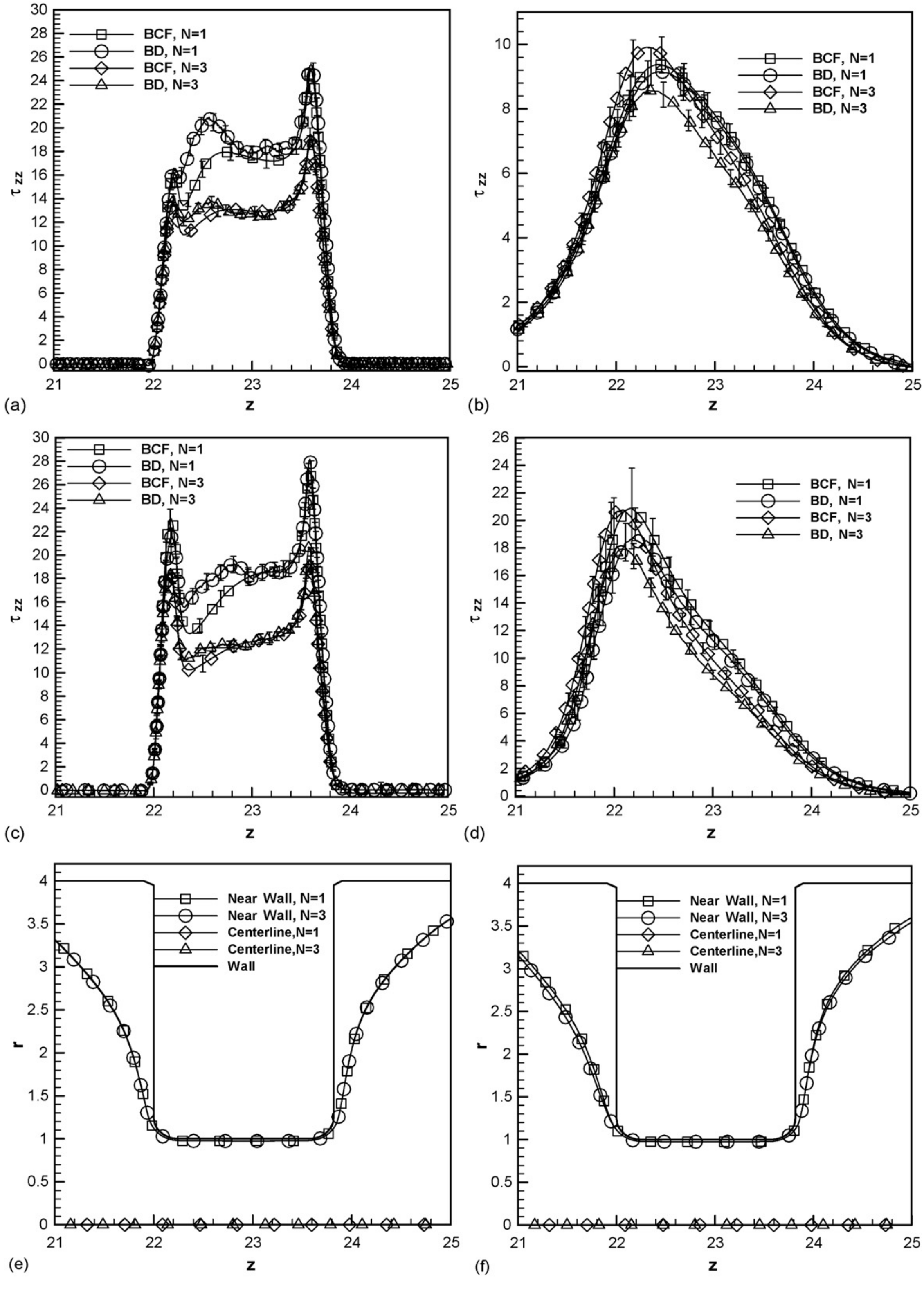

Fig. 8. Stress profiles ($\tau_{zz}$) along selected streamline originating from: (a) (0,3.9) at $We = 0.325$; (b) (0,0) at $We = 0.325$; (c) (0,3.9) at $We = 0.758$; (d) (0,0) at $We = 0.758$. Also shown are the paths of the selected streamlines at (e) $We = 0.325$ and (f) $We = 0.758$.

ing the CPU time for the first 50 time steps eliminates the time devoted to the initialization and data transfer processes, therefore, a better estimation of the CPU performance is obtained. Furthermore, we chose the first 500 time steps to get a conservative estimation of the CPU time since during the initial stage of a startup flow rapid changes to the polymer configuration are expected, therefore, a higher number of corrector steps might be needed.

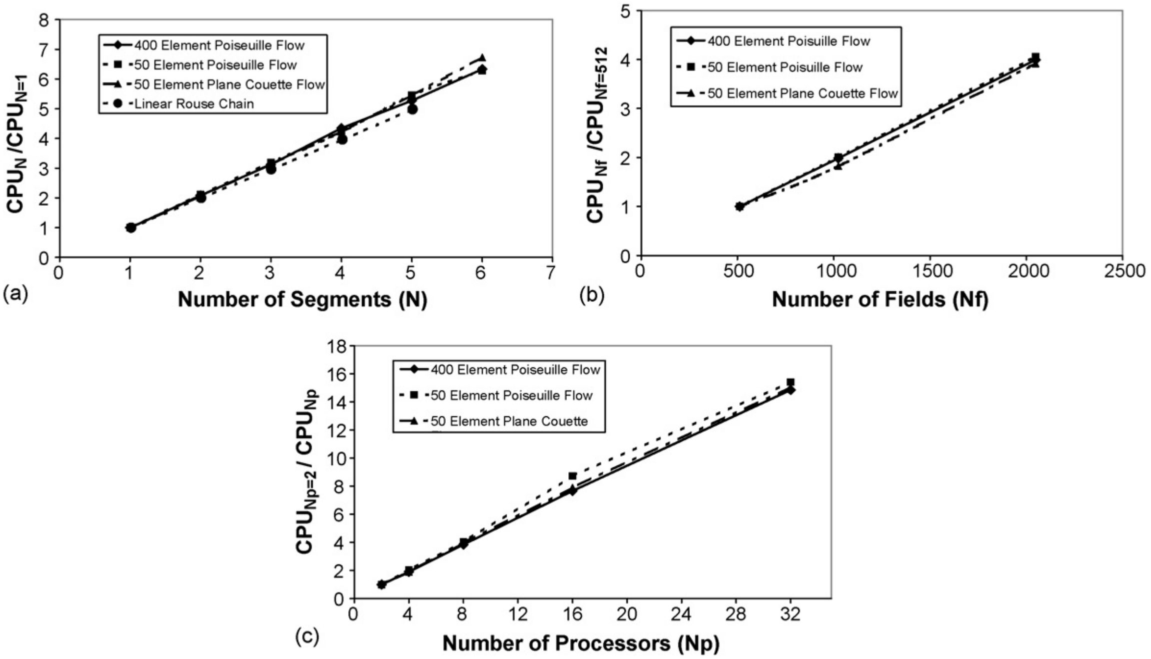

Fig. 9. CPU time for 450 time steps with the fixed parameters shown in Table 3 and varying: (a) number of segments; (b) number of configuration fields; (c) number of processors.

Fig. 9(a) shows the plot of the CPU time needed as a function of the number segments, $N = 1$–6, used in the FENE bead-spring model. The CPU time increases linearly with the number of segments, which is similar to the trend observed using linear Rouse chains (i.e., linear scale up is expected for Rouse chains) in the fully implicit method by Ramirez and Laso [14]. This is due to the particular choice of implementation, that is, the explicit update (Eq. (18)) of the semi-implicit solution at the local Gaussian qudrature points for each segment (Eq. (20)), which reduces the scale up of the CPU time with the number of segments for the non-linear FENE force law to be approximately the same as the linear force law. The slopes of all the three CPU plots in Fig. 9 are close to one for the first three segments, thereafter the slopes gradually increase to a value slightly above one. The increase in the slope is due to the fact that as the number of segments increase the maximum extensibility of each segment decreases, therefore, small deviations in the computation of the configuration fields may lead to a large residue, requiring additional corrector steps for the residue to meet the tolerance. Therefore, the increase in the slope can be attributed to a pronounced non-linear effect of the FENE force law. Clearly, increasing the number of configuration fields increases the CPU time linearly with a slope close to one provided the communication requirements are minor. This issue is handled efficiency in our computations (see Fig. 9(b)).

In order to directly compare the CPU time of the proposed algorithm with the fully implicit method by Ramirez and Laso [14] we simulated the startup of plane Couette flow of Hookean dumbbells on a single Intel Xeon computer with a CPU speed of 2.80 GHz using the 400 element mesh (Fig. 2(d)) at $We = 1$, $\beta = 0.5$, using $N_f = 1000$, with $\Delta t = 0.1$, and $\varepsilon = 10^{-3}$. These are the same simulation parameters as in the fully implicit method [14]. The comparison of the CPU time of the two methods showed that our method is approximately 50 times faster than the fully implicit method to perform the time integration of 10 dimensionless time units. A more comprehensive comparison of the CPU time and memory requirements of both the methods is underway, and these results will be a subject of a future communication. Furthermore, a comparison of the CPU time with the earlier semi-implicit algorithm [7] for dumbbell models, where the spring force is treated implicitly showed that the proposed algorithm is nearly four times faster in the case of the 400 element Poiseuille flow problem using $N_f = 2048$.

Fig. 9(c) shows the scale-up efficiency of the parallelization scheme with the number of processors used. The scale-up efficiency study has been performed using 2, 4, 8, 16 and 32 processors, and the scale-up efficiency was found to be about 90%. In the parallelization scheme, only the macroscopic variables are communicated, i.e., the velocity, velocity gradients, pressure and stress variables, so the scale-up efficiency will be affected by the mesh size. The comparison of the scale-up efficiency of the 400 element mesh and the 50 element mesh Poiseuille flow problems shows that the scale-up performance is not significantly different. Usually large finite element meshes are needed for solving complex flow problems, in such situations the scale-up efficiency will be slightly smaller, however, since the data communication involved is not very intensive we anticipate that the scale up would still be very efficient.

## 6. Summary

In this study, an efficient algorithm for multiscale flow simulation of dilute polymer solutions using bead-spring chain description of the polymer molecules has been presented. The algorithm self-consistently solves the stress and velocity fields at each time step in a decoupled fashion employing a highly computationally efficient semi-implicit predictor–corrector scheme to reduce the non-linear BCF equations into a linear system of equations. Flow of dilute polymeric solutions in three benchmark flow problems, namely, plane Couette, Poiseuille and 4:1:4 axisymmetric contraction–expansion have been simulated on a parallel hardware architecture using FENE bead-spring models with the number of segments varying from 1 to 6. The results of the benchmark problems have been verified by a direct comparison with Brownian dynamics results, and with the steady state velocity profiles obtained via a Carreau model description, as well as by performing streamline integration. The comparison showed that the results are in good agreement, hence demonstrating the fidelity of the algorithm.

The results also demonstrate the linear scale up of the CPU time with the number of segments, which is due to the semi-implicit treatment of the spring force that facilitates the performing of the BCF computations as if the segments were decoupled. In addition, due to the particular choice of our implementation, that is, obtaining the solution for the spring force at the local Gaussian quadrature points and employing it in the explicit update of the configuration field results in large saving in the computation time. Furthermore, a simple acceleration scheme to speed up the computations is implemented that results in a 40% saving in the computation time. A direct comparison of the algorithm with the fully implicit method by Ramirez and Laso [14] showed that the algorithm is approximately 50 times faster under the same simulation conditions. Currently, we are performing multiscale simulations of 4:1:4 axisymmetric contraction–expansion flows using the proposed algorithm and the results of these computations will be a topic of future communication.

## Acknowledgements

The authors acknowledge the National Science Foundation for supporting this work through grants CTS-0095098 and CTS-0335348.

## References

[1] M. Laso, H.C. Ottinger, Calculation of viscoelastic flow using molecular models: the CONNFFESSIT approach, J. Non-Newtonian Fluid Mech. 47 (1993) 1–20.

[2] 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.

[3] 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.

[4] 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.

[5] H.C. Ottinger, B.H.A.A. van den Brule, M.A. Hulsen, Brownian configuration fields and variance reduced CONNFFESSIT, J. Non-Newtonian Fluid Mech. 70 (1997) 255–261.

[6] J. Bonvin, M. Picasso, Variance reduction methods for CONNFFESSIT-like simulations, J. Non-Newtonian Fluid Mech. 84 (1999) 191–215.

[7] M. Somasi, B. Khomami, Linear stability and dynamics of viscoelastic flows using time-dependent stochastic simulation techniques, J. Non-Newtonian Fluid Mech. 93 (2000) 339–362.

[8] M. Bajaj, P.P. Bhat, J.R. Prakash, M. Pasguali, Multiscale simulation of viscoelastic free surface flows, J. Non-Newtonian Fluid Mech. 140 (2006) 87–107.

[9] M. Laso, J. Ramirez, M. Picasso, Implicit micro-macro methods, J. Non-Newtonian Fluid Mech. 122 (2004) 215–226.

[10] 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.

[11] J.K.C. Suen, Y.L. Joo, R.C. Armstrong, Molecular orientation effects in viscoelasticity, Annu. Rev. Fluid. Mech. 34 (2002) 417–444.

[12] E.S.G. Shaqfeh, The dynamics of single-molecule DNA in flow, J. Non-Newtonian Fluid Mech. 130 (2005) 1–28.

[13] R.G. Larson, The rheology of dilute solutions of flexible polymers: progress and problems, J. Rheol. 49 (2005) 1–70.

[14] J. Ramirez, M. Laso, Size reduction methods for the implicit time-dependent simulation of micro-macro viscoelastic flow problems, J. Non-Newtonian Fluid Mech. 127 (2005) 41–49.

[15] M. Somasi, B. Khomami, N.J. Woo, J.S. Hur, E.S.G. Shaqfeh, Brownian dynamics simulations of bead-rod and bead-spring chains: numerical algorithms and coarse-graining issues, J. Non-Newtonian Fluid Mech. 108 (2002) 227–255.

[16] R.B. Bird, C.F. Curtiss, R.C. Armstrong, O. Hassager, Dynamics of Polymeric Liquids, vol. 2, Wiley, New York, 1987.

[17] J.M. Wiest, R.I. Tanner, Rheology of bead-nonlinear spring chain macromolecules, J. Rheol. 33 (2) (1989) 281–316.

[18] R. Guenette, M. Fortin, A new mixed finite element method for computing viscoelastic flows, J. Non-Newtonian Fluid Mech. 60 (1995) 27–52.

[19] A.W. Liu, D.E. Bornside, R.C. Armstrong, R.A. Brown, Viscoelastic flow of polymer solutions around a periodic, linear array of cylinders: comparisons of predictions for microstructure and flow fields, J. Non-Newtonian Fluid Mech. 77 (1998) 153–190.

[20] J. Li, W.R. Burghardt, B. Yang, B. Khomami, Birefringence and computational studies of a polystyrene Boger fluid in axisymmetric stagnation flow, J. Non-Newtonian Fluid Mech. 91 (2000) 189–220.

[21] B. Szabo, I. Babuska, Finite Element Analysis, Wiley, New York, 1987.

[22] K.K. Talwar, H.K. Ganpule, B. Khomami, A note on selection of spaces in computation of viscoelastic flows using the hp-finite element method, J. Non-Newtonian Fluid Mech. 52 (1994) 293–307.

[23] H.C. Ottinger, Stochastic Processes in Polymeric Fluids, Springer, Berlin, 1996.

[24] I. Ghosh, Y.L. Joo, G.H. McKinley, R.A. Brown, R.C. Armstrong, A new model for dilute polymer solutions in flows with strong extensional components, J. Rheol. 46 (5) (2002) 1057–1089.

[25] A.P.G. van Heel, M.A. Hulsen, B.H.A.A. van den Brule, On the selection of parameters in the FENE-P model, J. Non-Newtonian Fluid Mech. 75 (1998) 253–271.

[26] R.B. Bird, R.C. Armstrong, O. Hassager, Dynamics of Polymeric Liquids, vol. 1, Wiley, New York, 1987.