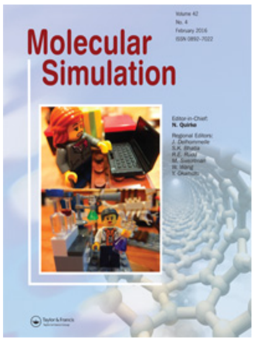
Lees–Edwards boundary condition for simulation of polymer suspension with dissipative particle dynamics method

Article *in* Molecular Simulation · July 2015

DOI: 10.1080/08927022.2015.1044455

Dingyi Pan
Zhejiang University
46 PUBLICATIONS **703** CITATIONS

Shao Xueming
Zhejiang University
119 PUBLICATIONS **1,900** CITATIONS



Molecular Simulation

ISSN: 0892-7022 (Print) 1029-0435 (Online) Journal homepage: http://www.tandfonline.com/loi/gmos20

Lees–Edwards boundary condition for simulation of polymer suspension with dissipative particle dynamics method

Dingyi Pan, Jianxin Hu & Xueming Shao

To cite this article: Dingyi Pan, Jianxin Hu & Xueming Shao (2016) Lees–Edwards boundary condition for simulation of polymer suspension with dissipative particle dynamics method, Molecular Simulation, 42:4, 328-336, DOI: 10.1080/08927022.2015.1044455

To link to this article: http://dx.doi.org/10.1080/08927022.2015.1044455

Published online: 07 Jul 2015.

Molecular Simulation, 2016
Vol. 42, No. 4, 328–336, http://dx.doi.org/10.1080/08927022.2015.1044455

Lees–Edwards boundary condition for simulation of polymer suspension with dissipative particle dynamics method

Dingyi Pan[a]*, Jianxin Hu[b]* and Xueming Shao[a]

[a]*State Key Laboratory of Fluid Power Transmission and Control, Department of Engineering Mechanics, Zhejiang University, Hangzhou 310027, P.R. China;* [b]*Department of Engineering Science, Oxford University, Oxford OX1 3PJ, UK*

(*Received 12 January 2015; final version received 19 April 2015*)

The Lees–Edwards boundary condition (LEbc) is widely used in particle-based simulation for producing shear flow. Application of traditional LEbc in dissipative particle dynamics (DPD) method may encounter certain problems, e.g. it will destroy the momentum conservation law at the near boundary region, and the coordinate system gives an incorrect end-to-end vector for polymer beads. Special treatments of the implementation of LEbc in DPD method are introduced in this paper. A single side ghost layer is used to keep the momentum conservation, and the global coordinate system is employed to obtain a correct calculation of the spring force between polymer beads. The simulation results give a good prediction of velocity profile and system temperature, and the elastic dumbbell model for current method can well represent the Oldroyd-B fluid.

Keywords: dissipative particle dynamics; Lees–Edwards; shear flow; polymer chain

1. Introduction

The Lees–Edwards boundary condition (LEbc) was first developed by Lees and Edwards [1] in 1970s to produce a simple shear flow in molecular dynamics (MD) simulation. In traditional simulation, the sliding solid walls are used to generate this flow motion, while in LEbc, the bi-periodic boundary conditions are used to reduce the finite size effect and eliminate the wall effects as well as the spatiality in homogeneities. The application of LEbc has been extended in a wide range in past three decades, which includes LEbc for dissipative particle dynamics (DPD),[2–4] LEbc for lattice Boltzmann simulation,[5] LEbc for direct simulation of particle suspension,[6,7] etc.

The implementation of LEbc is straightforward. As shown in Figure 1, the grey rectangle (frame) represents the computation domain, and two imaginary (ghost) frames are located on the top and bottom of the grey one, respectively. Initially, the three frames locate along the vertical direction. As the shear starts, the two imaginary frames begin to slide along the shear direction, and the slide length can be calculated according to the shear rate $\dot{\gamma}$ as

$$\delta = \dot{\gamma} W t, \tag{1}$$

where W is the width of the frame. Since there are no driving walls which confine the computation domain, only the periodic boundary conditions are used at the top and bottom boundaries. Once the particle moves out of the computation domain, it will be reintroduced into the domain from the opposite side. Taking particles i' and j' as examples, these two particles will re-enter into the domain as i and j with locations and velocities assigned as

$$x_i(t) = \begin{cases} \left(x_{i'}(t) - \dot{\gamma} W t\right) \mathbf{mod}\, L, & y_{i'}(t) \geq \frac{W}{2}, \\ \left(x_{i'}(t) + \dot{\gamma} W t\right) \mathbf{mod}\, L, & y_{i'}(t) < -\frac{W}{2}, \end{cases} \tag{2}$$
$$y_i(t) = y_{i'}(t)\, \mathbf{mod}\, W,$$

$$u_i(t) = \begin{cases} u_{i'}(t) - \dot{\gamma} W, & y_{i'}(t) \geq \frac{W}{2}, \\ u_{i'}(t) + \dot{\gamma} W, & 4y_{i'}(t) < -\frac{W}{2}, \end{cases} \tag{3}$$
$$v_i(t) = v_{i'}(t).$$

Here, **mod** represents the modular function and L is the length of the computation domain. With this boundary condition, a simple shear flow with shear rate $\dot{\gamma}$ is generated. Since in LEbc the particle moving out of the top or bottom boundary is re-entering into the domain, the LEbc is considered as one kind of the bi-periodic boundary conditions.

As a particle-based numerical method, the DPD method can also be applied to LEbc to produce shear flow. The DPD method, first introduced by Hoogerbrugge and Koelman [8] is meant to be a meso-scaled simulation technique yielding correct hydrodynamic behaviours and has its basis in statistical mechanics.[9,10] The DPD particles with soft potential is considered as a lump of molecules, and the method was conceived as an improvement over MD with high computational effi-

*Corresponding authors. Email: dpan@zju.edu.cn; jianxin.hu@eng.ox.ac.uk

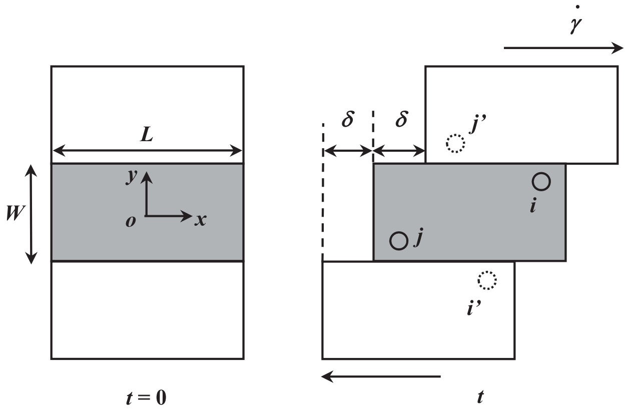


Figure 1. Implementation of original LEbc model.

ciency. As an attractive and powerful simulation tool, the application of DPD method is extended widely, especially in the rheological studies on complex fluid, e.g. rigid particle suspension,[11,12] deformable droplet suspension,[13,14] polymer and biological macromolecular suspension,[15,16] etc.

The simple shear flow is normally used as a canonical flow to investigate the rhological properties of fluid. Hence, simulation of shear flow is usually used to mimic the viscometer. In original DPD simulations, sliding solid walls formed with *frozen* particles are used to generate the shear flow. Unfortunately, this kind of simulation always suffers from density fluctuation in the region close to *frozen* particles. Recent efforts have been made to overcome this problem, by combining with a more complicated algorithm.[17,18] On the other hand, LEbc provides a simple way, and meanwhile eliminate the density fluctuation. However, two issues are required to fix when LEbc is applied in DPD simulation. First, the conservation of momentum in DPD is automatically fulfilled by pairwise interactions between every two DPD particles, whereas in LEbc for DPD simulation, the momentum conservation might be broken up during the calculation of random forces for particle interaction across the boundary. This problem also occurred when people parallelize DPD simulation with a message passing interface (MPI) technique.[19] Second, concerning DPD simulation of polymer suspension, once the spring between two beads is crossing the boundary, which means the beads are located in two different frames. In order to calculate the spring force, the position of one bead is needed to revert back with respect to the same frame of the other beads. To the knowledge of the authors, these two issues have not been discussed in the literatures. In this paper, special treatments of these two issues are provided, and relevant numerical tests are present. In the following sections, an overview of the DPD method is given in Section 2, relevant improvements of LEbc in DPD and numerical tests are provided in Sections 3 and 4, and the conclusion is drawn in Section 5.

2. Overview of DPD method

In DPD method, each DPD particle i is governed by Newton's second law of motion

$$\frac{\mathrm{d}\mathbf{r}_i}{\mathrm{d}t} = \mathbf{v}_i, \quad m\frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = \mathbf{f}_i + \mathbf{F}^e, \tag{4}$$

where $\mathbf{r}_i$ and $\mathbf{v}_i$ are the position and velocity vectors of particle i, m is the mass of a DPD particle, $\mathbf{f}_i$ is the summation of inter-particle forces exerted by all the other particles and $\mathbf{F}^e$ is the external. Español and Warren [9] proposed that the inter-particle force $\mathbf{f}_i$ can be decomposed by three parts, i.e. conservative force, dissipative force and random force, and all are pairwise and centre-to-centre:

$$\mathbf{f}_i = \sum_{j \neq i} \left(\mathbf{F}_{ij}^C + \mathbf{F}_{ij}^D + \mathbf{F}_{ij}^R \right). \tag{5}$$

Here, the sum runs over all other particles within a certain cutoff radius r_C, which is usually taken as unity ($r_C = 1$). Normally, the conservative force $\mathbf{F}_{ij}^C$ is a soft repulsion acting along the line of centres [20]

$$\mathbf{F}_{ij}^C = \begin{cases} a_{ij}(1 - r_{ij})\hat{\mathbf{r}}_{ij}, & r_{ij} < 1, \\ 0, & r_{ij} \geq 1, \end{cases} \tag{6}$$

where a_{ij} is the maximum repulsion between particle i and particle j, and $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, $r_{ij} = \left|\mathbf{r}_{ij}\right|$, $\hat{\mathbf{r}}_{ij} = \mathbf{r}_{ij}/\left|\mathbf{r}_{ij}t\right|$. The other two forces are dissipative force $\mathbf{F}_{ij}^D$ and random force $\mathbf{F}_{ij}^R$, given by:

$$\mathbf{F}_{ij}^D = -\gamma w^D\left(r_{ij}\right)\left(\hat{\mathbf{r}}_{ij} \cdot \mathbf{v}_{ij}\right)\hat{\mathbf{r}}_{ij}, \tag{7}$$

and

$$\mathbf{F}_{ij}^R = \sigma w^R\left(r_{ij}\right)\theta_{ij}\hat{\mathbf{r}}_{ij}, \tag{8}$$

respectively. Here, γ and σ are two coefficients characterizing the strength of dissipative force and random force; $w^D(r)$ and $w^R(r)$ are the weight functions for these two forces vanishing for $r \geq r_C$; $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$ is the relative velocity and θ_{ij} is a white noise with the properties

$$\left\langle \theta_{ij}(t) \right\rangle = 0$$

and

$$\left\langle \theta_{ij}(t)\theta_{kl}(t') \right\rangle = \left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right)\delta(t - t'),$$

where $\langle \cdots \rangle$ denotes an ensemble average with respect to its own distribution function.

On the other hand, Español and Warren [9] also showed that one of the two weight functions in Equations

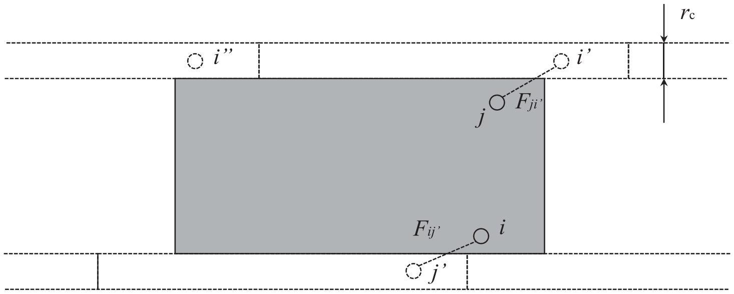


Figure 2. Implementation of normal LEbc model with DPD method.

(7) and (8) can be chosen arbitrarily and then the other is determined by

$$w^D(r) = \left[w^R(r)\right]^2, \quad \text{and} \quad \sigma^2 = 2\gamma k_B T, \tag{9}$$

where $k_B T$ is the Boltzmann temperature of the system (a measure of the specific fluctuating kinetic energy of the system). With this relation, the fluctuation–dissipation theorem can be satisfied. In our simulation, we employed a generalized form for this weight function proposed in Fan et al. [15]

$$w^D(r) = \left[w^R(r)\right]^2 = \begin{cases} \left(1 - \frac{r}{r_C}\right)^s, & r_{ij} < r_C, \\ 0, & r_{ij} \geq r_C, \end{cases} \tag{10}$$

where s is exponent. In the conventional DPD system as introduced by Groot and Warren [20] it is set as $s = 2$, while in order to improve the dynamic response and increase the Schmidt number of the DPD system, Fan et al. [15] preferred $s = 1/2$ and $1.0 \leq r_C \leq 1.5$. Throughout this paper, we employ $r_C = 1.0$ and $s = 1/2$.

Furthermore, in our numerical simulation, the initial velocities of all the particles are set randomly according to the system temperature $k_B T$. The explicit velocity-Verlet algorithm [20] is used to solve Equation (4). The stress tensor $S_{\alpha\beta}$ is calculated using the Irving–Kirkwood model,[21] where the contribution from each particle on the stress can be divided into two parts:

$$S_{\alpha\beta} = -\frac{1}{V}\left\langle \sum_i^{N_P} m u_{i\alpha} u_{i\beta} + \sum_i^{N_p} \sum_{j>i}^{N_p} r_{ij\alpha} F_{ij\beta} \right\rangle. \tag{11}$$

Here N_p is the number of particles, $u_{i\alpha}$ and $u_{i\beta}$ are the peculiar velocity components of particle i, defined by $u_{i\alpha} = v_{i\alpha} - \bar{v}_\alpha(\mathbf{x})$ ($v_{i\alpha}$ is the velocity component of particle i and $\bar{v}_\alpha(\mathbf{x})$ the stream velocity at position $\mathbf{x}$), and $F_{ij\beta}$ is the inter-particle force component between particle i and particle j.

3. Implementation of LEbc in DPD

The implementation of LEbc in normal particle-based method is shown in Figure. 2. Two imaginary (ghost) layers are created on the top and bottom of the computation domain, accounting the particle interactions which are crossing the top and bottom boundaries. The width of the layer is fixed as r_C, since particles can only interact with each other within this distance. The particle information in the top and bottom imaginary layers is reverted from the bottommost layer and the topmost layer inside of the computation domain. Taking particle i and particle j in Figure 2 as examples, in order to take account of the interaction between them, the imaginary particle i' or j' is created in the layers, while the position and velocity of the imaginary particle is reverted following Equations (2) and (3). It is worthy to mention that, the particle i'' will be introduced once the imaginary particle i' is out of the boundary along the x-direction.

As mentioned in the introduction, due to the nature of DPD method, special treatment on the LEbc is required when it is applied in DPD. First, the conservation of momentum should be fulfilled for particle interaction crossing the LEbc. Second, the distance of two neighbour beads in polymer should be calculated with respect to the same frame. In the following sections, the special implementation of LEbc in DPD is discussed in detail.

3.1 The momentum conservation for LEbc in DPD

Normally, two imaginary layers are created to simplify the calculation of interaction force which is crossing the boundary. As shown in Figure 2, particles i' and j' are the ghost particles with respect to particles i and j. Therefore, the interaction force between i and j is calculated independently as $\mathbf{F}_{ij'}$ for i and $\mathbf{F}_{i'j}$ for j. Meanwhile, in order to fulfill the momentum conservation, it is required that

$$\mathbf{F}_{ij'} = -\mathbf{F}_{i'j}. \tag{12}$$

In DPD simulation, considering the force components mentioned in Equations (6)–(8), both the repulsive force and dissipative force satisfy Equation (12) automatically since the relative distance and velocity between the particles are invariant. On the other hand, the random number generator is called twice to assign values to $\theta_{ij'}$ and $\theta_{i'j}$. Since there is no constrain assuring $\theta_{ij'} = \theta_{i'j}$, the conservation of momentum might be destroyed as $\mathbf{F}^R_{ij'} \neq -\mathbf{F}^R_{i'j}$. This situation also happens in domain decomposing for paralleling DPD simulation with MPI algorithm. One solution of this problem is proposed by Afshar et al. [19] and they developed a seeded pseudo random number generator which can generate a unique random number according to particles' label i and j.

We propose a straightforward approach to overcome this problem. In our simulations, only one imaginary layer, e.g. the top layer, is employed, and hence the interaction between particles i and j is only calculated once as $\mathbf{F}_{i'j}$. After that, the force contribution to particle i is sent to Newton's second law equation directly. In detail, the algorithm can be demonstrated using the following steps.

(1) The top imaginary layer is created by coping and modifying the particle information from the bottommost layer of the computational domain, in which the position of DPD particles satisfies $y_i < -W/2 + r_C$. The newly created imaginary particle i' is assigned with a modified position and velocity as

$$\begin{aligned} x_{i'} &= (x_i + \dot{\gamma}Wt)\,\mathbf{mod}\,L, \quad y_{i'} = y_i + W, \\ u_{i'} &= u_i + \dot{\gamma}W, \quad v_{i'} = v_i. \end{aligned} \tag{13}$$

(2) The interacting forces between particles j and i' are calculated as $\mathbf{F}_{ji'} = -\mathbf{F}_{i'j}$, following Equations (6)–(8).

(3) Adding this interacting force to the total force of particle i and j, respectively, as

$$\begin{aligned} \mathbf{f}_i &= \mathbf{f}_i + \mathbf{F}_{i'j}, \\ \mathbf{f}_j &= \mathbf{f}_j + \mathbf{F}_{ji'}. \end{aligned} \tag{14}$$

3.2 Calculation of spring force in polymer with LEbc

The second issue for LEbc in DPD that should be figured out is its application in polymer suspension simulation. Intrinsically, through the implementation of LEbc, the coordinates of particle in the computation domain are transformed to be the relative coordinate with respect to different frame centres. Taking particle j' in Figure 1 as an example, at this instance, particle j' has crossed the upper boundary, and it is reintroduced into the computation domain as j. With no doubt, the coordinate of j' is in respect to the origin of the computation domain (here, we define the particle coordinate that is in respect to the computation domain as the *global coordinate*). Meanwhile, the relative position of j with respect to the centre of computation domain is equilibrium to the relative position of $j'$with respect to the upper frame centre. Therefore, the coordinate values of j in the computation domain are actually the coordinate values of j' with respect to the upper frame centre. In fact, the real coordinate system in LEbc is more complicated when the periodic boundary condition along the x-direction is also taken into account. However, considering interacting force between polymer beads, it is expected that the coordination of all the beads should be in the same reference frame.

In this paper, in order to overcome this problem, we first calculate the *global coordinates* of all the beads. After then, the interacting force of the polymer beads can be calculated in the same coordinate system. The procedure of this is straightforward, and the following two steps help us to derive the *global coordinate*.

(1) The centre coordinate of the relative frame in which the polymer bead is located is calculated. One more vector associated with every bead particle is employed to record the number of times that the bead crosses the boundary. We use $\mathbf{n}_{cc} = (n^{(x)}_{cc}, n^{(y)}_{cc})$ to represent this vector in a two-dimensional simulation. With this vector, the coordinate centre of the relative frame can be derived as

$$\begin{aligned} x^{(g)}_c &= n^{(x)}_{cc}L + n^{(y)}_{cc}\dot{\gamma}Wt, \\ y^{(g)}_c &= n^{(x)}_{cc}W. \end{aligned} \tag{15}$$

Here, $x^{(g)}_c$ and $y^{(g)}_c$ represent the *global coordinate* of the relative frame centre. The first term on the right-hand side represents the contribution from the normal periodic boundary condition and the second term represents the contribution from LEbc.

(2) With the *global coordinate* of the relative frame centre, the *global coordinate* of every single bead can be calculated easily as

$$\begin{aligned} x^{(g)}_i &= x_i + x^{(g)}_c, \\ y^{(g)}_i &= y_i + y^{(g)}_c. \end{aligned} \tag{16}$$

4. Numerical tests

In this section, a set of numerical tests concerning the issues mentioned above is demonstrated. The Newtonian fluid in shear flow and Newtonian fluid with polymer suspension are simulated consequently. The DPD parameters are chosen as shown in Table 1. In our simulations, a large repulsive parameter and number density and a small system temperature are used to yield a highly incompressible fluid.[22,23]

Table 1. DPD parameters and properties in the present study.

Description	Symbol	Value
Repulsive parameter	a^{FF}	59.00
Dissipative parameter	γ	4.5
Viscosity	η_f	8.26
System temperature	$k_B T$	0.5
Number density	n	6.0
DPD particle mass	m	1.0
Shear rate	$\dot{\gamma}$	0.01–0.5

4.1 Simulation of Newtonian fluid

The Newtonian fluid in shear flow is simulated in this section. In order to obtain the local variables including velocity vectors and stress tensor, we divide the whole domain into bins and the flow properties are calculated by averaging over all sampled data in each bin. Figure 3 gives the profiles of velocity and system temperature along the shear rate direction (y-direction). For comparison, results by the normal LEbc (two imaginary layers) approach are also shown. Both approaches yield uniform temperature distribution and linear velocity profile. However, a close inspection (black circle in Figure 3) indicates that, in the results of the normal approach, distortions of temperature distribution and velocity profile occur at the region close to the upper and bottom boundaries, where the conservation of momentum is broken. Chatterjee [3] also found this phenomenon in his simulation with the normal approach, especially with a large dissipative force. In order to fix this problem, in his simulation, the dissipative and random interactions which cross the boundary are manually neglected.

The modified approach in this paper gives a good result at the near boundary region. Meanwhile, the velocity field and temperature contour are shown in Figure 4. The centre plane ($z = 0$) is picked out for illustration. Linear velocity profile and uniform temperature distribution around 0.48–0.52 are obtained. Moreover, results with high dissipative parameter, $\gamma = 50$ and $\gamma = 100$, are also given in Figures 5 and 6. According to the previous work [3,20], a high dissipative parameter results in a high Schmidt number, leading to a high dynamic response fluid. The Schmidt number is a non-dimensional number accounting to the ratio of momentum diffusion of the fluid to particle diffusion of itself. Our results show that, for high dissipative parameter cases, current LEbc gives linear velocity profiles and uniform density distributions.

4.2 Simulation of suspension of elastic dumbbell polymer chains

In order to verify the implementation for calculating polymer chain force, the elastic dumbbell model is used as a test case, which is a simple model for dilute polymer solution, i.e. Rouse bead-spring model.[24] In this model, a polymer chain is discretized into several effective segments (bead and spring), each of which has a point mass undergoing some motion in the solvent. As the simplest model, there are only two beads in elastic dumbbell model and the spring force can be described as

$$\mathbf{F}^{(p)} = H\mathbf{R}, \tag{17}$$

where $\mathbf{R}$ is the end-to-end vector of the dumbbell and H is the spring constant. In our DPD simulation, each bead is

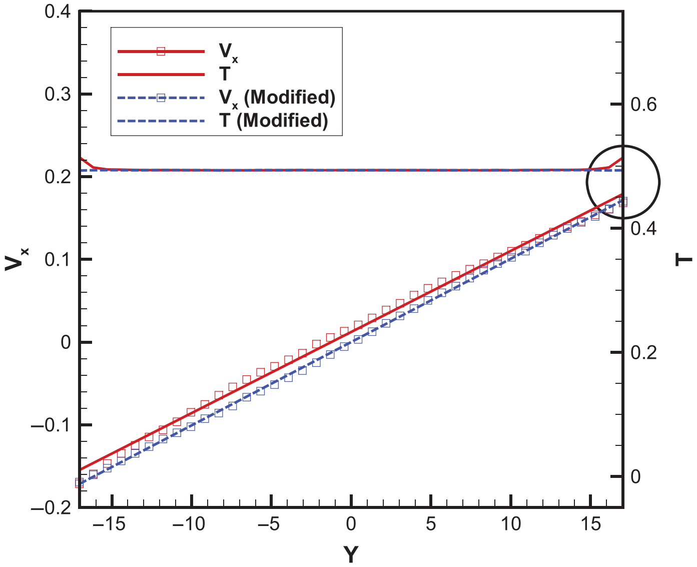


Figure 3. (Colour online) Profiles of velocity and temperature along the shear rate direction.

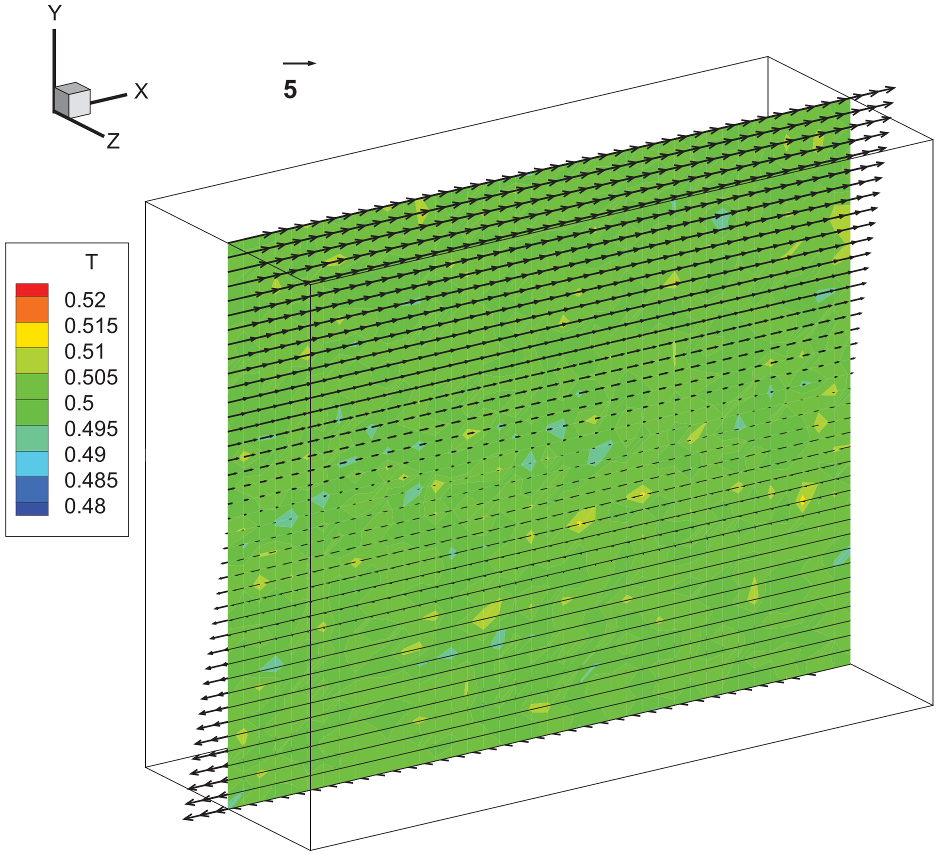


Figure 4. (Colour online) Velocity vectors and temperature contours in the plane of $z = 0$.

considered as a single DPD particle undergoing an additional spring force as described in Equation (17) and the LEbc implementation is following Equations (15) and (16). It is straightforward to extend the application of the current model to the complicated bead-spring polymer model, e.g. finitely extendable nonlinear elastic chain model and star-like polymer model. All the particles, including DPD particles and polymer beads, are in the

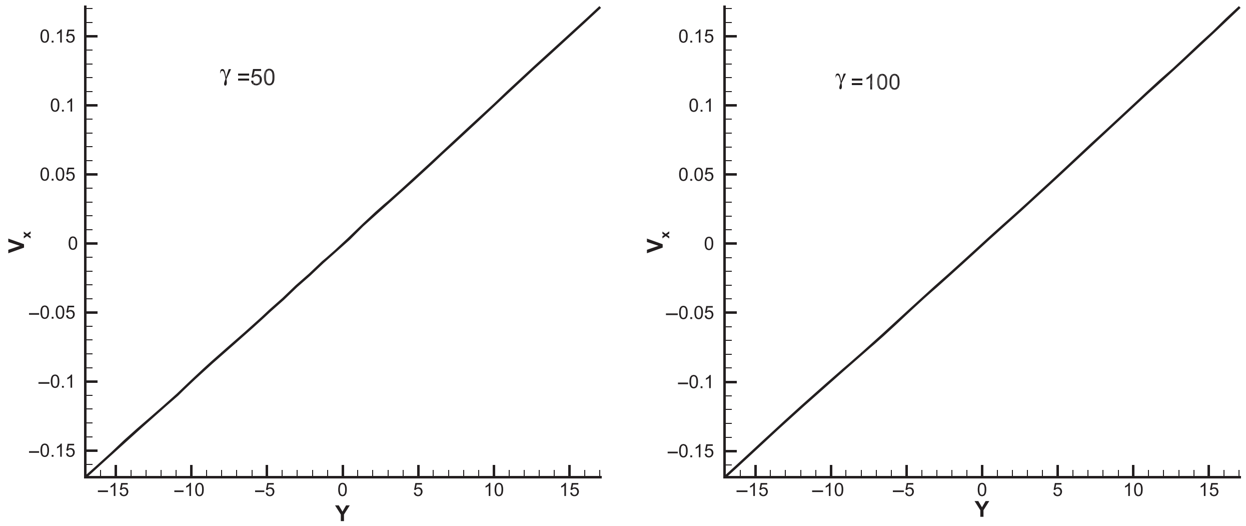


Figure 5. Linear velocity profiles with modified LE boundary condition at high Schmidt number (high dissipative parameter) cases with shear rate $\dot{\gamma} = 0.01$, left: dissipative parameter $\gamma = 50$; right: $\gamma = 100$.

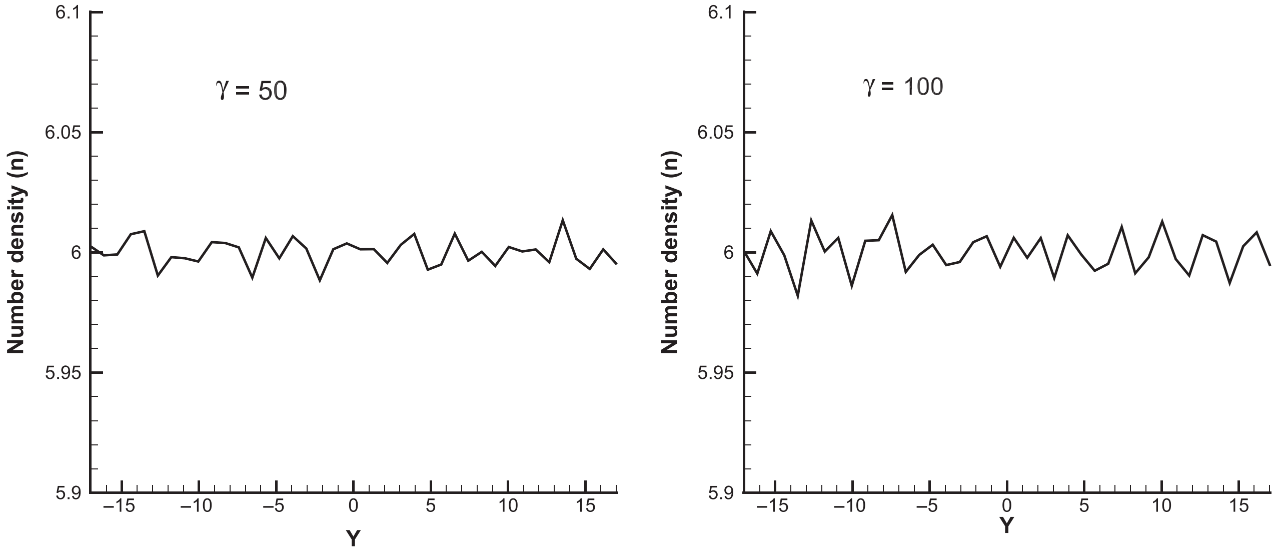


Figure 6. Number density profiles with modified LE boundary condition at high Schmidt number (high dissipative parameter) cases with shear rate $\dot{\gamma} = 0.01$, left: dissipative parameter $\gamma = 50$; right: $\gamma = 100$.

same framework, and the improved LE boundary condition can be applied to all the particles without any additional treatment. Concerning the computational efficiency, once the bead-spring polymer model is employed in simulation, only the computational cost used to calculate the spring forces is required to be taken into account.

On the other hand, theoretical work shows that the properties of this so-called ‘Boger fluid’ can be well presented by the Oldroyd-B model. The combination of the solvent- and polymer-contributed stresses can be written as

$$\mathbf{S} = \mathbf{S}^{(s)} + \tau^{(p)} = 2\eta_s\mathbf{D} + \tau^{(p)}. \tag{18}$$

Meanwhile, the polymer-contributed stress can be written as the familiar upper-convected Maxwell mode:

$$\tau^{(p)} + \lambda\frac{\delta}{\delta t}\tau^{(p)} = 2\eta_p\mathbf{D}. \tag{19}$$

Thus, we have

$$\mathbf{S} + \lambda_1\frac{\delta\mathbf{S}}{\delta t} = 2\eta\left(\mathbf{D} + \lambda_2\frac{\delta\mathbf{D}}{\delta t}\right), \tag{20}$$

where $\lambda_1 = \lambda$ is the relaxation time, $\eta = \eta_s + \eta_p$ is the total viscosity where η_s and η_p are the solvent viscosity and viscosity contributed by polymer chains, respectively. $\lambda_2 = \lambda\eta_s/\eta$ is the retardation time. In addition, $\delta/\delta t$ denotes the upper-convicted time derivative.

In a simple shear flow and stresses start from zero initial state, at steady state, the solution viscosity and the first and second normal stress differences can be calculated by

$$\begin{aligned} \eta &= \tfrac{S_{12}}{\dot{\gamma}} = \left(\eta_s + \eta_p\right), \\ N_1 &= \tau_{11}^{(p)} - \tau_{22}^{(p)} = 2\eta_p\lambda\dot{\gamma}^2, \\ N_2 &= \tau_{22}^{(p)} - \tau_{33}^{(p)} = 0. \end{aligned} \tag{21}$$

The DPD simulation results are shown in Figure 7. Both the DPD fluid particles and dumbbells are distributed along the shear rate direction (y-direction) uniformly, indicating that our simulation has reached the steady state. The system of this dumbbell suspension is kept at an equilibrium state with the system temperature very close to initial setup, $k_BT = 0.5$, and the temperature increase at the LEbc is not found. Positive first normal stress difference and zero second normal stress difference are also obtained which is in coincidence with the theoretical results (Equation (21)). Fluctuation of N_1 is observed, and it is probably related to the random motion of dumbbell particles.

In our simulation, the volume fraction of dumbbell particle is in the range of 2.5–10%, and shear rate varies from 0.01 to 0.5. Figure 8 gives the relation of N_1 and shear rate $\dot{\gamma}$ for different cases. The dash line with slope equal to 2.0 is also plotted in this figure as a reference. Linear fittings in this log − log coordinate are also made, and the slopes of all the case are in the range of 1.8–2.1. Thus, the parabolic relationship between N_1 and $\dot{\gamma}$ is almost satisfied in our DPD simulation.

Furthermore, in order to investigate the effect of spring constant, we varied the spring constant as $H = 0.05 - 0.5$. According to Equation (21), the relaxation time λ can be

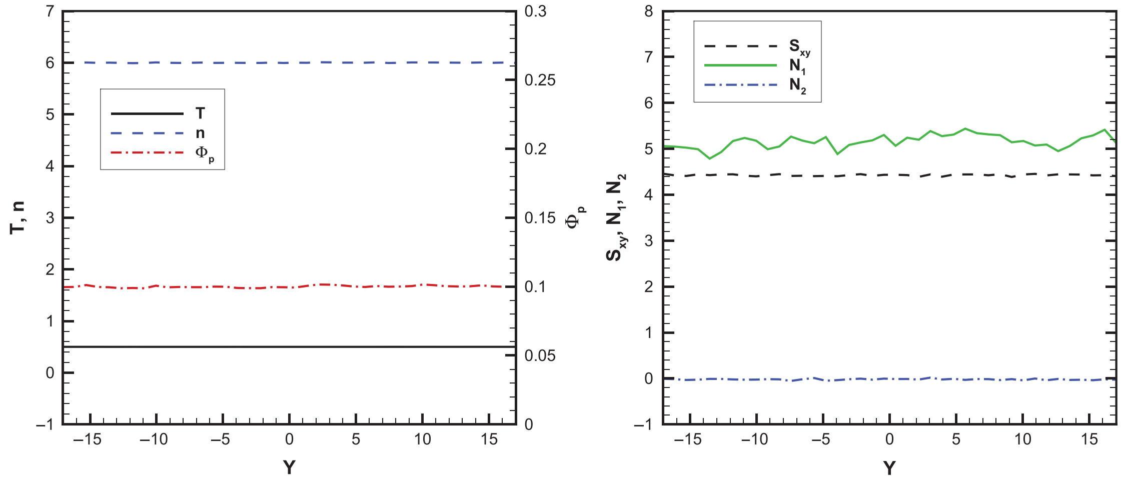


Figure 7. (Colour online) Distributions of fluid properties along the shear rate direction with volume fraction of dumbbell at 10%. Left: distributions of system temperature, number density and volume fraction of dumbbell; right: distributions of shear stress, first and second normal stress differences.

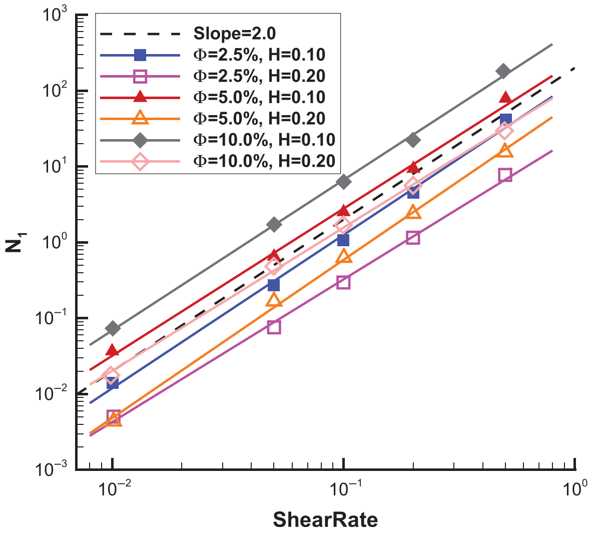


Figure 8. (Colour online) Variations of first normal stress difference (N_1) versus shear rate ($\dot{\gamma}$).

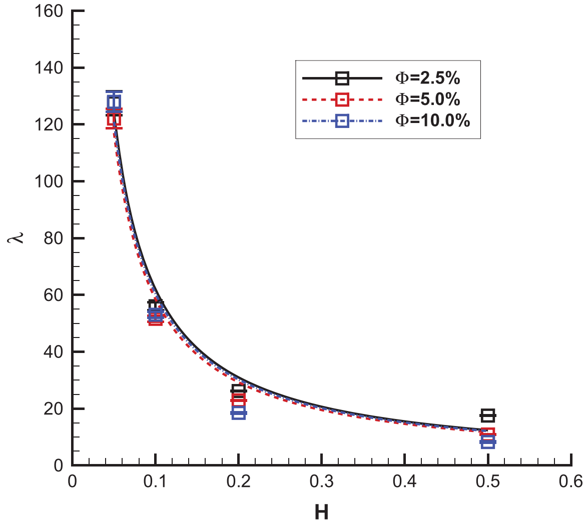


Figure 9. (Colour online) Variation of relaxation time (λ) versus spring constant (H).

obtained from N_1. On the hand, the Rouse relaxation time is normally defined as

$$\lambda = f(H^{-1}) = \frac{\zeta}{4H}, \tag{22}$$

where ζ is the frictional coefficient. Our DPD simulation results are given in Figure 9, in which the symbols represent the data derived from Equation (21), and the curves represent the fitting according to the relationship of Equation (22). From the curve fitting, results of the frictional coefficient are also given in Table 2. It is shown that the trend of λ follows the relation described in Equation (22) yielding a nearly constant frictional coefficient with magnitude around 6.0. Therefore, current elastic dumbbell model in DPD simulation can well capture the fluid properties, and hence represent the Oldroyd-B model. Meanwhile, the LEbc applied in this simulation also performs well to generate the simple shear flow.

5. Concluding remarks

The LEbc has been widely used to generate simple shear flow with certain advantages, e.g. it can reduce the finite size effect and eliminate the density fluctuation near the solid wall. The DPD method is conceived as an efficient numerical method in simulating polymer solution. Traditional implementation of LEbc in DPD simulation may cause momentum conservation to be destroyed and

Table 2. DPD simulation results of the frictional coefficient ζ.

Curves (%)	$\zeta/4$
$\Phi = 2.5$	6.19 ± 0.25
$\Phi = 5.0$	5.85 ± 0.27
$\Phi = 10.0$	6.05 ± 0.40

incorrect calculation of spring force within the polymer chain.

Special treatment for DPD particle interactions cross the sliding boundary is introduced in this paper, i.e. the random force between two DPD particles is only calculated once for interaction cross the sliding boundary. The momentum conservation is well satisfied through this application, and the temperature jump and distortion of velocity profile are avoided.

A *global coordinate* system is introduced for spring force calculation in polymer chain. The relative distance between two neighbour beads can be derived directly within this *global coordinate* system. The elastic dumbbell model is used to simulate dilute polymer solution. Our simulation results show that current DPD method combined with LEbc can well predict the fluid properties.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work is supported by the National Natural Science Foundation of China [grant number 11402230] and the Department of Education of Zhejiang Province, China [grant number Y201430480].

References

[1] Lees AW, Edwards SF. The computer study of transport processes under extreme conditions. J Phys C Solid State Phys. 1972;5:1921–1928. doi:10.1088/0022-3719/5/15/006.

[2] Boek ES, Coveney PV, Lekkerkerker HNW. Computer simulation of rheological phenomena in dense colloidal suspensions with dissipative particle dynamics. J Phys Condens Matter. 1996;8:9509–9512. doi:10.1088/0953-8984/8/47/053.

[3] Chatterjee A. Modification to Lees–Edwards periodic boundary condition for dissipative particle dynamics simulation with high dissipation rates. Mol Simul. 2007;33:1233–1236. doi:10.1080/08927020701713894.

[4] Chen S, Jin Y, Zhang M, Shang Z. Imposing Lees–Edwards boundary conditions in dissipative particle dynamics. J Tongji Univ (Nat Sci). 2012;40:137–142.

[5] Wagner AJ, Pagonabarraga I. Lees–Edwards boundary conditions for lattice Boltzmann. J Stat Phys. 2002;107:521–537. doi:10.1023/A:1014595628808.

[6] Hwang WR, Hulsen MA, Meijer HE. Direct simulation of particle suspensions in sliding bi-periodic frames. J Comput Phys. 2004;194:742–772. doi:10.1016/j.jcp.2003.09.023.

[7] Kobayashi H, Yamamoto R. Implementation of Lees–Edwards periodic boundary conditions for direct numerical simulations of particle dispersions under shear flow. J Chem Phys. 2011;134:064110. doi:10.1063/1.3537974.

[8] Hoogerbrugge PJ, Koelman JMVA. Simulating microscopic hydrodynamic phenomena with dissipative particle dynamics. Europhys Lett. 1992;19:155–160. doi:10.1209/0295-5075/19/3/001.

[9] Español P, Warren P. Statistical-mechanics of dissipative particle dynamics. Europhys Lett. 1995;30:191–196.

[10] Marsh C. Theoretical aspects of dissipative particle dynamics [D Phil thesis]. University of Oxford 1998.

[11] Pan WX, Caswell B, Karniadakis GE. Rheology, microstructure and migration in Brownian colloidal suspensions. Langmuir. 2010;26:133–142. doi:10.1021/la902205x.

[12] Bian X, Litvinov S, Qian R, Ellero M, Adams NA. Multiscale modeling of particle in suspension with smoothed dissipative particle dynamics. Phys Fluids. 2012;24:012002. doi:10.1063/1.3676244.

[13] Clark AT, Lal M, Ruddock JN, Warren PB. Mesoscopic simulation of drops in gravitational and shear fields. Langmuir. 2000;16:6342–6350. doi:10.1021/la991565f.

[14] Pan D, Phan-Thien N, Khoo BC. Dissipative particle dynamics simulation of droplet suspension in shear flow at low Capillary number. J Non-Newtonian Fluid Mech. 2014;212:63–72. doi:10.1016/j.jnnfm.2014.08.011.

[15] Fan XJ, Phan-Thien N, Chen S, Wu XH, Yong Ng T. Simulating flow of DNA suspension using dissipative particle dynamics. Phys Fluids. 2006;18:063102. doi:10.1063/1.2206595.

[16] Fedosov DA, Pan W, Caswell B, Gompper G, Karniadakis GE. Predicting human blood viscosity in silico. Proc Natl Acad Sci USA. 2011;108:11772–11777. doi:10.1073/pnas.1101210108.

[17] Pivkin IV, Karniadakis GE. Controlling density fluctuations in wall-bounded dissipative particle dynamics systems. Phys Rev Lett. 2006;96:206001. doi:10.1103/PhysRevLett.96.206001.

[18] Henrich B, Cupelli C, Moseler M, Santer M. An adhesive DPD wall model for dynamic wetting. EPL. 2007;80:60004. doi:10.1209/0295-5075/80/60004.

[19] Afshar Y, Schmid F, Pishevar A, Worley S. Exploiting seeding of random number generators for efficient domain decomposition parallelization of dissipative particle dynamics. Comput Phys Commun. 2013;184:1119–1128. doi:10.1016/j.cpc.2012.12.003.

[20] Groot RD, Warren PB. Dissipative particle dynamics: bridging the gap between atomistic and mesoscopic simulation. J Chem Phys. 1997;107:4423–4435. doi:10.1063/1.474784.

[21] Irving JH, Kirkwood JG. The statistical mechanical theory of transport processes. IV. The equations of hydrodynamics. J Chem Phys. 1950;18:817–829. doi:10.1063/1.1747782.

[22] Pan D, Phan-Thien N, Mai-Duy N, Khoo BC. Numerical investigations on the compressibility of a DPD fluid. J Comput Phys. 2013;24:2196–2210.

[23] Pan D, Phan-Thien N, Khoo BC. Studies on liquid–liquid interfacial tension with standard dissipative particle dynamics method. Mol Simul. 2014. doi:10.1080/08927022.2014.952636.

[24] Phan-Thien N. Understanding viscoelasticity: An introduction to rheology, graduate texts in physics. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-642-32957-9 2013.