Dissipative Particle Dynamics

Igor V. Pivkin, Bruce Caswell, and George Em Karniadakis

a Division of Applied Mathematics, Brown University, Providence, Rhode Island
b Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts

INTRODUCTION

Molecular modeling of liquid-state systems based on atomistic simulations remains computationally prohibitive for mesoscopic spatial domains and lengthy time scales despite the teraflop speeds of current parallel computer systems. Examples of such liquid-state systems being thwarted by this include polymer solutions and melts, multicomponent mixtures, amphiphilic systems, fluid vesicles, and cell suspensions. To obviate these problems, many new methods focusing on coarse-graining approaches have been proposed in the last few years that yield low-dimensional systems amenable to fast computations in simulation studies of simple and even complex systems such as biomolecular liquids. Typical coarse-graining approaches include the elimination of fast degrees of freedom, clustering of individual atoms into larger single-interaction particles, and spatio-temporal averaging of effective interaction potentials. Other recent coarse-graining approaches based on stochastic closures or approximate
inertial manifold ideas that could potentially be used for liquid-state systems can be found in references 2 and 3, respectively.

In this chapter we will consider dissipative particle dynamics (DPD), a relatively new mesoscopic method. Beginning from the basic ideas behind the DPD method, we next describe extensions of the DPD method that have been proposed in recent years. This is followed by some examples of the DPD method applied to mesoscale problems.

FUNDAMENTALS OF DPD

DPD is a coarse-graining method that employs both simplified potentials as well as grouping of atoms into a single particle (i.e., the DPD particle). The method describes interacting clusters of molecules, moving together in a Lagrangian fashion, subject to soft repulsive-only potentials. Specifically, for simple fluids, there are three types of forces acting on each dissipative particle: (1) purely repulsive conservative force, (2) dissipative force that reduces velocity differences between the particles, and (3) stochastic force directed along the lines connecting the particle centers.

In this section, we first present the mathematical formulation of the DPD method and then discuss units in DPD simulations. This is followed by the discussion of the DPD thermostats and integration schemes used in computer simulations. The section concludes with the discussion of boundary conditions in DPD.

Mathematical Formulation

Let us consider a system consisted of \( N \) particles having equal mass (for simplicity of presentation) \( M \), positions \( \mathbf{r}_i \), and velocities \( \mathbf{v}_i \). The aforementioned three types of forces exerted on a particle \( i \) by particle \( j \) are given by

\[
\mathbf{F}^C_{j} = F^C(r_{\bar{j}})\hat{r}_{\bar{j}}
\]

\[
\mathbf{F}^D_{j} = -\gamma \omega^D(r_{\bar{j}})(\hat{r}_{\bar{j}} \cdot \mathbf{v}_{\bar{j}})\hat{r}_{\bar{j}}
\]

\[
\mathbf{F}^R_{j} = \sigma \omega^R(r_{\bar{j}})\xi_{\bar{j}}\hat{r}_{\bar{j}}
\]

where \( r_{\bar{j}} = r_i - r_j \), \( r_{\bar{j}} = |r_{\bar{j}}| \), \( \hat{r}_{\bar{j}} = r_{\bar{j}}/r_{\bar{j}} \), and \( \mathbf{v}_{\bar{j}} = \mathbf{v}_i - \mathbf{v}_j \). The variables \( \gamma \) and \( \sigma \) determine the strength of the dissipative and random forces, respectively. Also, \( \xi_{\bar{j}} \) are symmetric Gaussian random variables with zero mean and unit variance, and they are independent for different pairs of particles and at different times; \( \xi_{\bar{j}} = \xi_{\bar{j}} \) is enforced to satisfy momentum conservation. Finally, \( \omega^D \) and \( \omega^R \) are weight functions.
All forces act within a sphere of interaction radius $r_c$, which is the length scale of the system. The conservative force is derived from a soft potential (the potential is designated as “soft” to distinguish it from the hard core of molecular potentials such as that described later in Eq. [10]):

$$F^C_{ij} = \begin{cases} 
  a_{ij}(1 - r_{ij}/r_c)\hat{r}_{ij}, & r_{ij} < r_c \\
  0, & r_{ij} \geq r_c
\end{cases} \quad [4]$$

$$a_{ij} = \sqrt{a_i a_j}, \quad \text{where } a_i \text{ and } a_j \text{ are conservative force coefficients for particle } i \text{ and particle } j.$$ 

The requirement of the canonical distribution sets two conditions on the weight functions and the amplitudes of the dissipative and random forces, (see Refs. 5 and 6). Specifically, we have that

$$\omega^D(r_{ij}) = \left[\omega^R(r_{ij})\right]^2 \quad [5]$$

and

$$\sigma^2 = 2\gamma k_B T \quad [6]$$

where $T$ is the system temperature and $k_B$ is the Boltzmann constant. The weight function in standard DPD takes the form

$$\omega^R(r_{ij}) = \begin{cases} 
  1 - r_{ij}/r_c & \text{for } r_{ij} \leq r_c \\
  0 & \text{for } r_{ij} > r_c
\end{cases} \quad [7]$$

The time evolution of DPD particles is described by Newton’s law

$$d\mathbf{r}_i = \mathbf{v}_i dt \quad [8]$$

$$d\mathbf{v}_i = \frac{1}{M} \left( F^C_i dt + F^D_i dt + F^R_i \sqrt{dt} \right) \quad [9]$$

Here $F^C_i = \sum_{j \neq i} F^C_{ij}$ is the total conservative force acting on particle $i$; $F^D_i$ and $F^R_i$ are defined similarly. The $\sqrt{dt}$ term multiplying random force $F^R_i$ in Eq. [9] is there to ensure that the diffusion coefficient of the particles is independent of the value of the time step used in simulations. This point is discussed in more detail in Ref. 6.
Units in DPD

A Lennard–Jones 12-6 potential,

\[ u(r) = 4\varepsilon \left[ \left( \frac{\sigma_{LJ}}{r} \right)^{12} - \left( \frac{\sigma_{LJ}}{r} \right)^{6} \right] \]  

[10]

is widely used in molecular dynamics simulations. It provides a reasonable description of the properties of spherically symmetric molecules such as argon, if the parameters \( \varepsilon \) and \( \sigma_{LJ} \) (not to be confused with parameter \( \sigma \) in DPD formulation) are specified appropriately. For this potential, the magnitude of the force between the particles increases to infinity as the separation distance \( r \) approaches zero. This hard core imposes severe constraints on the maximum time step that can be used to integrate the equations of motion numerically.

The major advantage of the DPD method is its soft conservative force potential Eq. [1], reflecting its mesoscale nature. The soft potentials similar to the one used in DPD were systematically calculated in Ref. 7. To support the use of soft potentials, we consider a fluid consisting of particles interacting through hard Lennard–Jones potentials. We group collections of particles into clusters as depicted in Figure 1. If we measure the effective potentials between these clusters, we find that as we make the clusters larger, and as we average the interactions between them over longer time intervals, the effective potentials between clusters become softer approaching the potentials used in DPD. The force between the particles is no longer infinite at zero separation, which is reasonable if we think about the particles as being packets of fluid whose centers can overlap as the packets move through each other.

In contrast to a molecular dynamics (MD) simulation where the choice of potential is based on a theoretical-molecular model of the physical system being simulated, a DPD simulation, in principle, involves potentials of a form that is independent of the physical system. The DPD potentials do, however, include parameters that must be chosen properly to provide an accurate approximation of the system. Also, an MD simulation contains a set of units that is intimately related to the theoretical model. Because the potentials in DPD are not given by

![Figure 1](MD.png)  

**Figure 1** DPD particles represent clusters of atoms of molecules. They interact with each other through soft pairwise forces.
a physical model, the relation of natural DPD length and time scales to physical units needs to be established.

The DPD scales can be linked to the macroscopic or molecular scales depending on the problem of interest. Linking to the macroscopic scales can be done by specifying the DPD simulation parameters so that relevant non-dimensional numbers (such as Reynolds number in flow simulations) or specific fluid properties are preserved.\textsuperscript{8–12} Other properties, such as dimensionless compressibility of the fluid, are typically not preserved in this approach.

Several different approaches for linking molecular with mesoscopic scales have been proposed as described, for example, in Refs. 6, 13, and 14. In these simulations, each DPD particle typically corresponds to a number of atoms or molecules; the number of atoms or molecules per DPD particle is known as the coarse-graining parameter and is denoted by \( N_m \). The link between the molecular (MD) and mesoscale approach (DPD) is established by determining a thermodynamic property, such as the compressibility modulus, from MD simulations and adjusting the parameters in the DPD model accordingly. Groot and Warren\textsuperscript{6} showed, through a series of equilibrium simulations with different values of DPD conservative force coefficient \( a_{ij} \) and density \( \rho \), that for sufficiently large number densities (\( \rho > 2 \)), the DPD equation of state, to a good approximation, is given by

\[
p = \rho k_B T + \alpha a_{ij} \rho^2 k_B T \tag{11}
\]

where \( p \) is pressure and \( \alpha \) was determined to be \( 0.101 \pm 0.001 \). The isothermal compressibility is by definition \( \kappa^{-1} = 1/k_B T (\partial p/\partial n)_T \). Here \( n \) is the number density of actual atoms or molecules in the physical system which, in units of \( r_c^{-3} \), is \( n = N_m \rho \). Following Ref. 15, we find

\[
\kappa^{-1} = \frac{1}{k_B T} \left( \frac{\partial p}{\partial \rho} \right)_T \left( \frac{\partial \rho}{\partial n} \right)_T = \frac{1}{N_m} (1 + 2\alpha \rho) \tag{12}
\]

Equation [12] provides the necessary relationship between the mesoscopic model parameter and the compressibility of the system. Therefore, with \( \kappa^{-1} \) determined by the physical system, the proper value of the conservative force coefficient can be found from

\[
a_{ij} = \frac{\kappa^{-1} N_m - 1}{2\alpha \rho} \tag{13}
\]

The conservative force increases with the number of atoms per DPD particle resulting in some coarse-graining artifacts, such as spontaneous solidification of the DPD liquid.\textsuperscript{16} Therefore, this approach for defining simulation parameters is limited to relatively low levels of coarse-graining.
In Refs. 15 and 17 the authors matched the diffusion constant of the DPD simulation to that of the molecule of interest to determine the time scale. The self-diffusion constant of water was considered in Ref. 15, whereas the diffusion constant of a surfactant micelle was used in Ref. 17. Keaveny et al.\textsuperscript{13} applied shear to the fluid, and the time scales were related through the diffusion of vorticity. As such, the DPD time scale was determined by matching the kinematic viscosities of the DPD and the physical systems.

The mapping of the DPD temperature to the physical temperature was established by Venturoli, Smit, and Sperotto\textsuperscript{18} using the bilayer phase transition in the simulations of lipid membranes.

**Thermostat and Schmidt Number**

DPD differs from MD in two main respects. First, as was discussed, the soft conservative forces between the DPD particles allow for simulations at longer time scales. Second, the DPD thermostat for the canonical ensemble implemented in terms of dissipative and random forces conserves momentum locally and allows the emergence of hydrodynamic effects in relatively small systems.

In the initial formulation of DPD by Hoogerbrugge and Koelman\textsuperscript{4} the conditions on dissipative and random forces, Eqs. [5] and [6], were not satisfied and energy was not conserved. This was corrected in Ref. 5 in conjunction with the solution of the Fokker–Planck equation, the basis of Brownian dynamics. Let us consider the distribution function \( f(r_i, p_i, t) \), which describes the probability of finding the system in a state with particles located at \( r_i \) having momenta \( p_i \) at time \( t \). The time evolution of this distribution is expressed by the Fokker–Planck equation, which represents the conservation of particles in phase space,

\[
\frac{\partial f}{\partial t} = \mathcal{L}^C f + \partial L^D f
\]

where \( \mathcal{L}^C \) denotes the Liouville operator of the Hamiltonian system interacting with conservative forces \( F^C \); also, \( \partial L^D \) represents the dissipative and random terms. If the latter two are set to zero, we obtain a Hamiltonian system, which admits the canonical Gibbs–Boltzmann distribution as a solution. That is, \( f^{eq}(r_i, p_i) = \exp\left(-\sum_i p_i^2 / 2mk_BT - U / k_BT\right) \) is a solution of

\[
\frac{\partial f^{eq}}{\partial t} = \mathcal{L}^C f^{eq} = 0
\]

However, in the presence of the extra two nonconservative forces, the equilibrium distribution will be different than the above unless

\[
\mathcal{L}^D f^{eq} = 0
\]
This condition is satisfied if the amplitude of the random and dissipative forces and the weight functions are related as presented earlier in Eqs. [5] and [6].

In practice, the random force coefficient \( \sigma \) and the dissipative force coefficient \( \gamma \) are chosen to yield an efficient and numerically stable DPD simulation that satisfies Eq. [6]. As the random force increases, the speed at which the system reacts to temperature variations increases, leading to efficient temperature equilibration. There is, however, an upper limit to the choice of \( \sigma \). This upper limit is based on the time integration scheme, time step, and temperature.

As it was pointed out in Ref. 6, the original DPD thermostat produces a gas-like Schmidt number, \( Sc \sim O(1) \). The Schmidt number is the ratio of the kinematic viscosity to the diffusion coefficient. In a fluid flow, momentum can be transported rapidly by the interparticle potentials, whereas mass transport occurs by the displacement of particles and is relatively slow. Therefore, \( Sc \) is a large number for fluids. For water, it is of the order \( 10^3 \). The Schmidt number characterizes the dynamic behavior of fluids and is an important parameter to consider in simulating unsteady and complex flows. The effect of the Schmidt number in DPD simulations of dilute polymeric systems was investigated in Ref. 12, where it was shown that \( Sc \) of the solvent strongly affects nonequilibrium polymeric quantities. The Schmidt number can be varied in the DPD method. For example, Fan et al. modified the weight functions for the dissipative and random forces by introducing a generalized function:

\[
\omega_D(r) = \left[ \omega_R(r) \right]^2 = \begin{cases} 
(1 - r/\rho) \gamma & \text{for } r \leq \rho \\
0 & \text{for } r > \rho 
\end{cases}
\]

with exponent \( \gamma \leq 2 (\gamma = 2 \text{ for standard DPD}) \). For a DPD system with \( \gamma = 1/2 \), the Schmidt number was estimated to be approximately 35 times larger than that for standard DPD. The \( Sc \) can be affected greatly by changing the cut-off radius \( \rho \) because \( Sc \) is proportional to \( \rho^8 \). It was found in Ref. 19 that for \( \gamma = 1/2 \) and \( \rho = 1.881 \), the Schmidt number is about \( 10^3 \).

An alternative way to achieve large Schmidt numbers in simulations is to use Lowe’s thermostat. Lowe’s method employs the Andersen thermostat with the particle velocities corrected at every time step using the Maxwell velocity distribution. In the absence of conservative forces, which are integrated in the velocity-Verlet manner, the scheme is shown to be independent of the time step, although recent work has shown that the scheme can give rise to some minor artifacts. The core operation in Lowe’s method involves re-equilibration of the particle momenta at one step with an updated interparticle relative velocity drawn from a Gaussian distribution. The scheme is characterized by the explicit calculation of \( F^c \) and the subsequent re-equilibration of all particle velocities. This is done using the relative velocities of particle pairs. The method conserves momentum and introduces an extra parameter \( \Gamma \) so that in the limiting case of \( \Gamma \times \Delta t \approx 1 \) thermalization/dissipation occurs at every
time step for all neighbors of the particular DPD particle examined. Peters\textsuperscript{24} recently introduced a modification of Lowe’s scheme by keeping the centroid velocity of a particle-pair unchanged before and after the re-equilibration. This results in an attractive scheme, still independent of the chosen time step (as opposed to the Verlet approach), that also discretizes the original DPD equations (Lowe’s method does not). An overview of the original Lowe scheme is given in Ref. 22. The Lowe–Anderson thermostat was applied recently in molecular dynamics simulations by Koopman and Lowe.\textsuperscript{25} It was also combined with a Nosé–Hoover thermostat in an attempt to construct a local, Galilean invariant, Nosé–Hoover thermostat.\textsuperscript{26}

**Integration Algorithms**

In computer simulations, the trajectories of DPD particles given by Eqs. [8] and [9] are calculated using numerical integration. One of the simplest integration methods is an Euler scheme, in which the new positions and velocities are derived from the positions and velocities at a previous time step

\[
\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t) \tag{18}
\]

\[
\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \Delta t \mathbf{F}_i(t) \tag{19}
\]

\[
\mathbf{F}_i(t + \Delta t) = \mathbf{F}_i(\mathbf{r}_i(t + \Delta t), \mathbf{v}_i(t + \Delta t)) \tag{20}
\]

This method, however, is avoided even in MD simulations, as it is known to produce a drift in the energy of the system and provide particle trajectories that are not time reversible. The energy drift may be reduced in DPD due to the embedded thermostat; however, the problem with the time reversibility results in simulation artifacts.

The time reversibility in MD simulations can be recovered by using a Verlet-type algorithm\textsuperscript{27} where the positions of the particles are updated as

\[
\mathbf{r}_i(t + \Delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t - \Delta t) + \frac{1}{M}(\Delta t)^2 \mathbf{F}_i(t) \tag{21}
\]

\[
\mathbf{F}_i(t + \Delta t) = \mathbf{F}_i(\mathbf{r}_i(t + \Delta t)) \tag{22}
\]

An equivalent velocity-Verlet algorithm,\textsuperscript{27,28} in which the particle velocities appear explicitly, is given by

\[
\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t) + \frac{1}{2}(\Delta t)^2 \frac{1}{M} \mathbf{F}_i(t) \tag{23}
\]

\[
\mathbf{F}_i(t + \Delta t) = \mathbf{F}_i(\mathbf{r}_i(t + \Delta t)) \tag{24}
\]
\[ v_i(t + \Delta t) = v_i(t) + \frac{1}{2} \Delta t \frac{1}{M} [F_i(t) + F_i(t + \Delta t)] \]  
\[ [25] \]

In DPD, however, the force between the particles depends on the positions and velocities of the particles, leading Groot and Warren\textsuperscript{6} to propose the modified velocity-Verlet algorithm:

\[ r_i(t + \Delta t) = r_i(t) + \Delta t v_i(t) + \frac{1}{2} (\Delta t)^2 \frac{1}{M} F_i(t) \]  
\[ [26] \]

\[ \tilde{v}_i(t + \Delta t) = v_i(t) + \frac{1}{M} F_i(t) \]  
\[ [27] \]

\[ F_i(t + \Delta t) = F_i(r_i(t + \Delta t), \tilde{v}_i(t + \Delta t)) \]  
\[ [28] \]

\[ v_i(t + \Delta t) = v_i(t) + \frac{1}{2} \Delta t \frac{1}{M} [F_i(t) + F_i(t + \Delta t)] \]  
\[ [29] \]

where the prediction for new velocity \( \tilde{v}_i(t + \Delta t) \) is made before the force calculation and is then corrected afterward. The variable \( \lambda \) is introduced in the algorithm to account for the effects of stochastic interactions.

The performance of the integration scheme in DPD can be evaluated by monitoring the temperature of the system, the radial distribution function (RDF), and other properties, such as the velocity autocorrelation function.\textsuperscript{29} Temperature control is a convenient monitoring method because the correct temperature at equilibrium is known from the fluctuation–dissipation relations of Eqs. [5] and [6]. The RDF is unknown in general, but for the ideal gas, it is equal to one in the continuum limit. Therefore, with conservative force equal to zero (i.e., \( a_{ij} \equiv 0 \)) deviations of the RDF from one characterize the artifacts due to the integration scheme employed.

The temperature drift and RDF artifacts associated with the velocity-Verlet method can be reduced by finding the optimal value of \( \lambda \) (which depends on simulation parameters and has to be determined empirically). However, optimizing \( \lambda \) does not eliminate the artifacts completely, and therefore, several improved integration schemes were proposed in the literature.\textsuperscript{22,30–36} Despite the absence of temperature drift and other artifacts in many of the proposed methods, the modified velocity-Verlet algorithm is still one of the most popular integration schemes in DPD.

The modified velocity-Verlet algorithm was extended for the simulation of complex fluids with soft/hard potentials by Symeonidis and Karniadakis.\textsuperscript{38} Figure 2\textsuperscript{37} shows polymeric chains moving freely in a DPD solvent of \( N \) particles. These chains consist of beads (DPD particles) subject to the standard DPD forces: soft repulsive (conservative), dissipative, and random. In addition to these forces, the polymers are subject to intrapolymer forces, arising from the combination of Lennard–Jones (LJ) forces, Hookean springs, Frenkel (stiff)
springs, finitely extensible, nonlinear elastic (FENE) springs, worm like chain (WLC) forces, and so on.\textsuperscript{39} The Lennard–Jones forces sometimes used in DPD polymer models for bead–bead pairs is a hard repulsion that requires a much smaller time step than the soft interaction forces of a typical DPD particle pair. This incongruence suggests the need for staggered integration schemes with two different temporal resolutions in polymer simulations. A large time step $\Delta t$ is employed for solvent particles, and a smaller one $\delta t$ is used for polymer particles belonging to a chain. The polymer particles are integrated $\Delta t$ times in a separate subcycle (using $\delta t$ for the time step). The intrapolymmer forces are updated during the subcycle, but the interparticle (total) forces are not. Updating the latter forces would require central processing unit (CPU) time for each subcycle equivalent to a standard DPD simulation. Hence, we cannot expect exact agreement of the new scheme with the classical one, but we can anticipate small differences if the ratio $\Delta t / \delta t$ is not too large and if the (outdated) forces are applied corrected during the $\delta t$ cycle.

**Boundary Conditions**

One of the main issues for DPD simulations in confined geometries is the imposition of boundary conditions, specifically at solid boundaries. To this
end, the boundary conditions that have been used in DPD are based on general ideas implemented both in the lattice Boltzmann method (LBM) and in the MD formulations. However, unlike the MD method, the soft repulsion between DPD particles cannot prevent fluid particles from penetrating solid boundaries, and thus, extra effort is required to impose accurately the no-slip (or partial slip) wall boundary condition.

A broad classification of the three main approaches used to impose boundary conditions in DPD provided by Revenga et al. is as follows:

1. Avoid direct modeling of the physical boundary by modifying the periodic boundary conditions. This method, which belongs to the first category, is the Lees–Edwards method. It is used by many researchers who use MD, DPD, LBM, and other simulation techniques. The method consists of a simple recipe that imposes periodic boundary conditions on particles leaving the simulation box in the directions perpendicular to the velocity gradient and allows for the simulation of planar shear flow.

2. Freeze regions of the fluid to create a rigid wall or a rigid body as, for example, in particulate flows.

3. Combine different types of particle layers with proper reflections, namely specular reflection, bounce-back reflection, or Maxwellian reflection.

The classical method, which belongs to the first category, is the Lees–Edwards method. It is used by many researchers who use MD, DPD, LBM, and other simulation techniques. The method consists of a simple recipe that imposes periodic boundary conditions on particles leaving the simulation box in the directions perpendicular to the velocity gradient and allows for the simulation of planar shear flow.

The periodic Poiseuille flow method developed by Backer et al. also falls into the first category. The method produces counter-flowing Poiseuille flows using periodic boundary conditions with the uniform density profile except the statistical fluctuations. The absence of density artifacts makes this method useful for studying the bulk Poiseuille flow (in the continuum limit) (i.e., without any density oscillations associated with the presence of solid boundaries). In addition, the method is commonly used to estimate the viscosity of the DPD fluid. We note that for estimating the viscosity of Newtonian fluids, one can alternatively use the Gosling method.

The second category includes methods that model solid walls and objects by grouping and “freezing” collections of DPD particles. The particles can be grouped in a specific way to reduce the roughness of the surface as in Ref. 44, or they can be grouped simply as a subset of the initial lattice. The motion of these DPD particles is constrained in simulations, so that their positions remain the same relative to one another. It was observed that the fluid particles can penetrate the solid objects constructed this way. To prevent particles from entering the wall region a higher particle density in the wall was used in Refs. 43 and 51, whereas in Ref. 52, the repulsive force from the wall particles was increased. Both these techniques result in a strong wall repulsion and a depletion of particles near the wall if the conservative force is nonzero ($a_{ij} \neq 0$).
The third category is broad in scope, and the technical details in the various implementations published so far are diverse. In Ref. 46 a particle layer is placed on the solid boundary and effective dissipative and random forces on the DPD fluid particles are obtained analytically by assuming a continuum limit. Even though the more difficult case (where conservative forces are present \( a_i \neq 0 \)) was not considered in Refs. 46, reflections were necessary to reinsert particles back into the fluid when they cross the wall because the effective computed forces alone were insufficient to prevent wall penetration.

In the work by Willemsen et al.,47 an extra particle layer was included outside of the domain with the objective of constructing a correct velocity profile that continues beyond the wall boundary. The position and velocities of particles inside that layer are determined from the layer of DPD particles adjacent to the boundary and within a distance \( r_c \). For example, to impose zero velocity at a solid boundary, points in the particle layer outside the domain have tangential and normal velocity components opposite from the original points within the boundary layer. When a DPD particle hits the boundary, a bounce-back reflection is imposed. This approach works well in the absence of conservative forces, but when conservative forces are present, density oscillations were found to occur in the near-wall region. To compensate for this, a second layer of DPD particles was introduced by Willemsen et al.47 between \( r_c \) and \( 2r_c \) in order to compute the repulsive interaction. This approach seems to reduce but not totally eliminate the density fluctuations. Overall, the method described in Ref. 47 is effective, but it may not be implemented easily in complex-geometry flows. In the work of Visser, Hoefsloot, and Ledema54 the extra layer of particles was eliminated by adding parallel twin systems that set up the wall by a back-to-back placement.

Another implementation of the no-slip boundary conditions by combining freezing of uniform layers of DPD particles and reflection at the solid–fluid interface was developed by Pivkin and Karniadakis.53 To achieve a no-slip condition at the wall with their method, the repulsive interactions from wall particles are adjusted so that the average force acting on the DPD particles from the wall in the near-wall region is equal to the average force from the fluid. Such tuning of the conservative force does not solve the problem of near-wall density fluctuations completely, but it does eliminate velocity slip for low and moderate shear rates in the standard DPD method.

In the work of Fan et al.,45 frozen particles are used to represent the wall, but an extra thin layer of DPD particles exists inside the domain and adjacent to the solid boundary where the no-slip boundary condition holds. A random velocity distribution with zero mean is enforced in this layer. Although that thin layer is needed to prevent the frozen wall from cooling the DPD fluid, some temperature drop at the wall boundaries is nevertheless present in the simulation results reported in Ref. 45, which is undesirable.

Finally, we also have included another implementation in the third category above. In this method, called adaptive boundary conditions,55 the region
Extensions of DPD 97

close to the wall is subdivided into bins, and the density is computed in each bin during the simulation. If the density is higher than a desired density, the repulsive force pushing fluid particles away from the wall inside the bin is increased, and likewise, if the density is lower than desired, the force is decreased. This adaptive procedure allows for any desired density profile to be obtained close to the wall (e.g., a flat profile for high levels of coarse-graining as well as profiles for lower levels of coarse-graining that include density fluctuations arising from the layering of atoms or molecules near the wall). The results obtained in simulations using this method were found to be in good agreement with continuum Navier–Stokes simulations and molecular dynamics simulations of a Lennard–Jones fluid.

EXTENSIONS OF DPD

Several extensions of the DPD method were proposed in recent years. In this section, we describe some of them briefly.

DPD with Energy Conservation

DPD could only be applied to isothermal systems in the original formulation because the energy is not conserved in the particle interactions. The original DPD system cannot sustain temperature gradients, and no heat flow can be modeled. A generalization of DPD to nonisothermal systems was developed by introducing particle internal energy, $\epsilon_i$. This concept was developed independently by Avalos and Mackie and by Español. The model is known as dissipative particle dynamics with energy conservation (DPDE), where for interacting particles, the mechanical energy dissipated by friction is transformed into internal energy. In addition, the temperature differences between particles produce variations in the internal energies through heat conduction. A local entropy function $s(\epsilon_i)$ is specified for each particle, so that the temperature $T_i$ can be defined.

The key aspect of DPDE is that the fluctuation–dissipation theorem relates the random force with the temperature $T_i$ rather than with the thermodynamic temperature $T$ as it is done in standard DPD. Because DPDE is defined in terms of particle properties only, with no reference to macroscopic parameters of the system, DPDE can be used in nonisothermal problems. Even though DPDE is potentially useful in a wide range of applications where thermal transport plays a critical role, it has received significantly less attention in the literature to date compared with the standard DPD method. We point out that in Ref. 61 an implicit algorithm was derived that conserves energy at every time step, rather than in the mean as was the case with the original DPDE formulation. Also, in Ref. 67, Peters and Lowe thermostats were extended to incorporate energy conservation.
Fluid Particle Model

The fluid particle model (FPM) was originally developed by Español.\textsuperscript{68} Compared with the standard dissipative particle dynamics (DPD) method,\textsuperscript{4,6} this model incorporates two additional, \textit{noncentral} shear components into the dissipative force. FPM can be considered as a generalization of the DPD method, which includes torques and angular velocities of the particles. Both linear and angular momenta are conserved in this model. The dissipative forces in the model are again coupled to the random forces by means of the fluctuation-dissipative theorem.

The modified FPM model was used to simulate dynamics of colloids,\textsuperscript{69,70} and later was applied to examine the clustering of red blood cells in capillary vessels.\textsuperscript{71–73} Dynamical and rheological properties of colloidal suspensions in simple fluid solvents were also investigated successfully using FPM.\textsuperscript{74} Each colloidal particle was represented by a single FPM particle and the conservative forces for solvent–colloid and colloid–colloid interactions were derived from the Lennard–Jones potentials. The simulations were focused on a range of rheological behaviors and volume fractions of the suspension. Good agreement with experimental and theoretical results for suspensions was reported.

DPD for Two-Phase Flows

As we discussed (see Eq. [11]), for a single-component DPD fluid where the conservative force coefficient is the same for all particles ($a_{ij} = a$), the equation of state (EOS) can be approximated as

$$p = \rho k_B T + a \rho^2 k_B T$$  \hspace{1cm} [30]$$

where $\rho$ is the number density of the particles and $a$ is a parameter equal to $0.101 \pm 0.001$.\textsuperscript{6} The quadratic EOS was also obtained in other systems with soft potentials.\textsuperscript{75} The parameter $a$, which is typically set by matching the dimensional compressibility, controls the thermodynamics of the DPD system. For $a < 0$, the system is unstable, and therefore, the DPD is limited to strictly repulsive forces. This means that at least a binary mixture of different kinds of particles is needed to achieve phase separation. For a single-component system, it is necessary to have a van der Waals loop in the EOS to model vapor–liquid coexistence and free-surface systems. It is possible to obtain a much wider range of possibilities for the EOS using the extension of the DPD developed in Refs. 76 and 77. That method is called multibody DPD (MDPD) because the conservative force in it is effectively a many-body force. The key difference between MDPD and DPD is that the conservative force in MDPD depends not only on the interparticle distance but also on the instantaneous local particle density. The interparticle force can be thought of as being derived from the excess free energy of the system, which gives freedom to the modeler to choose
the thermodynamics of the system. A van der Waals fluid was considered in Refs. 76 and 77 with an EOS that had additional cubic terms in density. The attractive and repulsive components of the conservative force between the particles had the same cut-off radius in those studies. Correction for particle correlations for strongly nonideal systems and the multicomponent generalization was described by Trofimov et al. Vapor–liquid coexistence was modeled by Warren using the model with different cut-off radii for the attractive and repulsive components of the conservative force. In the work of Tiwari and Abraham, the free energy of the fluid depends on the density and its gradients, which account for the excess free energy in the interfacial region and are related to the long-range attractive forces. Finally, we note that adhesive, solid boundaries for studying wetting phenomena in the framework of MDPD were developed in Ref. 82.

Another modification of the DPD method that uses a combination of short-range repulsive and long-range attractive forces to simulate liquid and liquid–gas systems was presented by Liu et al. In this model, the interaction potential is a combination of smoothed particle hydrodynamics weight functions with different interaction strengths and ranges.

Other Extensions

The viscosity of the DPD fluid can be controlled with the dissipative force coefficient that characterizes the strength of the drag force between interacting particles. Two methods for DPD simulations of binary mixtures with different viscosities were developed. Each fluid in these methods has its own dissipative force coefficient yielding the viscosity of that fluid. An extension of DPD for simulations at constant pressure was developed, and electrostatic interactions have been included.

APPLICATIONS

One of the key attractive features of DPD is the simplicity of modeling complex fluids by specifying appropriate interactions between DPD particles. By grouping and freezing particles, solid walls in flow simulations can be modeled, and by updating the positions of frozen particles according to the laws of rigid body motion, colloidal suspensions can be investigated. Several types of DPD particles with variable repulsive strength between them allows one to model multicomponent systems. By linking collections of particles into chains with appropriate spring forces between them, simple models of polymers can be created. Amphiphilic molecules are built by linking particles of different types. Coarse-grained simulations of lipid bilayers and whole vesicles can be carried out with ease using DPD. More and more papers on DPD are published each year expanding the range of its applications. Below we will mention just a few of them.
Polymer Solutions and Melts

A polymer chain can be represented in DPD by a chain of particles (beads) connected by springs. For polymer solutions, the solvent is represented explicitly in DPD simulations. The polymer beads exchange momentum with other beads in the chain according to the spring force in addition to ordinary DPD interactions. Hydrodynamic and thermodynamic interactions between the polymer and solvent then emerge naturally in these simulations. Similar to fluid particles that can be thought of as packets of fluid, the polymer beads can be thought of as polymeric chain segments consisting of number of monomeric units. Numerous simulations have verified that the DPD model can capture many essential physical phenomena of the polymer systems.

One of the first applications of DPD to simulations of dilute polymer solution was presented by Schlijper, Hoogerbrugge, and Manke.\(^9\) They investigated static and dynamic scaling relationships and showed that the scaling of radius of gyration and relaxation time with the number of beads is consistent with the predictions of the Rouse–Zimm model. The effect of solvent quality was investigated by Kong et al.\(^9\) Solvent quality (i.e., good solvent vs. poor solvent) was adjusted by varying the amplitude of the repulsive interactions among the particles of unlike types. The behavior of the polymer radii of gyration showed a collapse transition when proceeding from good to poor solvents. Scaling laws for polymer melts (as well as dilute polymer solutions) were calculated by Spenley.\(^9\) The dynamical behavior of entangled polymer melts is governed by topological constraints that prohibit adjacent polymer chain segments from crossing through each other, thereby restricting the available dynamics paths for chain motion. A segmental repulsion force was developed by Pan and Manke\(^9\) to reduce the frequency of artificial chain segment crossing events in DPD simulations of the dynamics of polymer melts.

DPD simulations of the dynamics of a polymer chain were conducted by those same authors.\(^9\) Viscosity and first and second normal stress coefficients were evaluated as functions of shear rate by performing DPD simulations at finite shear rates, which were implemented by means of Lees–Edwards boundary conditions.\(^9\) The predicted rheological material functions included realistic features such as shear thinning of the viscosity and first normal stress coefficient, and negative values for the ratio of second to first normal stress coefficients.

Simulations of polymer brushes (solutions with polymers that are grafted at one end to the surface) have also been published.\(^5\)\(^,\)\(^9\)\(^6\)\(^–\)\(^10\)^\(^0\)

Binary Mixtures

In DPD simulations of immiscible fluid mixtures, different types of DPD particles attract similar particles and repel dissimilar ones. An example of such a physical system is a mixture of oil and water below the critical temperature. The nonpolar oil molecules are hydrophobic and attract each other through
short-range van der Waals forces. The water molecules interactions are dominated by electrostatic interactions but also include complex, long-range hydrophilic attractions. MD simulations of such systems commonly require detailed treatment at the atomistic level, but the mesoscopic model of the system can be simplified significantly. In DPD simulations, the simplest implementation of the model of immiscible fluids would be to introduce two types of DPD particles. In this model, water and oil are represented by a single bead. Natural oil, however, is composed of hydrocarbon chains, and therefore, the model, which consists of several DPD particles linked in a chain, would seem to be more appropriate. The effect of the chain length was reported to have no qualitative effect on the simulation results, however,\(^{101}\) and therefore oil molecules are typically represented by a single DPD particle. When two particles of a different type interact, the conservative force between them is increased as

\[
a_{ij} = \begin{cases} 
  a_0, & \text{particles } i \text{ and } j \text{ of the same type} \\
  a_1, & \text{particles } i \text{ and } j \text{ of different type}
\end{cases}
\]

where \(a_1 > a_0\). This will produce simulations with immiscible fluids that are identical to each other and have equal viscosities. Examples of such simulations can be found in Refs. 102–104. The binary mixture model was extended to include fluids with different viscosities\(^{85}\) and coupled with polymerization.\(^{105}\)

Interfacial tension governs the properties of the oil–water interface sufficiently far from a critical point. The surface free energy of the system is minimized by adopting a configuration that minimizes the interfacial area. The properties of the interface can be changed dramatically by adding surfactant. Because of their amphiphilic nature, surfactants adsorb to the interface between oil and water, lowering the interfacial tension and promoting mixing. The efficiency of a surfactant is defined as the negative logarithm of the surfactant concentration in bulk needed to reduce the interfacial tension by a given amount. In other words, a surfactant is more efficient than another if fewer surfactant molecules are needed to obtain a given reduction of the interfacial tension. The efficiency of a surfactant is related to its chemical structure and depends on the partitioning of the surfactant between the bulk liquid and the interface. In simulations, surfactant molecules are constructed by linking hydrophilic head particles to hydrophobic tail particles. The effects of variation of size and branching of surfactant molecules on their ability to reduce the interfacial tension have been studied.\(^{101,106}\) Depending on the surfactant structure, the interfacial tension can become so low that the free energy associated with the changes of the curvature of the oil–water interface must be taken into account. The effect of surfactant chain length on the bending modulus of surfactant monolayer was investigated.\(^{107}\) In water-in-oil and oil-in-water emulsions, which are thermodynamically unstable, surfactant molecules can slow down or prevent the droplet coalescence process making emulsions kinetically stable. Rupture of
the film separating two droplets and their coalescence in oil/water/surfactant systems have been studied.\textsuperscript{108,109}

**Amphiphilic Systems**

Amphiphilic molecules, which contain a hydrophobic tail and a hydrophilic head group, exhibit a variety of aggregates in water solution. Various amphiphilic models are used in DPD simulations. The simplest model is composed of a single head bead, denoted by $H$, and a linear chain of tail beads, denoted by $T$. The amphiphiles are immersed in a solvent composed of $W$ beads. Typically, for simplicity, the $H-H$, $T-T$, and $W-W$ particle interactions are treated the same but are different from $H-T$, $T-W$, and $H-W$ interactions. The number of head and tail beads may vary in simulations. An amphiphile containing one head and $n$ tail beads can be represented using simple symbolism as $HT_n$.\textsuperscript{110} The size of the head can be changed by using different number of head beads (i.e., $H_mT_n$). $H_m(T_n)_2$ means the amphiphile has two linear hydrophobic tails. Both tails may be connected to a single hydrophilic bead, to which the remaining hydrophilic beads are also attached, or the tails may be attached to adjacent head beads. Various phases can be observed depending on the concentration, tail, and head structure of the amphiphilic molecules. When the tail length is short, the amphiphilic molecules will aggregate into spherical or rodlike micelles, because these molecules have a strong tendency to segregate from the water and bend toward the tail. At higher concentrations, when the interactions between the different micelles cannot be neglected, formation of hexagonal and lamellar phases occur. As the tail becomes longer, spherical or rodlike micelles are no longer stable and the disk-shaped micelles prevail. If the interaction between micelles cannot be neglected, the lamellar phase will form. When the tail length is long and the head size is not too large, the lipids have a tendency to curve toward the head. This results in the formation of inverted phases. A dense solution of the amphiphilic species was studied using DPD by Jury et al.\textsuperscript{111} A minimal amphiphile model, $HT$, was considered. With suitable parameters, this simple model gave a reasonable phase diagram, which was similar to one of a wide range of nonionic surfactants. Micellar, hexagonal, and lamellar phases were found in those simulations. The evolution of the contact zone between pure surfactant and solvent using this model was studied by Prinsen, Warren, and Michels.\textsuperscript{112} The inverted phases were obtained by adding the bond angle potential to the amphiphilic model.\textsuperscript{113} The bond angle potential introduced by Shillcock and Lipowsky\textsuperscript{110} is modeled by a three-body potential acting among adjacent bead triplets in the tail chain

\begin{equation}
\mathbf{u} = k[1 - \cos(\phi - \phi_0)],
\end{equation}

where $\phi$ is the bond angle of the two adjacent bonds connecting beads and $\phi_0$ is a preferred angle between the bonds.
At moderate concentrations of amphiphile molecules in aqueous solution, spontaneous formation of vesicles was observed. Vesicles are biologically important in terms of separability of the solution for various applications such as drug delivery systems and artificial cells. In simulations from randomly dispersed system, reported in Ref. 114, the aggregation of small micelles is first observed, and then large oblate micelles (bilayer membranes) are formed. In the next stage, the oblate micelles fluctuate and encapsulate water. Finally, they close and form vesicles to lower the edge energy between the hydrophobic tails and water. Other simulations of micelles and fluid vesicles have been published.114–118

A lipid bilayer, a structural component of the cell membranes, is a biologically important example of an amphiphilic system. The bilayer membranes are complex and dynamic structures. Their composition and membrane-bound proteins play a crucial role in the normal functions of living cells. Lipid bilayers around cells are often close to a tension-free state. However, the distributions of stresses within the bilayer are not uniform. The DPD method was used by Shillcock and Lipowsky110 to investigate the equilibrium structure of amphiphilic bilayers and to measure their lateral stress profile and its dependence on the amphiphile architecture. An improved model of the lipid bilayer was later presented.119 The phase behavior of lipid bilayers was also investigated by Kranenburg et al.120,121 and induced interdigitation of bilayers consisting of double-tail lipids by alcohol molecules to the bilayer was also modeled.122 Membrane fusion was considered by Li and Liu.123 Simulations of the lipid bilayer with embedded proteins of different size and hydrophobic lengths, protein-induced bilayer deformations, and lipid-induced protein tilting have also been published.18

**Red Cells in Microcirculation**

The human red blood cell (RBC) has a biconcave shape with a diameter of approximately 8 μm. When RBCs pass through capillaries, whose diameters are comparable or smaller than that of RBC, they deform under the flow resistance to a parachute shape and return back to their original biconcave form once the hydrodynamic forces cease. The ability of RBC to deform—which originates in the membrane properties— influences, in turn, the blood flow resistance. The RBC membrane is composed of a lipid bilayer and an attached cytoskeleton. The cytoskeleton consists primarily of spectrin proteins, which form the network by linking short actin filaments. A coarse-grained model124 based on the spectrin-level RBC model of Refs. 125 and 126 was developed using mean-field theory and then applied to DPD simulations in capillaries of 10 microns in diameter. In capillaries of this diameter, the blood velocity is typically about 1 mm/s.127 The RBC was modeled as a collection of DPD particles that was then immersed in a DPD fluid. The RBC particles interacted with the fluid particles through DPD potentials, and the temperature of the system was controlled through the
Figure 3  Successive snapshots of deformation of a coarse-grained model from the DPD simulations of the RBC flow in a microchannel. From left to right: (a) The RBC is placed in the channel with the fluid at rest. (b), (c) The deformation of the model 0.008 and 0.016 seconds after the body force driving the flow is applied. (d) The shape of the model at steady flow. (e) The model recovers its equilibrium biconcave shape 0.2 seconds after the body force is turned off. Only a portion on the microchannel is shown for clarity. (Reprinted with permission from Ref. 124).

DPD thermostat. Initially, the fluid is at rest, the RBC is placed in the middle of the capillary, and a body force is applied in the axial direction to drive the flow. The RBC was found to deform under the flow conditions and, after some transition period, assumed the parachute-type shape shown in Figure 3, which is commonly observed in experiments. After the body force is turned off, the flow slows down and eventually the DPD fluid returns to rest, whereas the RBC recovers its equilibrium biconcave shape.

SUMMARY

In this chapter we have considered dissipative particle dynamics method. The DPD method is relatively new; however, it has already established its place among other mesoscale simulation methods. Despite considerable attention in recent years, the DPD is still evolving. New variations and new interpretations of the method appear often in literature. The method has been applied to a large number of problems, but new applications are still emerging. Indeed, the ideas behind DPD are very attractive and there is plenty of room for future developments and improvements.

REFERENCES


