A Seamless Approach to Multiscale Complex Fluid Simulation

Dissipative particle dynamics is an efficient and accurate mesoscale simulation method that bridges the gap between molecular dynamics and continuum hydrodynamics. Using new time-staggered algorithms, it can simulate complex liquids and dense suspensions more than 100,000 faster than molecular dynamics.

The molecular dynamics (MD) method is suitable for simulating very small volumes of liquid flow with linear dimensions of 100 nanometers or less and for time intervals of several tens of nanoseconds. It effectively deals with nanodomains and is perhaps the only accurate approach for simulating flows involving very high shear in which continuum hydrodynamics or the Newtonian hypothesis might not be valid.

For length scales less than approximately 10 molecules, the continuum hypothesis breaks down for liquids, and MD should be used to simulate such a system's atomistic behavior.[1] For larger systems, multiscale approaches that rely on the efficiency of continuum-based discretizations must be employed. To this end, the coupling of MD to Navier–Stokes equations can extend the range of applicability of both approaches and provide a unifying description of liquid flows from nanoscales to larger scales. Efforts have been underway by many research groups,[2] but the proposed algorithms are complicated and not fully satisfactory because they either introduce noise at the interface or they may not conserve mass or momentum.

In this article, we demonstrate the potential and accuracy of the dissipative particle dynamics (DPD) method by computing well-known polymer scaling laws on a laptop computer. Specifically, we'll focus on using the basic DPD framework to formulate, implement, and compare different models for polymer chains in dilute solutions, and then we'll outline a time-staggered integrating scheme that efficiently addresses the issue of different time scale resolution requirements for various interactions.

Dissipative Particle Dynamics

In the mid-1990s, DPD emerged as a simple yet potentially very powerful alternative for mesoscopic simulations. It has features of MD and the lattice Boltzmann method (LBM),[3] and can be thought of as a coarse-grained version of MD, but it uses dissipative and stochastic forces to account for the degrees of freedom it eliminates with respect to MD. P. J. Hoogerbrugge and J.M.V.A. Koelman proposed the initial model as a simulation method to avoid artifacts associated with traditional LBM simulations while capturing spatio–temporal hydrodynamic scales much larger than MD could achieve.[4]

The DPD model consists of particles that corre-

1521-9615/05/$20.00 © 2005 IEEE
Copublished by the IEEE CS and the AIP

Vasileios Symeonidis, George Em Karniadakis, and Bruce Caswell
Brown University

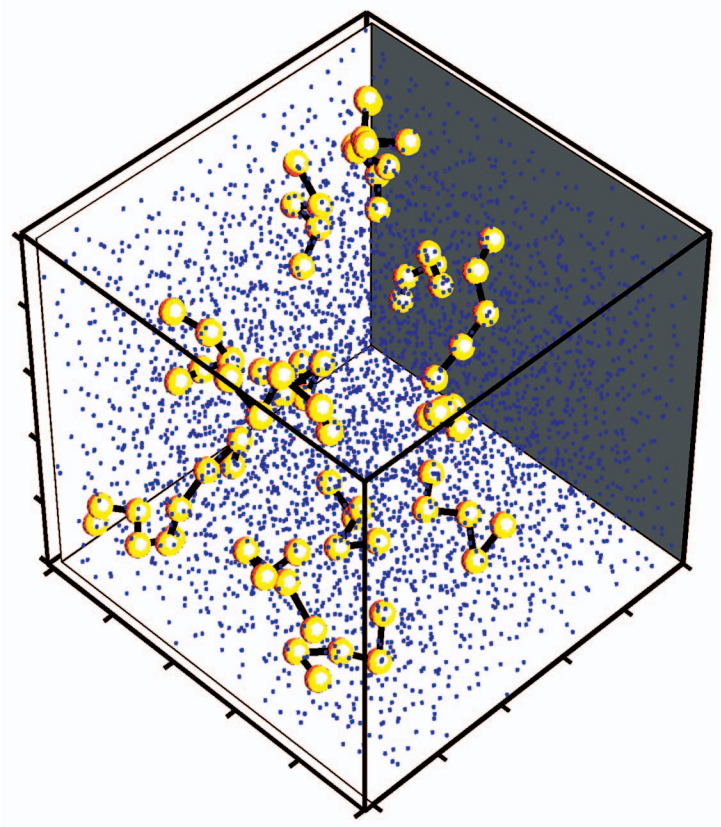

Figure 1. Dilute polymer solutions. Polymer chains (tethered spheres) are suspended in a solvent of dissipative particle dynamics (DPD) particles (smaller dots).

spond to coarse-grained entities, thus representing molecular clusters rather than individual atoms. The particles interact through a set of prescribed (conservative and stochastic) and velocity-dependent forces.[4,5] Specifically, three force types act on each dissipative particle:

- a purely repulsive conservative force,
- a dissipative force that reduces velocity differences between the particles, and
- a stochastic force directed along the line connecting the centers of the particles.

The last two forces effectively implement a thermostat to achieve thermal equilibrium. Correspondingly, these forces' amplitude is dictated by the fluctuation–dissipation theorem,[5] which ensures that the system will have a canonical distribution in thermodynamic equilibrium. A weight function modulates all three forces by specifying the range of interaction, or cut-off radius r_c, between the particles and renders the interaction local.

Thus, a conceptual picture of DPD is soft microspheres randomly moving around but following a preferred direction dictated by the conservative forces. We can interpret DPD as a Lagrangian discretization of the equations of fluctuating hydrodynamics because the particles follow the classical hydrodynamic flow while exhibiting thermal fluctuations. The principles of statistical mechanics govern the consistency of the fluctuations.

Complex Fluids

Several complex fluid systems in industrial and biological applications (for example, DNA chains, polymer gels, and lubrication) are characterized by inherent time and length scales that range from the atomistic level to a millimeter and beyond, often spanning several orders of magnitude. Traditional MD techniques attack the problem at the microscopic level, whereas continuum models might fail to capture smaller interactions because they resort to averaging techniques or predefined association rules. Dilute polymer solutions are a typical example, because individual polymer chains form a group of molecules that is large by atomic standards but still governed by forces similar to intermolecular ones. Thus, they form large repeated units exhibiting slow dynamics with possibly nonlinear interactions (see Figure 1).

The DPD method is attractive for computer simulations of polymer solutions because by employing bead–spring polymer chain representations, we can formulate and compare a variety of realistic conservative interbead forces.

To appreciate DPD's potential and computational complexity, we summarize the governing equations for simple and complex fluids, and subsequently present a time-staggered algorithm and typical results.

DPD Equations

A typical DPD system consists of N particles having (for simplicity) equal mass m, positions $\mathbf{r}_i$, and velocities $\mathbf{u}_i$. The repulsive, dissipative, and stochastic forces exerted on a particle i by particle j are given by

$$\mathbf{F}_{ij}^{c} = F^{(c)}(r_{ij})\mathbf{e}_{ij},$$

$$\mathbf{F}_{ij}^{d} = -\gamma\omega^{d}(r_{ij})(\mathbf{u}_{ij}\cdot\mathbf{e}_{ij})\mathbf{e}_{ij},$$

$$\mathbf{F}_{ij}^{r} = \sigma\omega^{r}(r_{ij})\xi_{ij}\mathbf{e}_{ij},$$

where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, $\mathbf{u}_{ij} = \mathbf{u}_i - \mathbf{u}_j$, $r_{ij} = |\mathbf{r}_{ij}|$, and the unit vector $\mathbf{e}_{ij} = \mathbf{r}_{ij}/r_{ij}$. The parameters γ and σ deter-

mine the strength of the dissipative and random forces, respectively, the ξ_{ij} are symmetric Gaussian random variables with zero mean and unit variance, and ω^d and ω^r are weight functions. All forces act within a sphere of radius r_c, which defines the system's length scale.

By averaging the Lennard–Jones (LJ) potentials or the corresponding molecular field over the rapidly fluctuating motions of atoms during short time intervals, we get an effective average potential[6] in the form of a *soft*, *repulsive-only* interaction (see Figure 2). A linear approximation of this is $\mathbf{F}^c_{ij} = a_{ij}(1 - r_{ij})\mathbf{e}_{ij}$ for $r_{ij} \leq r_c = 1$ and is otherwise zero.[7] Unlike the hard LJ potential, which is unbounded at $r = 0$, the *soft potential* employed in DPD has the finite value a_{ij} at $r = 0$. To find the value of a_{ij}, we follow the process laid out by Robert Groot and Patrick Warren[7] and Kenneth Rabone[7,8]—that is, we match the DPD system's dimensionless compressibility with that of the MD system:

$$\begin{aligned}\kappa^{-1}|_{\mathrm{DPD}} &= \frac{1}{k_B T_{\mathrm{DPD}}}\left[\frac{\partial p_{\mathrm{DPD}}}{\partial \rho_{\mathrm{DPD}}}\right]_{\mathrm{T}} \\ &= \frac{1}{k_B T_{\mathrm{MD}}}\left[\frac{\partial \rho_{\mathrm{MD}}}{\partial \rho_{\mathrm{DPD}}}\right]\left[\frac{\partial p_{\mathrm{MD}}}{\partial \rho_{\mathrm{MD}}}\right]_{\mathrm{T}} \\ &= N_m \kappa^{-1}|_{\mathrm{MD}}, \end{aligned} \quad (1)$$

where ρ is the number density, $N_m = \partial\rho_{\mathrm{MD}}/\partial\rho_{\mathrm{DPD}}$ is the coarse-graining parameter, k_B is the Boltzmann constant, and T the system temperature. In Equation 1, "DPD" refers to simulation and we note that in "MD" we have $N_m = 1$. Then, from the equation of state, Groot and Warren[7] give a_{ij} through

$$\kappa^{-1}|_{\mathrm{DPD}} \approx 1 + 0.2\frac{a\rho_{\mathrm{DPD}}}{k_B T_{\mathrm{DPD}}}.$$

By matching the diffusion constant (D_{DPD}) in the DPD simulation with that of water (D_{water}), we find the DPD time scale as

$$\tau = \frac{N_m D_{\mathrm{DPD}} r_c^2}{D_{\mathrm{water}}} \propto N_m^{5/3}.$$

This time scale and the soft potential explain why the DPD method is several orders of magnitude faster than straightforward MD. With respect to MD, the soft potential removes an atom's *caging effect*—constraints on an atom's motion due to nearby hard repulsions—so that the atom diffusivity increases by a factor of about 1,000, depending on the thermostat. (Christopher Lowe's approach,[9] for example, employs an Andersen thermostat, which doesn't decrease the Peclet number.) The effect of the time scale is to decrease the corresponding CPU time in proportion to the coarse-graining parameter N_m; hence, the total speedup with respect to MD is $1{,}000 \times N_m \times N_m^{5/3}$ for a given system volume. Thus, for $N_m = 5$ and 10, the speedup factor is 73,000 and 464,000, respectively.

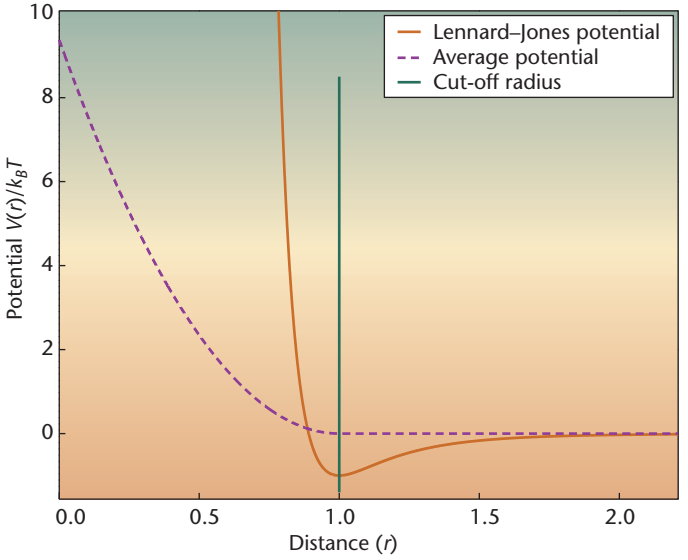

Figure 2. Lennard–Jones (LJ) potential and the soft-repulsive potential that results after averaging. The steep negative slope of the LJ potential gives the hard repulsion, compared with the moderate slope of the soft curve.

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

$$d\mathbf{r}_i = \mathbf{u}_i\, dt;\ \mathrm{k},$$

where $\mathbf{F}^c_i = \sum_{i \neq j}\mathbf{F}^c_{ij}$ is the total conservative force acting on particle i; $\mathbf{F}^d_i$ and $\mathbf{F}^r_i$ are defined similarly. The random force, which represents Brownian motion, appears with a factor of $\sqrt{dt}$ in the velocity increment. The dissipative and random forces, characterized by strengths $\omega^d(r_{ij})$ and $\omega^r(r_{ij})$, respectively, are coupled through the *fluctuation-dissipation* theorem[5] as follows:

$$\omega^d(r_{ij}) = [\omega^r(r_{ij})]^2 = \max\left\{\left(1 - \frac{r_{ij}}{r_c}\right)^2, 0\right\}$$

$$\sigma^2 = 2\gamma k_B T.$$

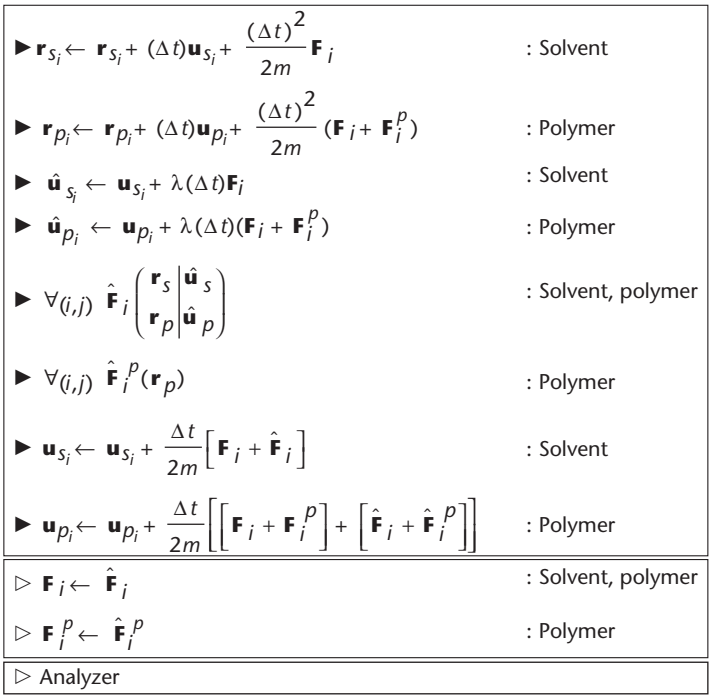

Figure 3. Overview of the traditional velocity–Verlet approach. It uses a single time step to advance all particles in the simulation.

Polymer Models

Unlike the MD equations, DPD equations are stochastic and nonlinear because the dissipative force depends on the velocity. For complex fluids in particular, the presence of both soft and hard potentials suggests the use of time-staggered algorithms for integrating the DPD motion equations. This allows the efficient study of useful polymeric physical quantities, such as the polymeric chain's radius of gyration.

The conservative forces present in usual DPD equations can be tailored to describe a variety of interactions—for example, LJ, linear and nonlinear springs, and van der Waals. Figure 2 illustrates the need for two different temporal resolutions: the LJ potential (for bead–bead pairs) is a hard repulsion that requires a time step much smaller than the soft interaction forces of a typical DPD particle pair (which we can think of as an averaged hard potential).

Figure 1 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. They are also subject to intra-polymer forces, arising from different combinations of the following types.

- *LJ*. The force for each pair of bead particles is given by the shifted (to avoid numerical instabilities) LJ potential

$$U_{\mathrm{LJ}} = 4\varepsilon\left[\left(\frac{L}{r_{ij}}\right)^{12} - \left(\frac{L}{r_{ij}}\right)^{6} + \frac{1}{4}\right],$$

  which is truncated to act *only* for pairs with $r_{ij} < r_c$. For the above formula, we pick $\varepsilon = k_BT$, $L = 2^{-1/6}$, and $r_c = L2^{1/6} = 1$. The LJ potential used here is defined at the mesoscopic level to improve polymeric chain self-avoidance (discussed later).
- *Hookean spring*. The interbead force is derived from a pairwise harmonic potential with spring constant κ,

$$U_{\mathrm{HOOKE}} = \frac{\kappa}{2}\left|\mathbf{r}_i - \mathbf{r}_{i-1}\right|^2, \text{ where } i = 2, 3, \ldots, M.$$

- *Fraenkel (stiff) spring*. The interbead force is derived from a pairwise Hookean potential with equilibrium length r_{eq}. Stretched to a length greater than r_{eq},the spring exerts an attractive force; when pushed to one smaller than r_{eq},it exerts a repulsive one. Its potential is

$$U_{\mathrm{STIFF}} = \frac{\kappa}{2}\left(\left|\mathbf{r}_i - \mathbf{r}_{i-1}\right| - r_{\mathrm{eq}}\right)^2, \text{ where } i = 2, 3, \ldots, M \text{ and } \kappa \text{ the spring constant.}$$

- *FENE spring*. Within a chain of M beads, each is subject to a pairwise nonlinear spring force. FENE springs have a maximum extensibility $r_{\max}$ beyond which the force becomes infinite, hence, any length greater than $r_{\max}$ isn't allowed. The potential is described by

$$U_{\mathrm{FENE}} = \sum_i -\frac{\kappa}{2} r_{\max}^2 \log\left[1 - \frac{\left|\mathbf{r}_i - \mathbf{r}_{i-1}\right|^2}{r_{\max}^2}\right], \text{ where } = 2, 3, \ldots, M \text{ and } \kappa \text{ the spring constant.}$$

The interbead force is given in each case by $\mathbf{F}^p = -\nabla U$.

Time-Staggered Velocity–Verlet Algorithm

The basic DPD integrator is a modified version of the classical velocity–Verlet (first proposed by Groot and Warren[7]). The velocity–Verlet scheme is characterized by explicit calculation of all forces $\mathbf{F}^c$, $\mathbf{F}^d$, and $\mathbf{F}^r$ (conservative, dissipative, and random) and is step-dependent, yet straightforward and relatively accurate. It relies on a basic predictor–corrector approach that uses velocity provisional values for the force calculations, which are

corrected at the end of each time step. DPD dissipative forces depend on the particles' relative velocities, hence, this prediction is crucial.

Next, we outline the modified velocity–Verlet scheme with parameter λ, whose theoretical value is 0.5, but researchers have shown empirically that for a certain range of Δt, its optimal value is closer to 0.65.[7] In this article, we use the latter because our time step is in the optimal range of 10^{-2}.

Denoting the total forces by

$$\mathbf{F}_i = \Sigma_{j \neq i} \left[\mathbf{F}_{ij}^c + \mathbf{F}_{ij}^d + \frac{\mathbf{F}_{ij}^r}{\sqrt{\Delta t}} \right]$$

and the extra polymeric forces by $\mathbf{F}_i^p = \sum_{j \neq i} \mathbf{F}_{ij}^p$, Figure 3 outlines the basic (classical thermostat) velocity–Verlet scheme (subscripts p and s correspond to polymer and solvent quantities, respectively).

To extend this algorithm for simulating complex fluids with soft–hard potentials, we use a large time step Δt for solvent particles and a smaller one δt for polymer particles that belong to a chain. Thus, we use provisional values for the solvent and polymer velocities and the polymer's position. The CPU-expensive step of collective force computation is done only once. The velocity and the position of the polymer are corrected in the subsequent loop, in which we integrate the polymer particles $\Delta t/\delta t$ times in a separate subcycle (using δt for the time step). The varying polymeric force $\mathbf{F}_i^p$ is updated in the subcycle, following the change in $\mathbf{r}_{p_i}$ (the position of the polymer particles). Hence, during the subcycle, we update the intrapolymer forces, but not the interparticle (total) ones, because it would require CPU time for each subcycle equivalent to a standard one. Although we can't expect exact agreement of the new scheme with the classical one, we can anticipate small differences if the ratio $\Delta t/\delta t$ isn't too large and if the (outdated) forces are applied in the correct manner during the δt cycle. The proposed scheme is the simplest version of a family of time-staggered schemes under development, which use a wider range of relaxation parameters for the prediction and the scaling of the applied forces in the subcycle.[10] Figure 4 summarizes the algorithm.

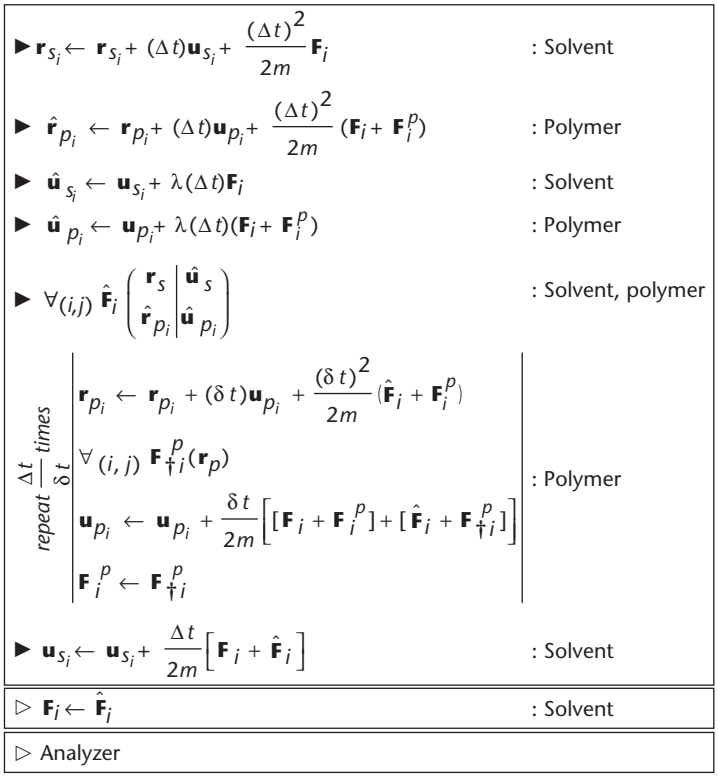

Figure 4. Overview of the time-staggered velocity–Verlet approach. It uses two different time steps, a large one for solvent particles and a small one for particles that belong to a polymer chain (beads).

Computational Complexity and Accuracy

CPU time savings is the prime motivation for using a time-staggered scheme with two different time steps. Figure 5a summarizes results for four different chains in a 4,000-particle DPD simulation, each having 20, 50, and 100 beads. Efficiency depends (among other factors) on how we handle the intrapolymer pairwise interactions. Because all the forces are pairwise, researchers usually don't explicitly compute all the pairs in the domain—instead, they introduce neighbor (or cell) lists or boxes and search only within them. This dramatically reduces the computational cost, which would be quadratic in N (the total number of DPD particles). Although we use a brute-force method for searching through all the pairs in a chain, further improvement is possible by using a linked-list method for the polymer chain, as we do for the solvent, but this benefits only large-chain systems. If we consider a staggered simulation of $\{\Delta t, \delta t\}$ time steps, the method's speedup is defined as the ratio

$$\text{Speedup} = \frac{\left[\text{total CPU time for a } \Delta t \text{ simulation}\right]}{\left[\begin{array}{l}\text{total CPU time to advance to the same}\\ \text{solution time for a hybrid staggered}\\ \{\Delta t, \delta t\} \text{ simulation}\end{array}\right]}.$$

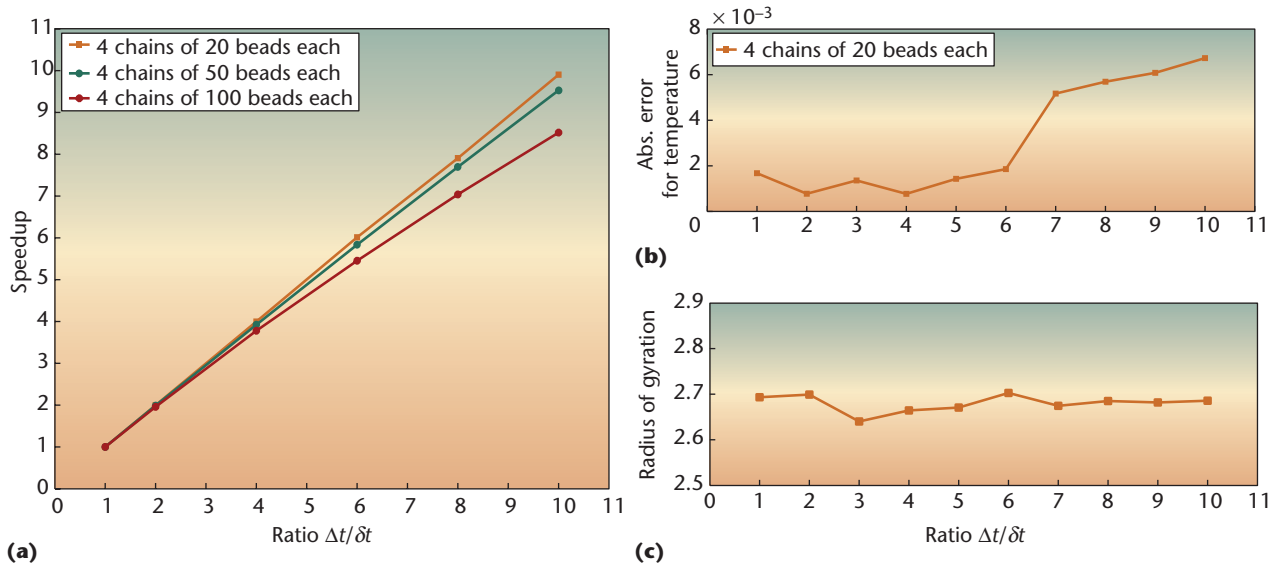

Figure 5. Efficiency and accuracy of the proposed time-staggered method. (a) Speedup, (b) temperature, and (c) radius of gyration accuracy tests versus $\Delta t/\delta t$ for the modified velocity–Verlet method for a 4,000 dissipative particle dynamics (DPD) fluid. The polymer beads interact with each other through a finitely extensible nonlinear elastic (FENE) force and a pairwise Lennard–Jones (LJ) hard repulsion.

The chosen polymer force largely influences the speedup: LJ interactions require calculations of all possible pair combinations, whereas spring forces alone don't. Figure 5 shows the effect of the $\Delta t/\delta t$ ratio on both the temperature and the radius of the polymer chain's radius of gyration R_g. The absolute k_BT error due to the time-staggering alone doesn't exceed 0.008; similarly, the effect on R_g is of the order of 10^{-2}. In fact, the error largely depends on the stiffness of the polymer forces. In our accuracy tests, we deliberately picked the spring constant $\kappa = 7$ and the FENE parameter $r_{max} = 3r_c$, which resulted in a relatively stiff system of forces, considering that the effective LJ range is $2r_c$ (r_c is the DPD radius cut-off). For higher-order schemes, we anticipate an even smaller accuracy loss. For the same examples, the effect of the $\Delta t/\delta t$ ratio on solvent temperature is negligible.

Polymer Scaling Laws

In effect, our DPD models represent dilute polymer solutions, thus, the dynamics of a single flexible polymer chain is of great importance for validation and physical understanding the DPD methods. In 1995, researchers used stiff (Fraenkel) and weak (Hookean) springs, without hard LJ potentials, to map polymer-chain scaling exponents to DPD results.[11] Our approach introduces more complex (nonlinear) forces, and the complex process of phantom collisions appears to be negligible.

Ideal chains are characterized by linear springs that exhibit *phantom collisions*—that is, polymeric bonds can pass through each other. Alternatively, real chains in solvents that energetically favor solvent–polymer interactions (good solvents) behave like self-avoiding walks on a lattice that forbid phantom collisions. The radius of gyration R_g of a chain of M beads essentially measures the size of the chain and can reveal information about the occurrence of phantom collisions. It is defined as

$$\left\langle R_g^2 \right\rangle = \left\langle \frac{1}{M} \sum_{i=1}^{M} \left(R_i - R_{cm} \right)^2 \right\rangle ,$$

where R_i is the position vector of each bead, R_{cm} is the chain's center of mass position vector, and $\langle \cdot \rangle$ denotes time-averaging. Pierre de Gennes considered a critical exponent for a single chain ν relating to chain size.[12] An ideal chain has scaling exponent of $\nu = 0.5$, whereas a real chain scales with the Flory exponent $\nu = 3/(d + 2) = 0.6$ for three dimensions. We can consider the Flory formula exact for all practical purposes. Linear chain DPD simulations[11] have shown a close mapping to the 0.5 exponent, which, in turn, relates to the Rouse–Zimm harmonic spring model.[13] Thus, R_g for ideal chains scales as

$$R_g \propto (M - 1)^{0.5},$$

whereas light-scattering measurements verify the chain size power law[12] to be

$$R_g \propto (M-1)^{0.6}.$$

Our simulations aim to accurately describe real chains. Figure 6 shows a very good agreement of scaling exponents 0.5 and 0.6 plotted against the chain size. The LJ repulsion seems to be primarily responsible for capturing the self-avoidance, whereas the underlying spring force (Hookean or FENE force of maximum extensibility) seems to have a secondary effect on the scaling exponent when coupled with hard repulsions.

Our results used a periodic domain of equal dimensions $L_x \times L_y \times L_z$ in each direction, where L_x = 10 and the number density ρ is set to 4 (which fixes the total number of DPD particles to 4,000). Moreover, to accommodate different chain size simulations, we stored the polymer coordinates without mapping them back in the original domain. This allowed the intrapolymer forces to be calculated properly, while the collective solvent–solvent and polymer–solvent interactions were calculated with the mapped (periodic) images.

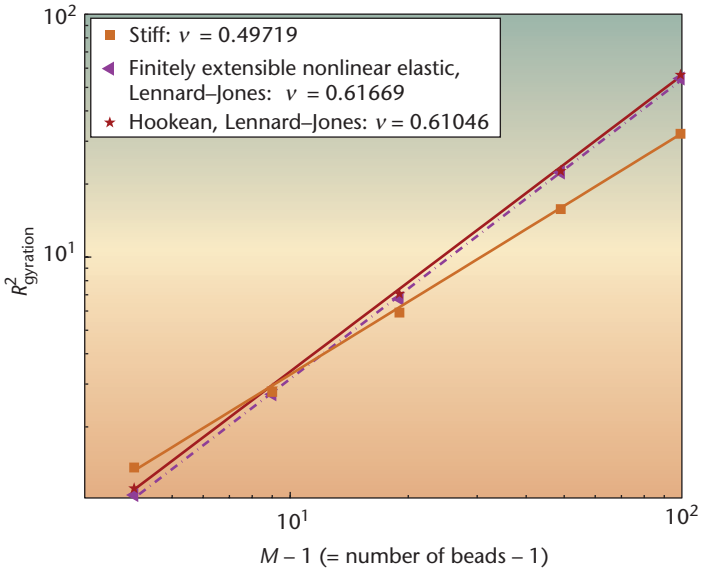

Figure 6. Polymer static exponent. Scaling of the radius of gyration of a single polymer chain for Hookean and finitely extensible nonlinear elastic (FENE) springs coupled with Lennard–Jones (LJ) forcing, compared to stiff (Fraenkel) springs. The solvent consists of 4,000 dissipative particle dynamics (DPD) particles, and the chain sizes vary from 5, 10, 20, 50, up to 100 beads. Here κ = 7 for all models, $r_{eq} = r_c$, and $r_{max} = 2r_c$.

We used DPD to simulate polymer chains with more than 100 beads on a laptop computer (3-GHz Intel Pentium IV with 1-Gbyte memory) for long times (seconds), similar to Brownian dynamics (BD) simulations.[14] However, unlike BD, in which the simulation imposes a prescribed flow and only one-way interactions, DPD accounts for two-way interactions between the polymer chains and the solvent. DPD has good scalability and thus can be employed to simulate much more complex fluid systems, from nanometer to millimeter scales and beyond.

For DPD to realize its full potential, we must address several issues. These issues primarily relate to boundary conditions, complex-geometry domains, high-order time integrators, and validation against available experimental results. To impose the no-slip boundary condition in this method isn't a trivial matter, for example, because the soft potentials allow for particles to escape the domain; hence, we need to develop special procedures based on the wall–fluid interparticle force.[15] With respect to validation, no DPD results are available in the literature for high shear rates or for large extensibility.

Another open issue relates to the limits of the coarse-graining procedure that DPD uses—that is, defining the largest value of N_m (Equation 1). In preliminary work, we've observed that the DPD results exhibit relatively large fluctuations, even for simple fluids subject to modest shear rates for $N_m > 5$, and this might be related to boundary conditions.[16]

Finally, we've made no attempts to interface DPD with MD or with continuum hydrodynamics. We believe such coupling will be more successful than previous efforts attempting to couple MD–continuum directly because DPD exhibits the correct behavior in both the atomistic and the continuum limits.

Acknowledgments

This work was supported by US National Science Foundation grant number CTS-0326720.

References

1. G.E. Karniadakis and A. Beskok, *Microflows: Fundamentals and Simulation*, Springer, 2001.
2. X.B. Nie et al., "A Continuum and Molecular Dynamics Hybrid Method for Micro- and Nano-Fluid Flow," *J. Fluid Mechanics*, vol. 500, Feb. 2004, pp. 55–64.
3. S. Succi, *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*, Oxford Univ. Press, 2001.
4. P.J. Hoogerbrugge and J.M.V.A. Koelman, "Simulating Microscopic Hydrodynamic Phenomena with Dissipative Particle Dy-

namics," *Europhysics Letters*, vol. 19, no. 3, 1992, pp. 155–160.

5. P. Espanol and P. Warren, "Statistical Mechanics of Dissipative Particle Dynamics," *Europhysics Letters*, vol. 30, no. 4, 1995, pp. 191–196.
6. B.M. Forrest and U.W. Suter, "Accelerated Equilibration of Polymer Melts by Time-Coarse-Graining," *J. Chemical. Physics*, vol. 102, no. 18, 1995, pp. 7256–7266.
7. R.D. Groot and P.B. Warren, "Dissipative Particle Dynamics: Bridging the Gap Between Atomistic and Mesoscopic Simulation," *J. Chemical Physics*, vol. 107, no. 11, 1997, pp. 4423–4435.
8. R.D. Groot and K.L. Rabone, "Mesoscopic Simulation of Cell Membrane Damage, Morphology Change and Rupture by Nonionic Surfactants," *Biophysical J.*, vol. 81, Aug. 2001, pp. 725–736.
9. C.P. Lowe, "An Alternative Approach to Dissipative Particle Dynamics," *Europhysics Letters*, vol. 47, no. 2, 1999, pp. 145–151.
10. V. Symeonidis, "Numerical Methods for Multiscale Modeling of Non-Newtonian Flows," doctoral dissertation, Brown Univ., 2005.
11. A.G. Schlijper, P.J. Hoogerbrugge, and C.W. Manke, "Computer Simulation of Dilute Polymer Solutions with the Dissipative Particle Dynamics Method," *J. Rheology*, vol. 39, no. 3, 1995, pp. 567–579.
12. P.-G. de Gennes, *Scaling Concepts in Polymer Physics*, Cornell Univ. Press, 1979.
13. R.G. Larson, *The Structure and Rheology of Complex Fluids*, Oxford Univ. Press, 1999.
14. T. Schlick et al., "Computational Challenges in Simulating Large DNA over Long Times," *Computing in Science & Eng.*, vol. 2, no. 6, 2000, pp. 38–51.
15. I. Pivkin and G.E. Karniadakis, "A New Method to Impose No-Slip Boundary Conditions in Dissipative Particle Dynamics," to be published in *J. Computational Physics*, 2005.
16. E. Keaveny et al., "A Comparative Study between Dissipative Particle Dynamics and Molecular Dynamics for Simple- and Complex-Geometry Flows," submitted to *J. Chemical Physics*, 2004.

How to Reach *CiSE*

Writers

For detailed information on submitting articles, write to cise@computer.org or visit www.computer.org/cise/author.htm.

Letters to the Editors

Send letters to Jenny Ferrero, Contact Editor, jferrero@computer.org. Please provide an email address or daytime phone number with your letter.

On the Web

Access www.computer.org/cise/ or http://cise.aip.org for information about CiSE.

Subscribe

Visit https://www.aip.org/forms/journal_catalog/order_form_fs.html or www.computer.org/subscribe/.

Subscription Change of Address (IEEE/CS)

Send change-of-address requests for magazine subscriptions to address.change@ieee.org. Be sure to specify CiSE.

Subscription Change of Address (AIP)

Send general subscription and refund inquiries to subs@aip.org.

Missing or Damaged Copies

If you are missing an issue or you received a damaged copy (IEEE/CS), contact membership@computer.org. For AIP subscribers, contact kgentili@aip.org.

Reprints of Articles

For price information or to order reprints, send email to cise@computer.org or fax +1 714 821 4010.

Reprint Permission

To obtain permission to reprint an article, contact William Hagen, IEEE Copyrights and Trademarks Manager, at copyrights@ieee.org.

www.computer.org/cise/

Vasileios Symeonidis is a PhD candidate in the Division of Applied Mathematics at Brown University. His research interests include the development of numerical methods for the continuum and discrete simulation of non-Newtonian fluids. Symeonidis has an M.Math in mathematics from the University of Oxford, UK, and an MSc in applied mathematics from Brown University. He is a member of Sigma Xi, the AMS, and the American Institute of Chemical Engineers. Contact him at sjoh0341@dam.brown.edu.

George Em Karniadakis is professor of applied mathematics at Brown University. He pioneered spectral methods on unstructured grids, parallel simulations of turbulence in complex geometries, and microfluidics simulations. Karniadakis has a PhD in mechanical engineering from MIT. He is a fellow of the APS and ASME, and a member of AIAA and SIAM. Contact him at gk@dam.brown.edu.

Bruce Caswell is professor emeritus at the Division of Engineering at Brown University. His research interests are in the fluid mechanics of non-Newtonian fluids. Caswell has a PhD in chemical engineering from Stanford University. He is a member of the American Institute of Chemical Engineers and the Society of Rheology. Contact him at caswell@dam.brown.edu.