

Research Article

A Parallelization Scheme for New DPD-B Thermostats

S. A. Moga,¹ G. Dragoi,² A. Hadar,¹ and N. Goga^{2,3}

¹ Faculty of Engineering and Management of Technological Systems (I.M.S.T.), University Politehnica of Bucharest, Splaiul Independenței 313, 060042 Bucharest, Romania

² Faculty of Engineering in Foreign Languages (F.I.L.S.), University Politehnica of Bucharest, Splaiul Independenței 313, Room JE 105, 060042 Bucharest, Romania

³ Molecular Dynamics Group, Gron Inst Biomolecular Sciences & Biotechnology, Faculty of Mathematics and Natural Sciences, University of Groningen, Nijenborgh 7, 9747 GN Groningen, The Netherlands

Correspondence should be addressed to S. A. Moga; moga.sunita.andreea@gmail.com

Received 28 March 2013; Revised 7 August 2013; Accepted 9 September 2013

Academic Editor: George C. King

Copyright © 2013 S. A. Moga et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents the MPI parallelization of a new algorithm—DPD-B thermostat—for molecular dynamics simulations. The presented results are using Martini Coarse Grained Water System. It should be taken into account that molecular dynamics simulations are time consuming. In some cases the running time varies from days to weeks and even months. Therefore, parallelization is one solution for reducing the execution time. The paper describes the new algorithm, the main characteristics of the MPI parallelization of the new algorithm, and the simulation performances.

1. Introduction

Molecular simulation (MD) is a useful tool for studying the physicochemical properties of molecular systems. Nowadays, MD is one method used by the scientific community to analyze the properties of polymers, proteins, lipids, and other cellular systems. It is also used as an alternative to the laboratory experiments for the design of new materials. Molecular dynamics is based on the Newtonian equations of motions.

The purpose of this paper is to present an MPI parallelization of a new molecular dynamics algorithm DPD-B that resulted from the combination of a dissipative particle dynamics (DPD) algorithm [1] with Berendsen (B) thermostat [2]. It should be taken into account that molecular dynamics simulations are time consuming. In some cases the running time varies from days to weeks and even months. Therefore, parallelization is one solution for reducing the execution time. This paper describes the new algorithm, the main characteristics of the MPI parallelization of the new algorithm, and the simulation performances. The work was done through research collaboration between Molecular Dynamics Group, University of Groningen, one of the well-known groups in the MD domain, and researchers from

Politehnica University of Bucharest. The parallelization was done in Gromacs [3, 4], an open-source software from this domain used worldwide in universities and companies.

It should be noted that, ideally, force fields used in molecular dynamics to describe particle interactions should be calibrated such that the desired reference temperature of a simulated system be maintained. However, in practice the average temperature of a simulated system deviates from the desired reference temperature. Therefore thermostats are used for assuring that a simulated system will maintain the reference temperature. Global thermostats apply a collective correction to all particles of a system while dissipative particle dynamics applies temperature corrections to pairs of particles. In this paper we present the parallelization of a new algorithm that is composed by a combination of a global thermostat and a DPD one. The new thermostat has better properties in eliminating ice-cube effects in multiscale simulations [5, 6]. The new thermostat was introduced in [7]. The goal of this paper is to present the computer science (CS) optimizations done for the combined thermostat.

This paper is organized as follows. The next section presents the DPD-B theory, the Gromacs engine of molecular dynamics, design, and implementation of the new algorithm

in Gromacs. Section 3 outlines parallelization performance and (parallel) simulation results. Section 4 discusses conclusions and future directions of development.

2. Materials and Methods

This section is organized as follows. First subsection presents global Berendsen thermostat and the next one the new DPD-B thermostat; then Gromacs engine for molecular dynamics will be described and at the end of the section, design and implementation issues will be discussed.

2.1. Global Berendsen. In what follows we present an outline of the relevant theory for global Berendsen thermostat. For an exhaustive presentation we refer to [2, 8]. Let us summarize below this theory.

A coupling between a molecular system with temperature T and a bath [2, 8] with temperature T_{ref} can be realized by inserting extra terms—friction and stochastic—in the equations of motions, which will result in a Langevin equation

$$m_i v_i' = F_i - m_i \gamma_i v_i + R(t), \quad (1)$$

where R_i is a stochastic Gaussian variable with null mean and with intensity

$$\langle R_i(t) R_j(t + \tau) \rangle = 2m_i \gamma_i k_B T_{\text{ref}} \delta(\tau) \delta_{ij}. \quad (2)$$

And F_i is the systematic force.

The strength of the coupling to the bath is determined by the γ_i constants. The correspondent physical phenomenon for (1) is the one of frequent collisions with light particles that form an ideal gas at temperature T_{ref} .

Through the Langevin equation the system is locally subjected to random noise and couples globally to a heat bath. In order to impose global coupling with minimal local disturbance we should modify (1) so that only the global coupling remains.

We will analyze how the system temperature, T , behaves under the influence of stochastic coupling.

For an easier analytical computation we chose the friction constant to be the same for all particles: $\gamma_i = \gamma$. This is a matter of choice; different classes of degrees of freedom can be coupled to the bath with different friction constants. From the derivation of the kinetic energy— E_k —we can see the time dependence of T :

$$\frac{dE_k}{dt} = \lim_{\Delta t \rightarrow 0} \frac{\sum_{i=1}^{3N} (1/2) m_i v_i^2(t + \Delta t) - \sum_{i=1}^{3N} (1/2) m_i v_i^2(t)}{\Delta t}, \quad (3)$$

where N is the number of particles, and

$$\begin{aligned} \Delta v_i &= v_i(t + \Delta t) - v_i(t) \\ &= \frac{1}{m_i} \int_t^{t+\Delta t} [F_i(t') - m_i \gamma v_i(t') + R_i(t')] dt. \end{aligned} \quad (4)$$

Using the relation (2),

$$\sum_{i=1}^{3N} \int_t^{t+\Delta t} dt' \int_t^{t'+\Delta t} dt'' R_i(t') R_i(t'') = 6N m \gamma k_B T_{\text{ref}} \Delta t, \quad (5)$$

and the fact that $R_i(t')$ is uncorrelated with $v_i(t)$ and $R_i(t)$ for $t' > t$, we obtain

$$\frac{dE_k}{dt} = \sum_{i=1}^{3N} v_i F_i + 2\gamma \left(\frac{3N}{2} k_B T_{\text{ref}} - E_k \right). \quad (6)$$

Let us look to the different terms from (6). The first term corresponds to minus the time derivative of the potential energy; the second term is an additional term describing the global coupling to the heat bath. In terms of temperature this extra term looks like the following

$$\left(\frac{dT}{dt} \right)_{\text{bath}} = 2\gamma (T_{\text{ref}} - T). \quad (7)$$

We can remark that the time constant τ_T equals $(2\gamma)^{-1}$. Returning to (1), it can be noted that the global additional temperature coupling (7) is accomplished by

$$m_i v_i' = F_i + m_i \gamma \left(\frac{T_{\text{ref}}}{T} - 1 \right) v_i. \quad (8)$$

Without adding local stochastic terms, from (8) it follows that

$$\frac{dE_k}{dt} = \sum m_i v_i v_i' = \sum_{i=1}^{3N} v_i F_i + 3N \gamma k_B (T_{\text{ref}} - T). \quad (9)$$

This is equivalent to (6). In this way we come to (8) as our new equation of motion. This represents a proportional scaling of the velocities per time step in the algorithm from v to λv with (to first order)

$$\lambda = 1 + \frac{\Delta t}{2\tau_T} \left(\frac{T_{\text{ref}}}{T} - 1 \right) v_i. \quad (10)$$

The change in temperature per step can also be made exactly equal to $(T_{\text{ref}} - T)\Delta t/\tau_T$, yielding

$$\lambda^2 - 1 = \frac{\Delta t}{k_{\text{Ber}}} \left(\frac{T_{\text{ref}}}{T} - 1 \right), \quad (11)$$

where

$$k_{\text{Ber}} = \frac{0.5 * 3N k_B}{C_V \tau_T}, \quad (12)$$

$$C_V = N * c_V \quad (13)$$

is the heat capacity of the system.

2.2. Dissipative Particle Dynamics-Berendsen Thermostat. In what follows we present an outline of the relevant theory for combined DPD-Berendsen thermostat. For an exhaustive presentation of the combined thermostat and critical tests we refer to [7]. Let us summarize below this theory.

The main idea of the DPD-like friction and noise is that they are applied to relative velocities between pairs of particles. Theoretically, if the pairwise application is well resolved it should ensure the conservation of total linear momentum. The friction and noise can be applied isotropically to the velocity difference vector. This is a simple and easy form for a DPD thermostat that will be used in this work. For DPD there is a velocity reduction factor $f = 1 - \exp(-\gamma\Delta t)$ and a damping rate factor γ . These factors depend on the interparticle distance: a cutoff distance should be chosen beyond which the impulsive friction and noise are not applied ($f = 0$).

In order to compute the forces of the system a pair particle list is obtained. Then, with this list the interparticle distance are determined. In DPD the dumping factor is scaled with a factor which depends on the distance between two pairs of particles; the original DPD chooses a linear bound between 1 ($r = 0$) and 0 ($r = r_c$).

A first substep is the selection of a pair of particles that will be subject to the impulsive friction and noise. This pair selection can be done in several ways. The selection can be made at random, but it can also be based on a distance-weighted probability, for example, proportional to $1 - (r_{ij}/r_c)$. A selected pair can be subject to friction and corresponding noise, with the friction isotropically in the direction of the velocity itself (iso). In this paper we will restrict ourselves to one case (ISO)—see [7]; the simplest and most easy to be used algorithm from the three presented in that paper. In what follows we will call the algorithm DPD-B. In the following we will present the DPD-B algorithm.

Algorithm.

(I) For all particles

$$v_i = \lambda_B * v_i. \quad (14)$$

(II) For i and j simulation pair step do

(I.1) choose the velocity reduction factor $f = f_{iso}$,

(I.2) determine the velocity noise factor g

$$g = \sqrt{\frac{f(2-f)k_B T_{ref}}{\mu}}, \quad (15)$$

where

$$\mu = \frac{m_i m_j}{m_i + m_j} \quad (16)$$

is the reduced mass of the two particles,

(I.3) construct the relative velocity vector v

$$v = v_i - v_j, \quad (17)$$

(I.4) for DPD-B,

(I.4.a) choose 3 random numbers $\xi = (\xi_1, \xi_2, \xi_3)$ from a standard normal distribution (mean = 1, sd = 1).

(I.4.b) construct the vector

$$\Delta v = -f v + g \xi. \quad (18)$$

Proceed to step (I.5).

(I.5) Distribute the relative velocity change over the two particles:

$$v_i \leftarrow \lambda_B v_i + \frac{\mu}{m_i} \Delta v, \quad (19)$$

$$v_j \leftarrow \lambda_B v_j - \frac{\mu}{m_j} \Delta v. \quad (20)$$

In this way the velocities of the particles are updated while the total momentum $m_i v_i + m_j v_j$ is conserved.

The particle velocities are updated after each impulsive event, not at the end of each step. This is necessary, as a single particle may be involved in more than one pair event.

In the next section we will shortly outline the Gromacs tool.

2.3. Gromacs. Gromacs (Groningen Machine for Chemical Simulations) is a molecular dynamics package primarily designed for simulations of proteins, lipids, and nucleic acids at molecular level. It implements the Newtonian equations of motions for different systems [4]. It was developed by the MD Group, Biophysical Chemistry department, University of Groningen, The Netherlands, and at the moment it is maintained by contributors from universities and research centers across the world. Gromacs is one of the fastest and most popular software packages available in the molecular dynamics field.

Gromacs is widely used in the Folding@home project for simulations of protein folding, which underlines the popularity and usage of this software [9]. The package itself does not just perform molecular dynamics simulations, but it is also capable of analyzing the output data [10]. It also provides calculation progress, a trajectory viewer, and an extensive library for trajectory analysis.

2.4. Implementation Details. There are several innovative elements related to the implementation of the new algorithm that results in reduced complexity and more efficient usage of computational resources. As explained earlier the implementation was done in Gromacs. The choice for Gromacs is motivated by the fact that it is open-source software, worldwide used, and known. Second, it was originating from the Molecular Dynamics Group which participates in the research described here.

A first innovative element is the use of a random selection of one neighbor per particle for the DPD component. This is possible through the physicotheoretical design of the algorithm. The selection can be made at random, but it can also be based on a distance-weighted probability, for example, proportional to $1 - r_{ij}/r_c$. It should be noted that traditionally [11] DPD algorithms use all-neighbors particles

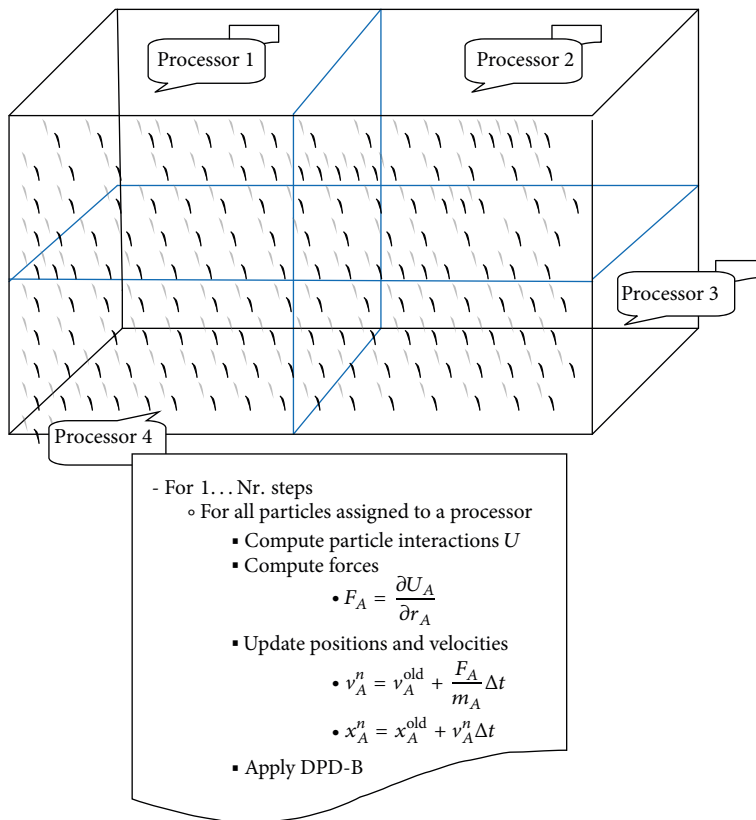


FIGURE 1: Domain decomposition. Simulation box is decomposed in small boxes assigned for different processors.

(in a given radius) of a particle. In our case just one neighbor is selected. This (as the results section will show) will result in a significant efficiency improvement.

Parallelization scheme is based on domain decomposition which means that the simulation box is divided in small boxes that are assigned to one processor each. This model of parallelization has the advantage that messages that are exchanged for communicating different information needed for the simulation happen mostly between neighbor processors (in number of nine). As a consequence the amount of messages is usually constant per processor and therefore scalability is almost linear per number of processors. Figure 1 presents the decomposition domain of the simulated box for four processors and outlines a short preview of the algorithm executed by each processor.

Another innovative element that is introduced for the parallelization of the DPD-B algorithm is the fact that neighbor-particle list is restricted to the particles that are computed by one processor. We will show in the results section that this choice preserves system properties. The fact that particles will be selected at random from the neighbor particles assigned to a processor makes no need to extra communication messages, thus reducing overall computation complexity. This fact is underlined in Figure 2.

3. Results

First we will discuss the efficiency gain obtained through the random selection of a pair of particles as compared to

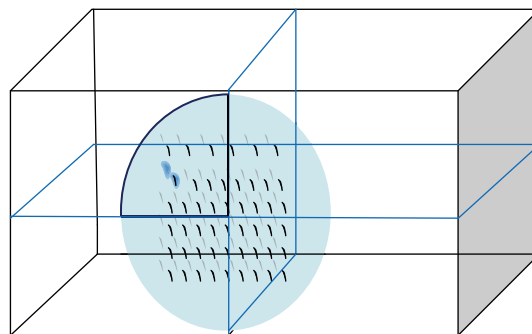


FIGURE 2: Pairs of particles are randomly selected only from the neighbor-list particles assigned to a processor.

the situation of using all-neighbor lists. The simulations are done on a molecular system of Coarse Grained Martini [12] Water with 3200 particles at a reference temperature of 320 K, with the time step of 2 fs. The implementation was made on Gromacs version 4.0.7. The temperature coupling factor for the Berendsen thermostat is 1.0 while for DPD-ISO it is 1.0. The system was tested on a dual-core machine with processor type Intel (R) Core (TM) 2 Duo CPU T6600 with 2.20 GHz and 4.00 GB installed memory (RAM).

For DPD-B, we compared the random selection of one pair of particles with the usage of all particles from the neighbor list. We also reported the times for existing thermostats from Gromacs such as Berendsen, Langevin type,

TABLE 1: The speed is reported as a number of nanoseconds simulated within 24 h.

Method	ns/day
Berendsen	34.63
No Thermostat	34.69
DPD-B, all neighbour list	6.94
DPD-B, random selection of pairs	29.31
Langevin	30.01

and no thermostat. In Table 1, the “speed” is reported as the number of nanoseconds simulated per day (24 h). From these efficiency experiments, we can conclude that our new algorithm performs better than the existing algorithms from the literature. Random selection gives a 5-fold efficiency gain to apply friction and noise to only one random pair per particle and its execution time is comparable with Langevin type. We made 8 independent runs for each type of thermostat and the numbers below report the averages.

For comparing molecular properties for the parallelization-optimization of using only particles pairs assigned to a processor (see Section 2.4), we used the same water system and we simulated on serial (1 processor) and parallel on 2 processors for 2 ns on the same machine. Analysis was done with standard Gromacs tools. In Figure 3 we show the computed temperatures for the 2 different runs—serial and parallel. It can be seen that the results are almost identical. Average temperatures are of 319.90 K for the parallel and 319.92 K for serial, both very close to the reference temperature of 320 K. Pressures were of 501,332 bars for the parallel and 502,008 bars for the serial, again similar. Diffusion was of $1.98e - 5 \text{ cm}^2/\text{s}$ for serial and of $1.94e - 5 \text{ cm}^2/\text{s}$ for parallel. It can be concluded that properties are similar and preserved by the optimization.

Below we represent the computational speed-up for the Martini CG water system. The simulations were done on a machine with 4 processors, of type Intel(R) Core(TM) i7 CPU 920 with 2.67 Hz and 2 Gb of memory per core. As it can be seen in Figure 4 the speed up is almost linear with the number of cores.

At the end we show how the temperature vanishes to 300 K for a starting gradient temperature condition. The box is split in seven zones; temperature is being increased symmetrically from 310 K (the first and the last zone) to a pick of 340 K (the middle zone); see Figure 5.

We performed the simulations for 1 processor (serial case), 2, and 4 processors. The measured temperature per each zone is within the normal statistical error for each case (see Figure 6).

4. Conclusions

The purpose of this paper is to present the MPI parallelization of a new molecular dynamics algorithm DPD-B that resulted from the combination of a dissipative particle dynamics (DPD) algorithm and Berendsen (B) thermostat. It should be taken into account that molecular dynamics simulations are time consuming. In some cases the running time varies from

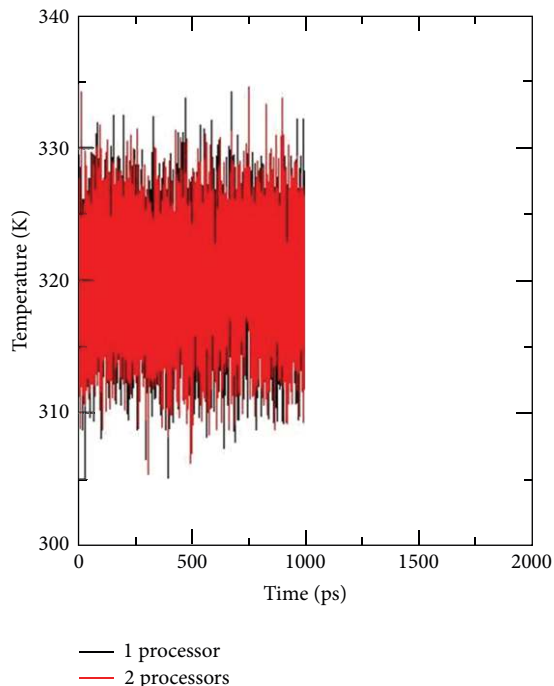


FIGURE 3: Temperatures for the runs of a Martini Coarse Grained System on serial and parallel.

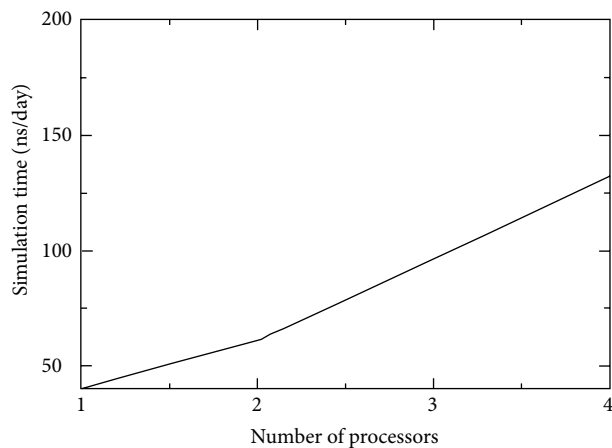


FIGURE 4: Speed-up function of a number of processors.

days to weeks and even months. Therefore, parallelization is one solution for reducing the execution time.

The paper describes the new algorithm, the main characteristics of the MPI parallelization of the new algorithms, and simulations performances. The work was done through research collaboration between Molecular Dynamics Group, University of Groningen, one of the well-known groups in the MD domain and researchers from Politehnica University of Bucharest. The parallelization was done in Gromacs [3, 4], an open-source software from this domain, used worldwide in universities and companies.

Several innovative elements related to the algorithm implementation are reported in the paper. A first innovative

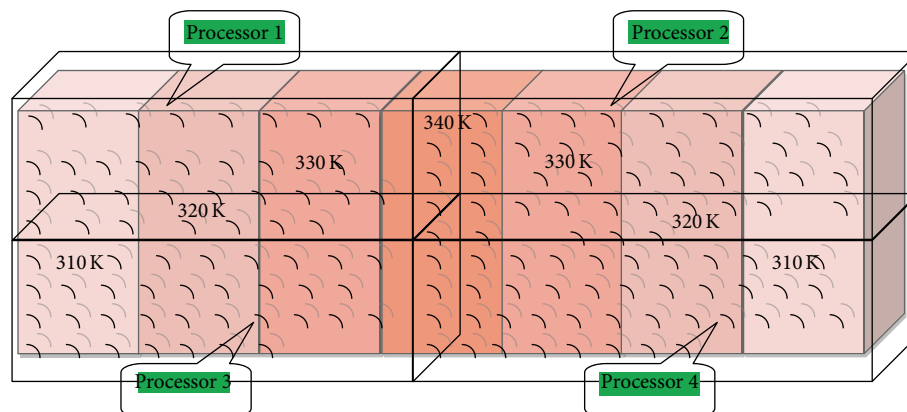


FIGURE 5: Gradient of temperature.

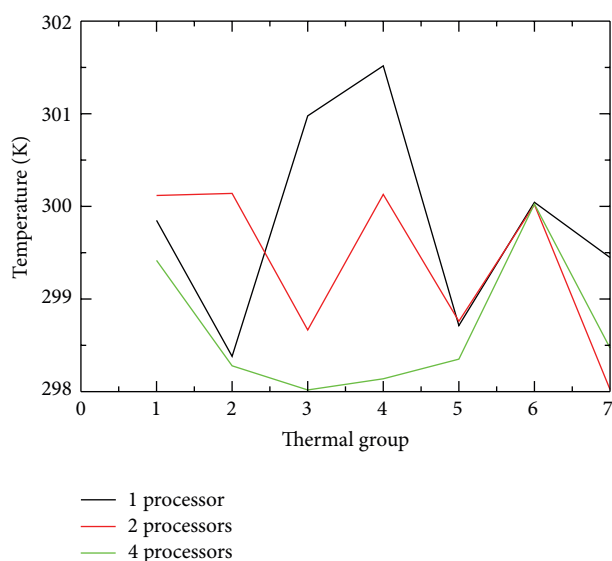


FIGURE 6: Temperature per thermal zone (group) after temperature gradient is let to vanish to 300 K, for 1, 2, and 4 processors. All temperatures are within statistical errors.

element is the use of a random selection of one neighbor per particle for the DPD component. The selection can be made at random, but it can also be based on a distance-weighted probability, for example, proportional to $1 - r_{ij}/r_c$. It should be noted that traditionally [11] DPD algorithms use all neighbors (in a given radius) of a particle. In our case just one neighbor is selected. This, as the results section shows, results in a significant efficiency improvement, till 5 times more than traditional algorithms from the literature.

Another innovative element that is introduced for the parallelization of the DPD-B algorithm is the fact that neighbor-particle list is restricted to the particles that are computed by one processor. We show in the results part that this choice preserves system properties. The fact that particles are selected at random from the neighbor particles assigned to a processor makes no need to extra communication messages, thus reducing the overall computation complexity.

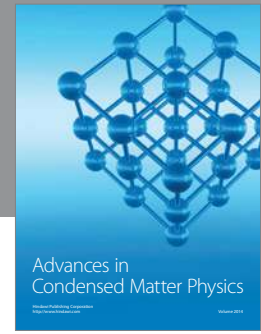
When looking on how the temperature vanishes for a starting temperature gradient condition, it can be observed

that the measured temperature is within statistical errors for the cases of 1, 2, and 4 processors.

As future work we intend to implement more new DPD algorithms.

References

- [1] N. Goga, A. J. Rzepiela, A. H. de Vries, S. J. Marrink, and H. J. C. Berendsen, "Efficient algorithms for Langevin and DPD dynamics," *Journal of Chemical Theory and Computation*, vol. 8, no. 10, pp. 3637–3649, 2012.
- [2] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. Dinola, and J. R. Haak, "Molecular dynamics with coupling to an external bath," *The Journal of Chemical Physics*, vol. 81, no. 8, pp. 3684–3690, 1984.
- [3] Gromacs tool and documentation, <http://www.gromacs.org/>.
- [4] D. van der Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, and H. J. C. Berendsen, "GROMACS: fast, flexible, and free," *Journal of Computational Chemistry*, vol. 26, no. 16, pp. 1701–1718, 2005.
- [5] N. Goga, A. J. Rzepiela, M. N. Melo et al., *Advances in Planar Lipid Bilayers and Liposomes: (APLBL)*, vol. 15, Academic Press, Elsevier, 2012.
- [6] N. Goga, S. Costache, and S. Marrink, "A multiscaling constant lambda Molecular dynamic Gromacs implementation," *Materiale Plastice*, vol. 46, no. 1, pp. 53–57, 2009.
- [7] S. A. Moga, N. Goga, and A. Hadar, "Combining berendsen thermostat with dissipative particle dynamics (DPD) for polymer simulation," *Materiale Plastice*, vol. 50, no. 3, 2013.
- [8] H. J. C. Berendsen, "Transport properties computed by linear response through weak coupling to a bath," *Computer Simulations in Material Science*, vol. 205, pp. 139–155, 1991.
- [9] Folding@home, <http://folding.stanford.edu>.
- [10] D. van der Spoel, E. Lindhal, B. Hess et al., Gromacs user manual version 3. 3., 2006.
- [11] J. M. V. A. Koelman and P. J. Hoogerbrugge, "Dynamic simulation of hard sphere suspensions under steady shear," *Europhysics Letters*, vol. 21, no. 3, pp. 363–368, 1993.
- [12] S. J. Marrink, H. J. Risselada, S. Yefimov, D. P. Tieleman, and A. H. De Vries, "The MARTINI force field: coarse grained model for biomolecular simulations," *Journal of Physical Chemistry B*, vol. 111, no. 27, pp. 7812–7824, 2007.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

