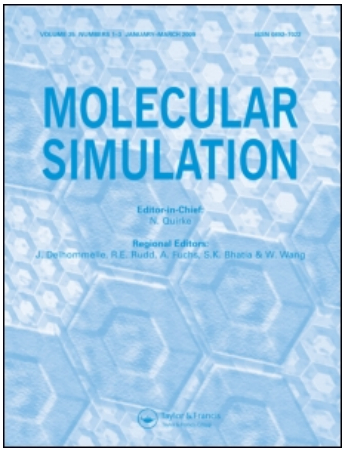

Molecular Simulation
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713644482

Using Discrete Particles As a Natural Solver In Simulating Multiple-Scale Phenomena

W. Dzwinel[a]; W. Alda[a]; J. Kitowskia[ab]; D. A. Yuen[c]
[a] Institute of Computer Science AGH, Kraków, Poland [b] ACK CYFRONET, Kraków, Poland [c] Minnesota Supercomputing Institute, University of Minnesota, Minneapolis, USA

To cite this Article Dzwinel, W. , Alda, W. , Kitowskia, J. and Yuen, D. A.(2000) 'Using Discrete Particles As a Natural Solver In Simulating Multiple-Scale Phenomena', Molecular Simulation, 25: 6, 361 — 384
To link to this Article: DOI: 10.1080/08927020008028167
URL: http://dx.doi.org/10.1080/08927020008028167

Molecular Simulation, 2000, Vol. 25, pp. 361–384
Reprints available directly from the publisher
Photocopying permitted by license only

USING DISCRETE PARTICLES AS A NATURAL SOLVER IN SIMULATING MULTIPLE-SCALE PHENOMENA

W. DZWINEL[a,*], W. ALDA[a], J. KITOWSKI[a,b] and D. A. YUEN[c]

[a]*Institute of Computer Science AGH, Al. Mickiewicza 30, 30-059 Kraków, Poland;*
[b]*ACK CYFRONET, ul. Nawojki 11, 30-950 Kraków, Poland;*
[c]*Minnesota Supercomputing Institute, University of Minnesota, Minneapolis, USA*

(Received November 1999; accepted November 1999)

We propose here the usage of different discrete particle methods for simulating both mesoscopic and macroscopic phenomena with hierarchical structures. Results of large-scale MD simulations of the Rayleigh–Taylor instability are shown and discussed. Using about one million Lennard–Jones particles, we have simulated particle fluid environments with a length scale of about 0.5 μm. For encompassing greater length and time scales, either more complex particle method or simplified MD model must be used. In the first case, we study dissipative particle dynamics (DPD) in modeling complex multi-component fluids flows, *e.g.*, bubble formation, fluid flows in porous media, mixing driven by sedimentation. We also demonstrate how the particle method can be utilized for simulating a flexible surface with multiple scales present. We emphasize that the inherent parallelism of the MD method makes this approach a powerful natural solver of a wide variety of physical phenomena with multiple scales.

Keywords: Discrete particles; multiple scales; Rayleigh–Taylor instability

1. INTRODUCTION

According to the laws in physics, particles interact with one another by an exchange of virtual objects, *e.g.*, photons in electrodynamics and phonons in solid-state physics. Changes in the physical states of particles, *i.e.*, their positions, momenta, spins *etc.*, result from their mutual interactions. In this

*Corresponding author.

century this atomistic approach plays an important role in the construction of physical models.

A virtual particle (VIP) [1, 2] is the basic element of the particle based computational model. VIP can be defined at various levels of abstraction [2] *e.g.*, atom, particle, cluster of particles, vehicle-target-obstacle, genotype, multidimensional point, UNIX process, single processor, *etc.* For example, taking into account that UNIX processes can "interact" by sending and receiving messages, we can easily imagine about the direct transformation of the VIP model into the message-passing model of parallel computations. This would involve the change of the VIP level of abstraction from the particles to the processes in exchanging messages.

The main suggestion put forward in [1, 2] consists in the elaboration of a new strategy for parallelizing an application, by using two stages of numerical mapping (see Fig. 1). First, a problem is transformed into one of the natural solvers (or their hybrid) and virtual particles are defined. Then the method is implemented on a multicomputer system by transforming of virtual particles into a virtual parallel machine model [1]. Widely used natural solvers such as: molecular dynamics, lattice gas, Boltzmann lattice gas, simulated annealing, direct Monte-Carlo, cellular automata, genetic algorithms, neural networks and others, can be treated as particle methods according to the definition presented in [2]. All these solvers, have been used in the fields of physics, chemistry, biology and geology. Therefore, the second stage of mapping (*i.e.*, its implementation on a multiprocessor

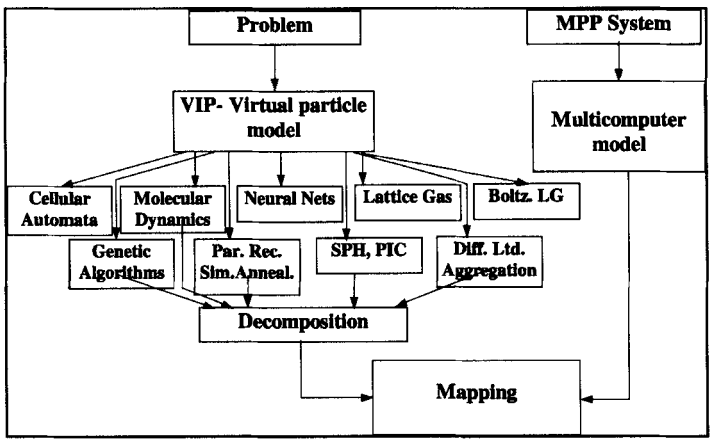

FIGURE 1 Problem of mapping onto the multiprocessor model by transforming to a natural solver involving discrete particles [1] (where: SPH – smoothed particle hydrodynamics, PIC – particle in cell, MPP – massively parallel processor).

architecture) often allows us to exploit the ready-to-use parallel algorithms or at least existing knowledge about the ways for parallelization of the particle method. This sort of mapping demands a creative and abstract way for modeling of complex multiple-scale phenomena on current and future generations of computer systems.

Molecular dynamics method (MD) – a well known technique of computational physics and one of the Grand Challenges of Science [3] problems – can be regarded as a pure particle paradigm. Our goal is to show that MD can be treated as a natural solver, *i.e.*, a universal paradigm, whose principles come from natural laws and which can be used as a solver in various fields of science and engineering. Due to their inherent parallelism because of the short-range potentials, MD and other natural solvers constitute the class of powerful computational tools, when they are empowered by a parallel system. The increasing interest in the implementation of these techniques on multiprocessor systems follows naturally from this scaling property in the parallel architecture.

First, the mathematical background and computer realization of the MD method are briefly discussed. Then we present representative samples of MD applications in large-scale simulations of the Rayleigh–Taylor instability, which represents a fluid-dynamical phenomenon with hierarchical, multiple-scale structures and scaling property. We show also the results from simulations, which were obtained by using dissipative particle dynamics method, which is a mesoscopic counterpart to MD. In the following section we demonstrate that a simplified computer realization of the MD method can be used as an efficient animation technique. Since the visual impression of movement plays the principal role in visualization, some physical details can be substantially simplified. Finally, we discuss the conclusions of this paper.

2. THE PRINCIPLES OF MOLECULAR DYNAMICS (MD)

From over than 30 years molecular dynamics (MD) has been a widely-used computational technique (*e.g.* [4]). Its basic principles are outlined in Figure 2.

Each particle i interacts with all others located in a sphere with R_{cut} radius according to potential energy of interactions. In the simplest case two-body pair radial potential function $\phi(r_{ij})$ depends on the distance r_{ij} between the particles. For more complex molecules with internal degrees

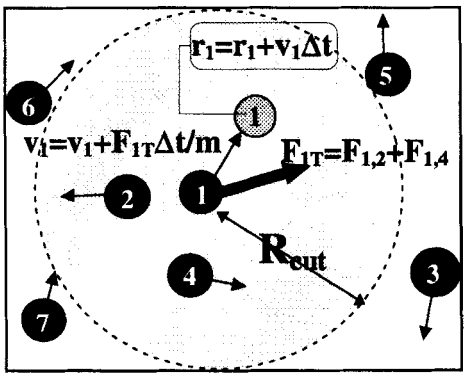

FIGURE 2 Basic principles of the MD paradigm.

of freedom, the potential function can be more sophisticated. Let the pair force $\mathbf{f}_{ij} = -\nabla\phi(r_{ij})$, while the total force $\mathbf{F}_i$, which acts on a single particle i, is the sum of pair forces $\mathbf{f}_{ij}$ of its neighboring particles within the R_{cut} sphere.

The temporal evolution of the particles, $i = 1, \ldots, M$, in the presence of gravitational and friction forces is described by the Newtonian momentum equations, which can be represented in the following discrete form:

$$\begin{aligned} \Delta\mathbf{p}_i &= \sum_{i \neq j} \left(\Omega_{ij}\mathbf{e}_{ij} + m_i g \cdot \mathbf{e}_y - \lambda_i \mathbf{p}_i\right) \cdot \Delta t \\ \Delta\mathbf{r}_i &= \frac{\Delta t}{m_i}\left(\mathbf{p}_i + \Delta\mathbf{p}_i\right) \\ r_{ij} &= |\mathbf{r}_i - \mathbf{r}_j| \end{aligned} \tag{1}$$

where $\Delta\mathbf{r}_i$, $\Delta\mathbf{p}_i$ – are the changes of position and momentum, respectively, for particle i of mass m_i, Δt – timestep, Ω_{ij} – collision operator, g – gravitational acceleration, λ – friction coefficient, $\mathbf{e}_{ij}$ – unit vector pointing from particle j to particle i, r_{ij} – the distance between particles i and j. For MD $\Omega_{ij} = \mathbf{f}_{ij}$ is derived from a short-range two-body potentials (*e.g.*, 6–12 Lennard–Jones) or more complicated many-body potentials (*e.g.*, Stillinger–Weber three-body potential). The parameters of these interactions are usually fit to the specific materials, comparing experiment and MD simulation results using empirical data (*e.g.*, see [5]). The potential functions and their parameters can be also obtained employing *ab initio* procedures. The computer implementation of MD techniques consists of a sequential calculation of the forces and particle movements at each time step.

A set of simulated particles is placed in a rectangular box with periodic boundary conditions (PBC). The number of particles, M, is limited by the computational power ($M = 10^9$ on the fastest parallel systems [6]). In the real world, one mole of liquid contains 10^{26} molecules. PBC enables us to mimic the large number by using a few molecules. However, this assumption works well only for time scale limited by the size of computational box divided by sound speed in the medium being simulated. Because the size of box depends on M, to get more accurate results of phenomena under investigation, larger samples of molecules should be taken into account. Assuming that a molecule may consist of hundred and thousands of atoms, simulating a system consisting of such molecules is much more difficult than, for example, a simple one-atom-molecule system of liquid Ar. The evolution of very large number of particles $O(10^{10})$ simulated in longer and longer time scales up to 500 nanosecs represents a grand challenge for the fastest computers in the year 2000. Therefore, serious research efforts have been going on for years now on implementing MD codes on high-performance computers [6].

For parallel implementation of MD method, a geometric decomposition is usually used. In Figure 3 we can see typical decomposition of the computational box for distributed computations on the ring of workstations (Fig. 3a) and for parallel processing on tightly coupled massively parallel processors (MPP), such as the shared memory architecture (Fig. 3b).

As shown in [6], the progress in hardware and software development has led to an increase in the number of atoms, simulated by MD codes from hundreds in the seventies to billions in the mid nineties [11]. The parallel MD codes can reach 95% efficiency on hundreds of processors. Vast

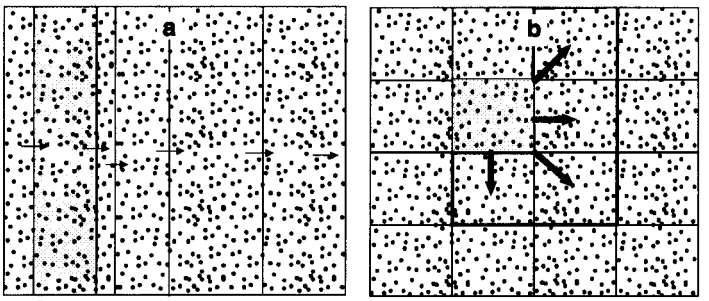

FIGURE 3 Two approaches for MD domain parallelism. The arrows show directions of information exchange between a domain (shaded) and its neighborhood. For (a) the load balancing is realized by changing the strips width, while for (b) it is more fine grained but complicated.

amount of literature and MD software for the full spectrum of vector and multiprocessor architectures is now available (see the overview paper [6], for example). From this point of view, the MD method fulfills the fundamental criterion, which a natural solver should fulfill. However, the most relevant feature of natural solvers lies in their universality.

3. LARGE-SCALE MD SIMULATIONS OF MULTIPLE-SCALE PHYSICAL PHENOMENA

Classical MD simulations describe the microscopic O(100 Å), short-time phenomena $O(10^{-9}\,\text{sec})$ in liquids and solids. Due to the time and spatial averaging of stochastic functions and variables, one can obtain integral and/or differential parameters of a medium investigated. By fitting results of simulation to the experimental and theoretical values, one can find the proper model of molecules and/or potential energy of the interacting particles. All these phenomena occur in the microscopic world, which apparently limits the fields of potential application of the MD method.

The first MD experiments [7, 8] in which collective movements of simple Lennard–Jones particle ensembles were found, showed that, even for a relatively small number of particles in short-time simulations, it was still possible to observe the striking resemblance of patterns created in both the micro- and macroscale. By increasing the number of particles to millions and billions, it is now possible to simulate the phenomena in the mesoscale, *i.e.*, where the size of sample is 1 μm of order and simulation time reaches up to tens of nanoseconds. The particle fluid flows [8, 9], crack propagation [10], nanostructured materials [11] and the development hydrodynamical instabilities [12] can be simulated. Such investigations are important in which classical models based on continuum equations in hydrodynamics (*e.g.*, the Navier–Stokes equations) are found to be insufficient and the assumptions of continuity are violated. The same applies to the description of phenomena with their origins in the microscale and capable of being resolved in the macroscale. In simulating them by the classical continuum models, artificial fluctuations are introduced. Such an approach loses the information concerning the initial stage of the mixing process, its causality and the start-up time.

The 2-D numerical results of the Rayleigh–Taylor instability in the macroscale, using MD parallel code, can be found in [12]. The computer experiment consists in the simulation of mixing of two equal layers filled with particles. The first layer consists of heavy particles and the second

one – placed below – is made up of lighter particles. The gravitational field, which is directed from the heavy layer to the lighter one, renders the system mechanically unstable. Due to the statistical fluctuations the two fluids begin to mix. This type of instability belongs to the most difficult situation in simulations with classical fluid-dynamics (FD) codes. In particular, the initialization is not investigated yet in detail. Because of the lack of causality factor in the classical equations of fluid dynamics, artificial disturbances must then be introduced.

As shown in Figure 4, the evolution of mixing process by using MD code is similar to that observed in experiments and those obtained from simulations using FD codes. Different from FD simulations, the instability process develops spontaneously, *i.e.*, not initialized artificially. The

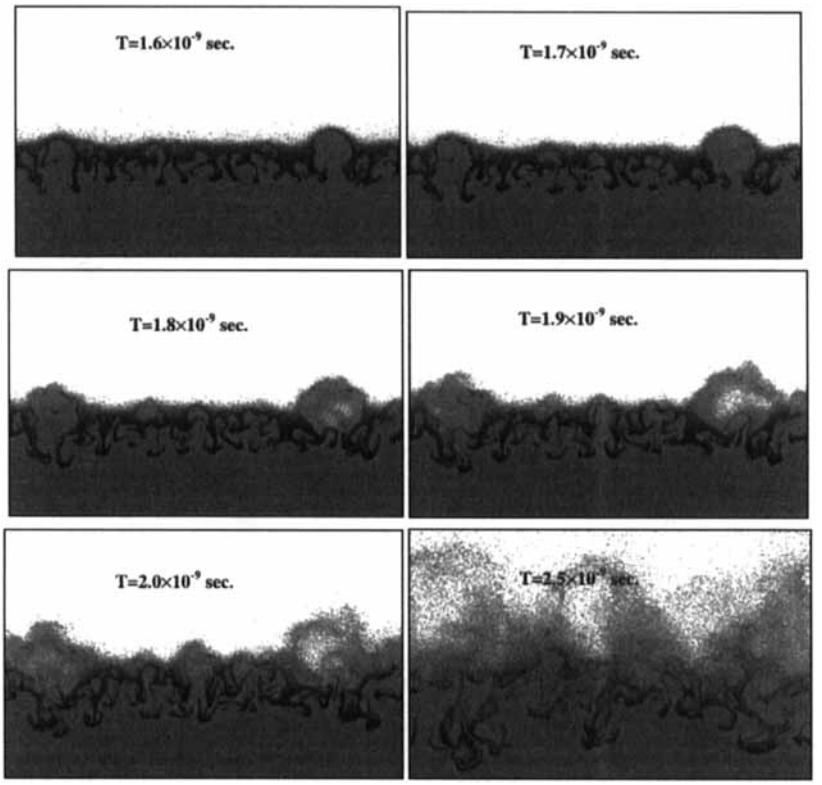

FIGURE 4 Snapshots from the Rayleigh–Taylor instability simulation. One million particles in 3×10^5 timesteps were simulated. The gray-shades show the particles density. Simulation was performed using MD parallel code in PVM environment on 48 CRAY T3E processors. Free surface is simulated self-consistently by leaving the box half-filled initially.

fluctuations represent the real causality factor, lacking in the FD models. Moreover, the particle approach overcomes many difficulties with boundary conditions present in the FD models. Taking advantage of these, we can investigate more thoroughly the temporal evolution of the mixing layer for liquids closed in a container or particle system with a free surface (see Fig. 4).

Assuming that both upper and lower fluids thickness are the same, the mixing layer for closed container boundary conditions grows as $A \cdot t^2$ with the same value of A as that obtained for infinitely thick liquid layers in the macroscale experiments [12]. However, for the free surface case, t^2 rule is not exactly fulfilled. For example, two mixing regimes can be distinguished in Figure 5 for this case. The first one is observed at the beginning of the process, when only thin boundary layers of two liquids take part in mixing. The process changes in character in the second regime. The mixing is rapidly accelerated by the break-up of the surface. In the case of thicker

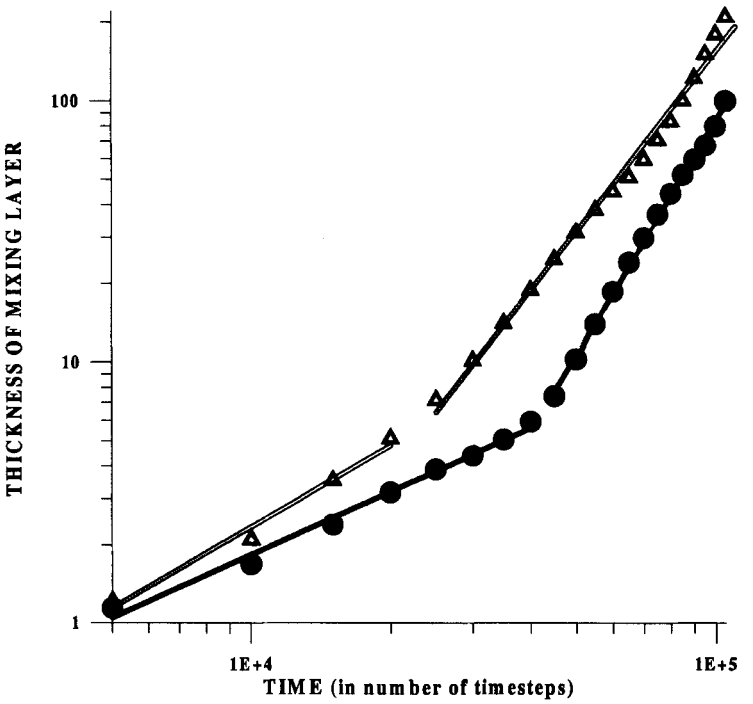

FIGURE 5 The growth of mixing layer (in dimensionless units) for two different thickness of the heavy fluid. The upper diagram corresponds to the situation, when the heavy fluid layer is two times thinner than that in the lower plot.

heavy layer, mixing is slower in the first regime while faster after the break-up, due to the higher upward pressure (see Fig. 5). We did not find any theoretical explanation why the mixing for the free surface case can be described by power laws, that suggest the fits made in Figure 5.

We can conclude on the basis of the resemblance of the simulation results of similar processes in the micro- and macroscales that by rescaling, changing the definition of a particle and interparticle potential, we can employ the MD model for simulating physical phenomena in both the meso- and macroscales [13].

4. DISSIPATIVE PARTICLE DYNAMICS IN COMPLEX FLUIDS SIMULATION

We can extend the rescaling concept and thus interpret the particle as a "lump of fluid". The problems with defining the interparticle potential can be overcome by using the dissipative particle dynamics method (DPD) [14], which is a mesoscopic counterpart to MD. Let us assume that a system of mesoscopic particles can interact *via* direct conservative potentials, as in MD simulations but, additionally, the particles exert friction and Brownian forces on each other. The collision operator can be defined as follows:

$$\Omega_{ij}^n(r_{ij}^n, \mathbf{p}_{ij}^n) = \left[\alpha_{ij} - \gamma\omega_{ij}^n(\mathbf{e}_{ij}^n \cdot \mathbf{p}_{ij}^n) + \frac{\sigma\theta_{ij}}{\sqrt{\Delta t}}\right]\omega_{ij}^n$$
$$\omega_{ij}^n = 1 - \frac{r_{ij}^n}{R_{\text{cut}}} \tag{2}$$

where R_{cut} – cut off radius; α_{ij}, γ_{ij}, σ – scaling factors for conservative, dissipative and Brownian interactions, respectively; θ_{ij} – a random number from $[-1, 1]$ interval; $\mathbf{p}_{ij}$ – difference between momenta of particle i and j; n – timestep number. The collision operator for the three types of particle models is short-ranged, *i.e.*, for $r_{ij} > R_{\text{cut}}$ $\Omega_{ij} = 0$.

The dissipative and random interactions are chosen in such a way that the center of mass motion of each interacting pair is insensitive to their overall motions. In this way the system relaxes to its equilibrium much faster than in MD simulations and, at the same time, the interaction conserves the total momentum. The second feature allows the system to behave hydrodynamically over large length scales. The DPD particle model is isotropic, Galilean-invariant, and has the potential to be computationally efficient.

We use in our simulations, multicomponent isothermal version of DPD method. Each specie $k = 1, \ldots, l, \ldots, M$ consists of particles, which have own "color", mass m_k and α_{kl}, γ_{kl}, σ_{kl} parameters in collision operator given by Eq. (2). They are chosen in such a way that:

$$\frac{M_{kl}\sigma_{kl}^2}{\gamma_{kl}} = KT \quad M_{kl} = \frac{2m_k m_l}{m_k + m_l} \tag{3}$$

where T is the absolute temperature of the whole particle system and K is the Boltzmann constant.

We assumed also that particles are placed in the box with periodic boundary conditions along x-axis and reflective along y-axis. The gravitational field is directed down the plate. Simulation time is given in dimensionless units in number of timesteps Δt, where Δt is set to 1.

The mapping DPD collision operator onto the microscopic interactions or the macroscopic properties of fluid constitutes a serious problem. It can be partially solved by using kinetic theory equations [15, 16] and Voronoi tessellation scheme [17]. In Figure 6 we display two snapshots from R–T instability simulations employing MD and DPD methods, respectively. The physical conditions for the both simulations are the same, *e.g.*, the sample size ($0.12 \times 0.48\,\mu$m), initial temperature (140 K), the Atwood number (0.38), fluid density. The gravitational acceleration is set to 10^{12} m/sec^2. We employ kinetic theory equations for matching the compressibility and

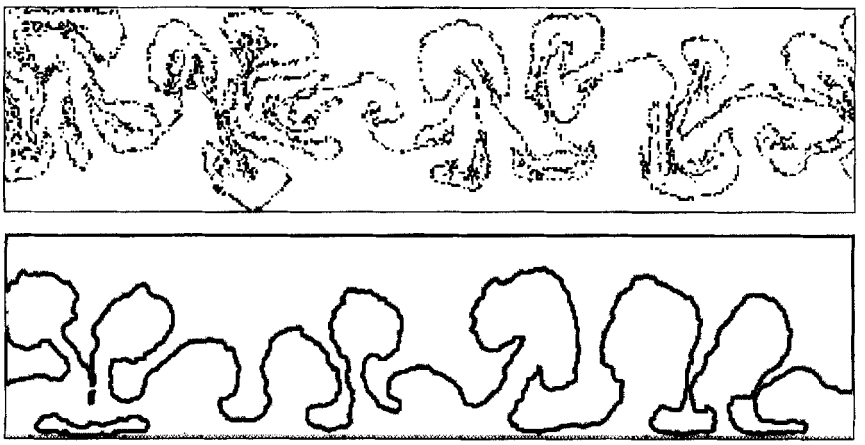

FIGURE 6 The snapshots from MD (above) and DPD (below) simulations of the R–T instability for the same moment of time ($t = 1.6 \times 10^{-9}$ sec). Only interfaces between two particle systems are shown.

kinematic viscosity of DPD fluid to the values obtained for the corresponding MD system (sonic speed $c \sim 850\,\text{m/sec}$, viscosity $\nu \approx 4.0 \times 10^{-7}\,\text{m}^2/\text{sec}$). The snapshots represent the same moment of time ($t = 1.6 \times 10^{-9}\,\text{sec}$). The number of atoms used in MD simulation is 7.5×10^5 but corresponding DPD system is made of 3×10^4 particles. The respective timesteps are $\Delta t = 2.0 \times 10^{-14}$ and $\Delta t = 2.0 \times 10^{-13}$. Therefore, the DPD method requires more than 100 times less CPU time than the average sized MD simulation but at the expense of lower spatial resolution.

In Figure 7 we display the process of bubbles formation in DPD fluid. The ambient fluid is more viscous and heavier than the thin layer of fluid placed below. The particle mass ratio of ambient and thin-film fluids is 3 : 1. The number of particles simulated is 1.6×10^5. The thin fluid layer consists of about 2×10^4 DPD particles. We can trace the spontaneous process of bubbles formation and breaking-up of spikes. The remaining thin layer of the fluid contracts and creates the subsequently bubble.

As shown in Figure 8, the DPD method allows one to simulate very complex flows in the mesoscale, which would be extremely hard-pressed in FD simulations. The release of the gas trapped in the porous solid to the ambient fluid represents a good example of such a phenomenon. This can be achieved by using fewer particles than for MD (here 5×10^5) and in considerably fewer timesteps (about 5×10^5).

The snapshots shown in Figure 9 present the process of mixing in the presence of sediment particles for closed container boundary conditions.

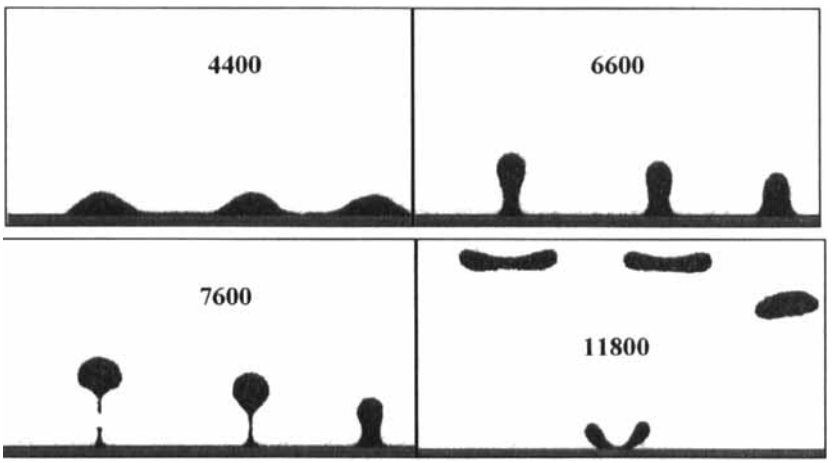

FIGURE 7 The snapshots from DPD simulation of the process of bubbles formation. The number of timesteps is shown.

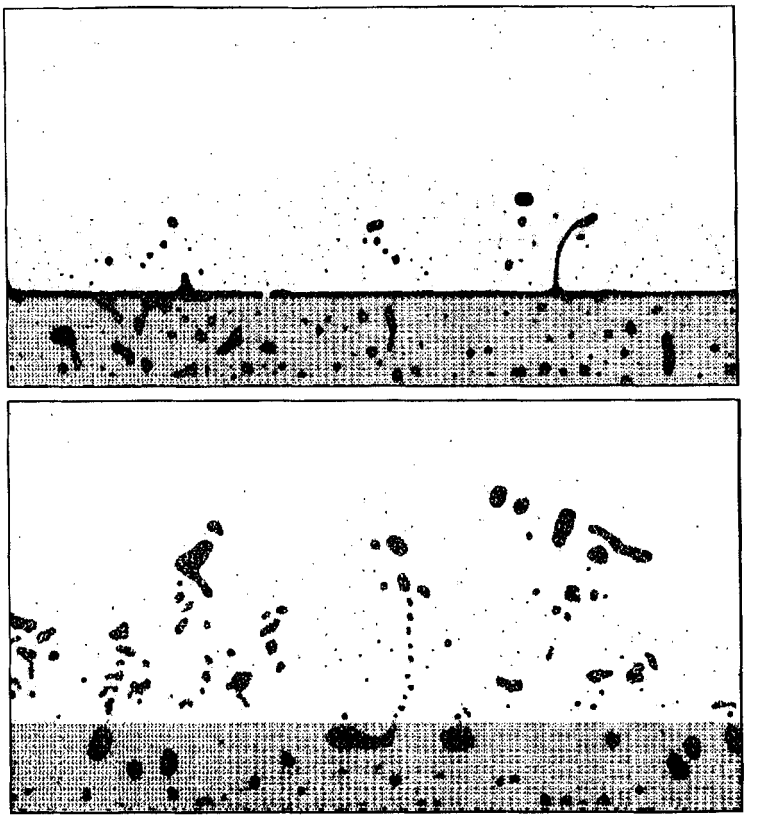

FIGURE 8 DPD simulation result of gas release process from the porous matrix. Porous surface is portrayed in light gray and an ambient fluid is white.

The fluid placed below is composed of two types of particles. The lighter particles stand for interstitial fluid and the heavy ones for sediment. Their mass ratio is $m_2 : m_1 = 10 : 1$. Initially, sediment particles were scattered randomly in the bulk of lighter fluid. The fluid placed above consists of particles, which mass m_3 is 5 times greater than m_1. The average density of upper and bottom systems is the same. The simulations were performed for the two types of boundary conditions, periodic (in x-axis direction) and in the closed container. In the first case, the mixing is more vigorous. After 5,500 timesteps the sediment particles are still moving upward, while for the closed container they begin fall down.

Because of the reflective walls, the sediment particles find themselves mainly in the center of the bottom, whereas the convection takes place close to the walls. In the both cases, the lighter, interstitial particles are squeezed out by the upper fluid.

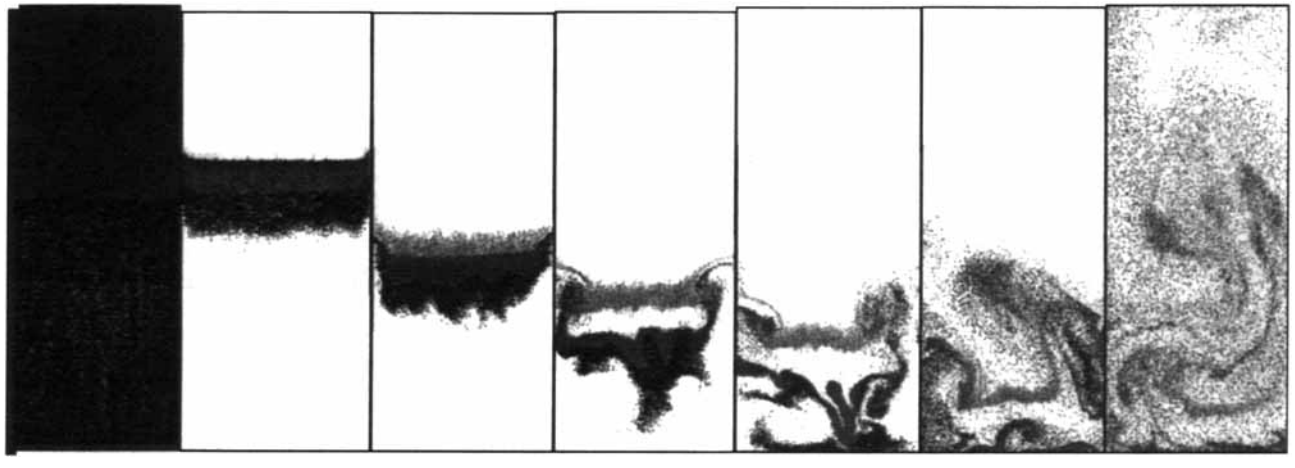

FIGURE 9 Mixing driven by sedimentation in vapor in the closed container. The first picture displays the initial conditions but the following show the temporal evolution of the interface layer. Each snapshot represents particle system after following $500 \times \Delta t$ time-steps, where $\Delta t = 1$. The number of DPD particles simulated is 5×10^5, the mass ratio – solvent (black) : heavy fluid (gray) : sediment (light gray) = 1 : 5 : 10.

The dissipative particle dynamics is a mesoscale technique like the well-known lattice-Boltzmann gas method (LBG) [18]. In comparison to the lattice gas automata (LGA), LBG reduces considerably the statistical noise. The same can be observed for DPD when confronted with MD method (see Fig. 6). Nevertheless, LBG is still tied to the inconvenience of a lattice, which makes the treatment of complex boundary conditions cumbersome. Moreover, LGA and LBG are irreversible algorithms and there is unknown detailed-balance condition, which means that the Gibbsian equilibrium state of the system is not guaranteed to exist.

In Figure 10 we display the snapshots from LBG and DPD simulations of phase separation in binary fluid. In spite of apparent similarity of pictures obtained, the two processes observed are different in character. As it is shown in [19, 20], for binary systems in the regime of sharp domain walls, time-dependent average domain size $R(t)$ follows algebraic growth laws of the form $R(t) \propto t^{\beta}$. LBG studies suggest that $\beta = 1/3$, but both MD and DPD find the same values $\beta = 1/2$ in the beginning and $\beta = 2/3$ at late times. However, the lattice-Boltzmann method does not include any thermal fluctuations, which most certainly play a crucial role in causing this exponent to assume the value of 1/2 [19]. We assume in both models very strong repulsive local interactions between DPD particles and LBG

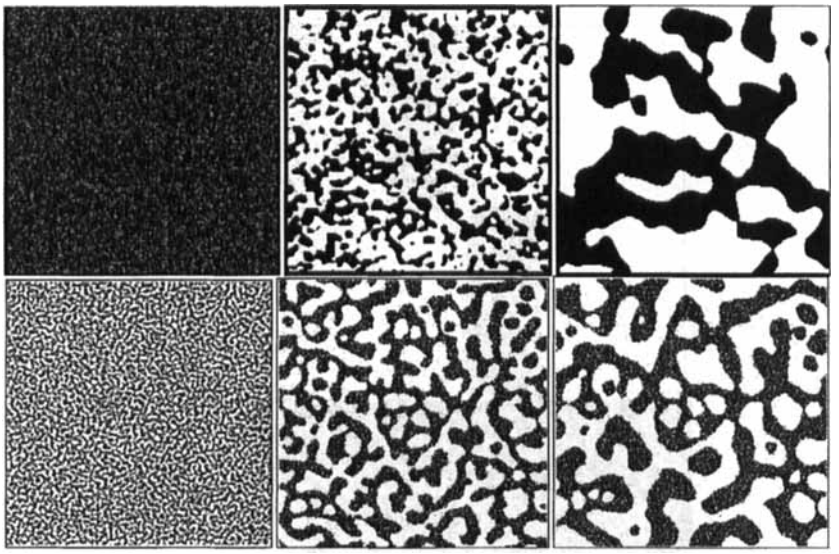

FIGURE 10 The confrontation of the snapshots from LBG (above) and DPD (below) simulations of the phase separation in a binary fluid.

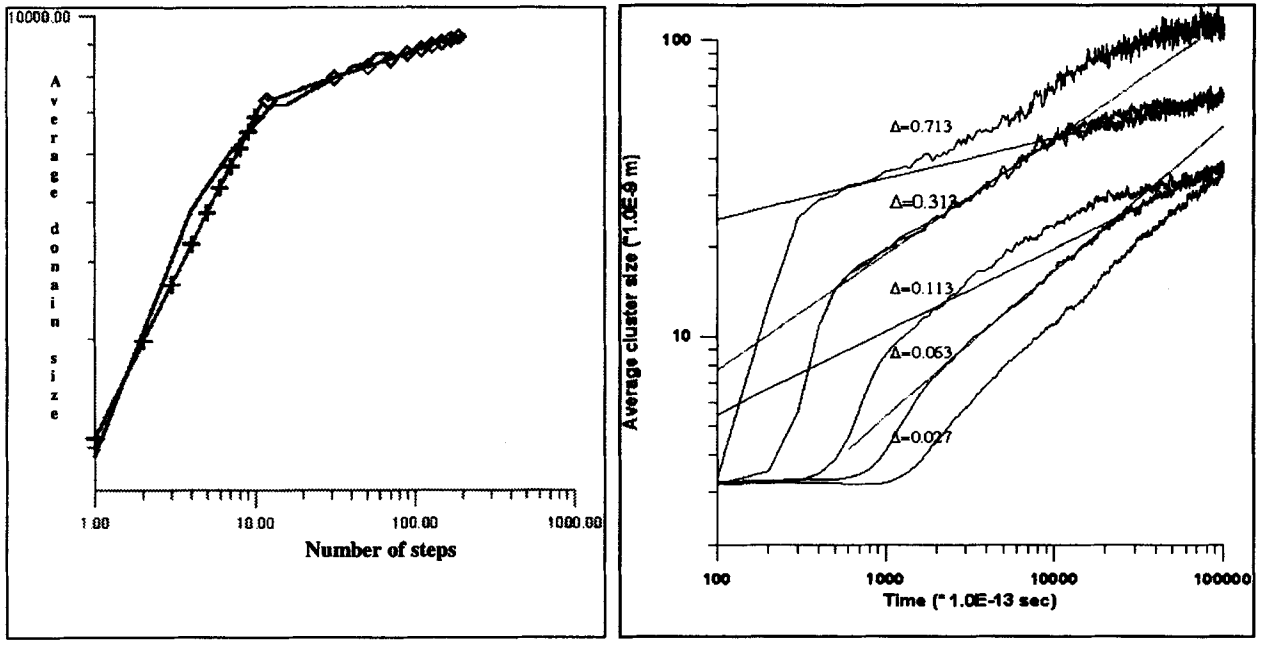

FIGURE 11 The log–log plots of average domain size (axis Y-dimensionless) *versus* number of steps (axis X) for LBG (on the left) and average domain size *versus* time for DPD (on the right). The process of phase separation for increasing immiscibility factor Δ in DPD becomes similar to that obtained by using LBG method.

sites of different types, respectively. In this case the thermal fluctuations contribution to the domain growth process is small.

As we see in Figure 11, both for LGA and DPD models we observe very fast domain growth at the beginning of simulations. Then for large immiscibility factor Δ in DPD simulations the β value becomes the same as in LBG, *i.e.*, 1/3.

Another approach is used in simulations of dynamics in granular media (*e.g.* [21]), where the particles have different shapes and interaction potential is very complex. Nevertheless, the "backbone" of all these models lies on the MD formulation and their rapid realization by means of the MD parallel algorithms.

We may expect, that by making the model more exact and realistic we can obtain eventually the results of MD simulations, which can model adequately laboratory experiments. However, we emphasize the universal character of MD that even for the simplest implementation of the MD

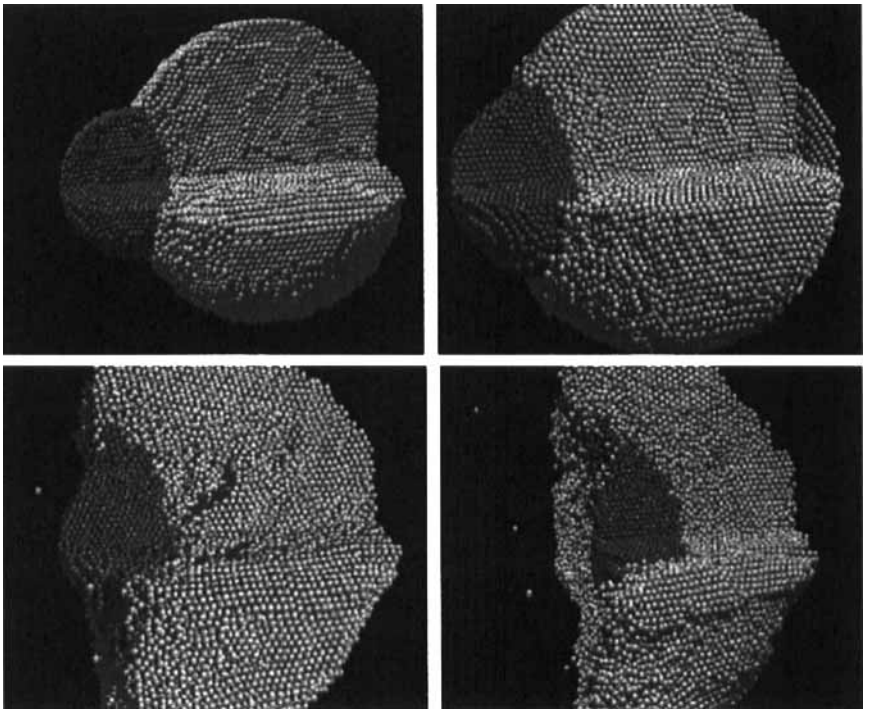

FIGURE 12 Two balls made of particles hitting one another in a 3-D MD simulation.

method, we have up to now obtained astonishingly good quality results. For example, some effects in granular dynamics, similar to these observed in the reality, can also be simulated using very simple MD algorithms (see Fig. 12). We can display them visually for a greater sense of reality.

Particle methods can also be combined with the continuum approach. For example the problems of sand transport in pipe or convection driven by sedimentation can be solved simulating particles temporal evolution in flow field generated by FD methods [22] (see Fig. 13). However, due to serious problems in matching the parameters of these two heterogeneous models [23], we propose in [24] combined MD, DPD and SPH (smoothed particle dynamics [25]) based approach, which encompasses micro, meso and macroscale features in one homogeneous particle model. The distinct advantages of this particle approach over the conventional FD methods, involving finite elements or finite differences, are evident. The most important factors are as follows:

1. simple and flexible computational model bridging multiple-scales,
2. gridless approach,
3. simple definition of discontinuities,
4. easy-to-write, efficient parallel codes,
5. minor problems dealing with complicated boundaries and internal inhomogenities.

5. METHOD OF PARTICLES USED IN VISUALIZATION

In visualization, an impressive visual representation means much more than accurate quantitative agreement with the reality. Assume for now, that we are going to animate a thin flexible surface. This is, in fact, a very complicated task. As was shown in [26], an animation in real time is not possible due to the complicated mathematics models describing fabric dynamics. Moreover, the simulation needs considerable computational power for typical FEM algorithm.

Now imagine that the fabric consists of discrete particles. At the beginning of simulation the particles are placed in the nodes of a hexagonal or rectangular grid (see Fig. 14).

Each particle interacts with its neighbors *via* a semi-harmonic two-body potential. Let us introduce gravitation and friction forces in Eqs. (1). Using a *leap-frog* numerical scheme applied to the Newton equations (1), we

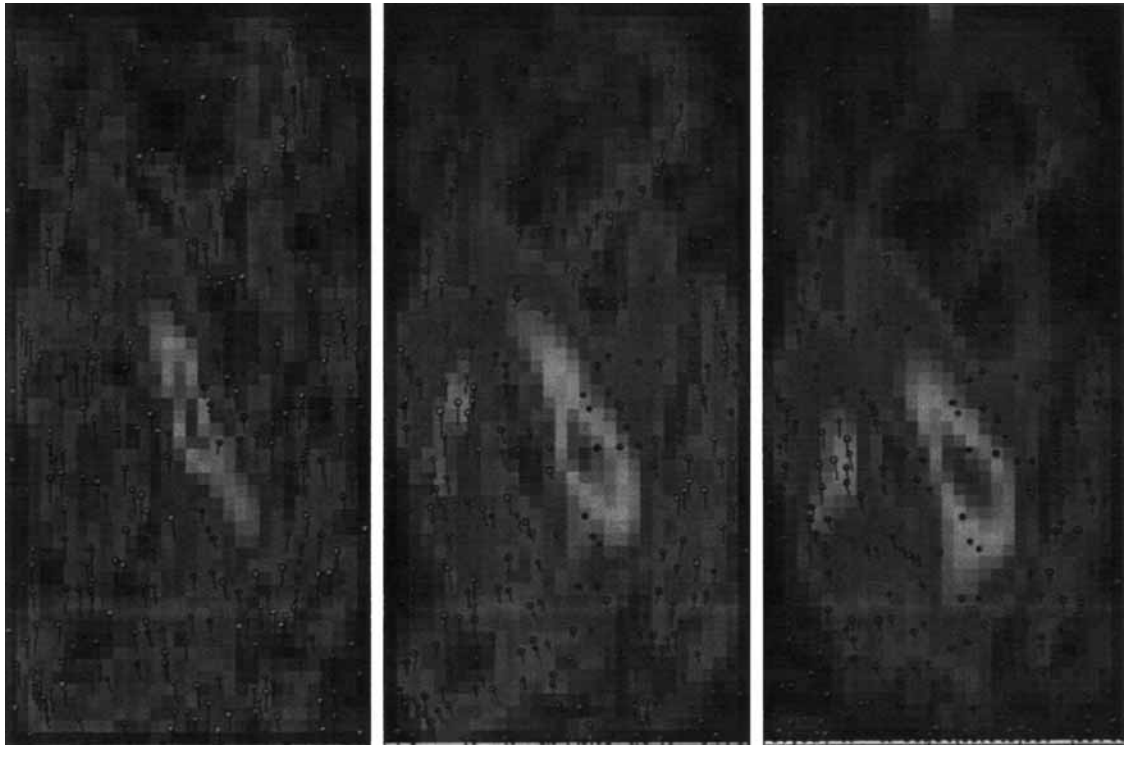

FIGURE 13 Falling up and down sediment particles in continuum media results in a stable 2-D convective pattern.

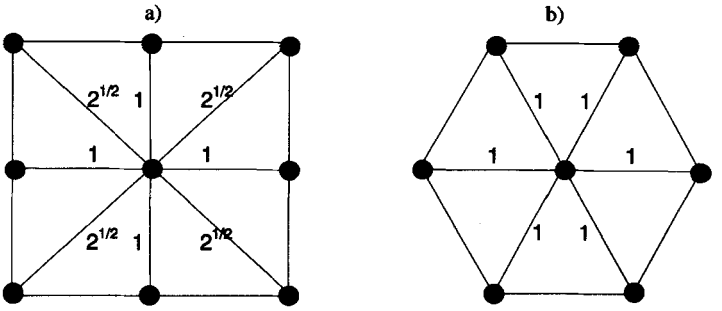

FIGURE 14 Two types of particle meshes in a fabric animation [26].

obtain:

$$\mathbf{v}_i^{n+1/2} = \frac{(1-\varphi)}{(1+\varphi)} \cdot \mathbf{v}_i^{n-1/2} + \frac{\alpha \Delta t}{(1+\varphi)} \cdot \left\{ \sum_{j=1}^{K} \left(r_{ij}^{n^2} - a_{ij}^2 \right) \mathbf{r}_{ij}^n + \frac{g}{\alpha} \mathbf{i}_z \right\},$$
$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \mathbf{v}_i^{n+1/2} \cdot \Delta t \tag{4}$$

assuming that the friction force is:

$$\mathbf{F}_i = -\lambda \cdot \mathbf{v}_i \quad \text{and} \quad \alpha = \frac{k}{m}, \quad \varphi = \frac{\lambda}{2m} \cdot \Delta t$$

r_{ij} – current distance between particles i and j,
a_{ij} – initial distance between i and its neighbors on the mesh at the beginning of simulation,
k – a parameter of the semi-harmonic interparticle potential assumed.

A MD code modified this way can provide realistic pictures of the fabric dynamics with multiscale features and on-line animation on a standard Pentium II based PC (see Fig. 15 for example, see also [27, 28]).

Next, assume that several moving objects are animated. For very simple objects (see Fig. 16) it can be done easily with the MD code on a PC. However, when the objects are more complicated and each consists of about 100,000 particles (*e.g.*, the fiber in Fig. 15) a fluent on-line animation is feasible on a parallel machine.

As shown in [29], objects-to-processors mapping can be used, *i.e.*, for distributing objects evenly over all of the processors. More than one object on a single processor is recommended. Additionally, two processors are used for graphical service and animation supervision, respectively. Load balancing is organized in such a way, that two colliding objects are moved

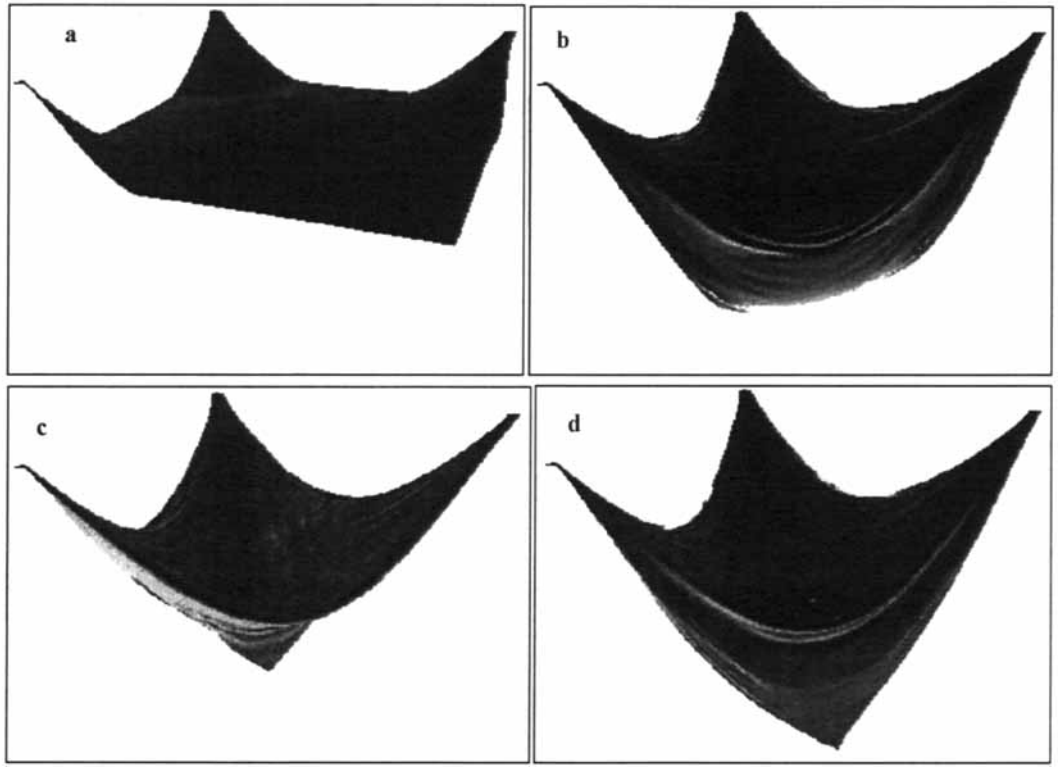

FIGURE 15 Snapshots of animation of the flexible surface using MD code.

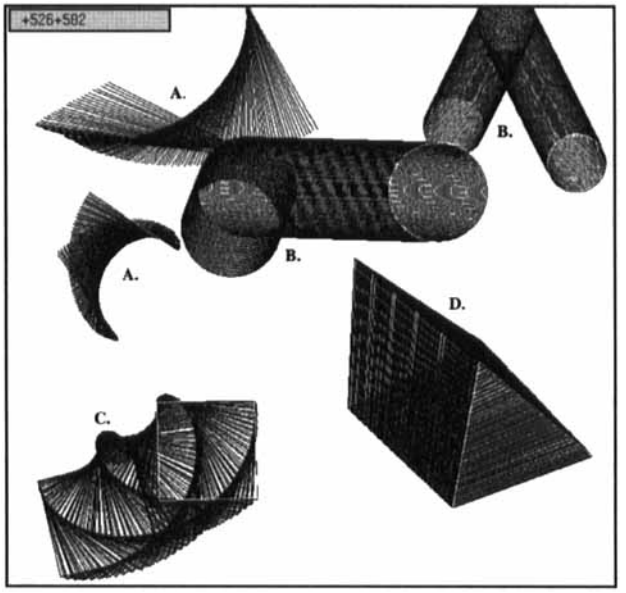

FIGURE 16 Fragments of trajectories of the simple objects animated with the MD approach. The scene consists of: 2 sticks (A), 2 circles of various radiuses (B), a square (C) and a triangle (D). One can see the collisions between the objects and the square rotating after a collision with the wall.

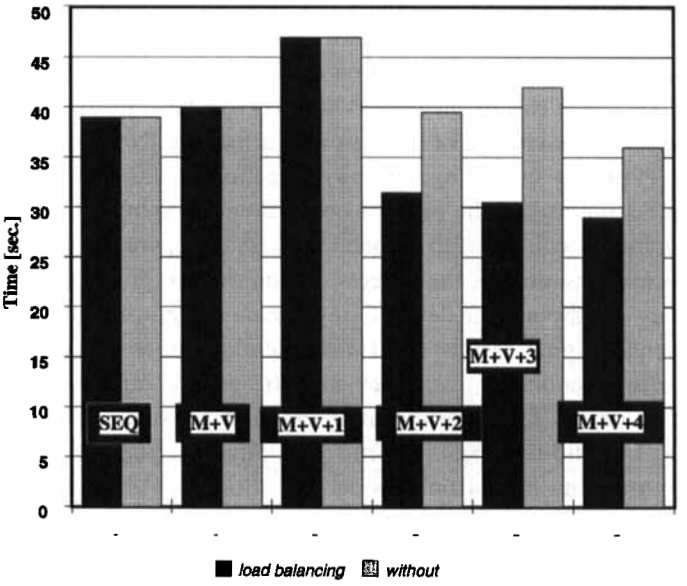

FIGURE 17 Timings for animation of a scene (with and without of load balancing), consisting of four moving cubes (2000 particles each). SEQ – sequential version, M – master processor, V – processor for visualization, K – slaves. Only MD solver was parallelized.

to a single processor. If the number of objects taking part in collision is larger than 2, the number of processors used for simulation of this event is increased. The processors, which are used for simulating the dynamics of the remaining objects, communicate only with the master processor in order to check collision conditions. As shown in Figure 17, for four colliding objects, the optimal number of slaves is 2 (plus master and visualization processors).

6. CONCLUSIONS

We have shown that the MD model can be fruitfully employed as a natural solver in many circumstances. MD simulation can be applied in mesoscopic scales for studies of collective movement of particles. In studies dealing with nonlinear phenomena, such as the Rayleigh–Taylor instability, which have their origins in the microscale, MD can be regarded as a useful tool for simulating the initial phase of mixing and observation of the evolution of multiscale instabilities. Moreover, MD algorithms yield a simple and effective path to parallellization. Such codes can serve as a "backbone" for other more sophisticated particle methods. The most known of them are the dissipative particle dynamics [14] and smoothed particle hydrodynamics [25], which are used in simulating the meso- and macroscopic world phenomena, respectively. We have demonstrated that a complex multiscale phenomenon, like gas release from porous matrix, can be captured by using the DPD method. The change of definition of a particle from a single atom to the cloud of matter and changes in the interaction potential, do not influence greatly the structure of the parallel codes used for pure MD formulation. The MD model can be also applied to visualizing macroscopic objects, giving an impression that the objects dynamics is in good agreement with physical laws, although the detailed physics may be considerably simplified.

These encouraging results of MD applications have demonstrated that very diverse phenomena can obey similar, general and universal rules. Complexity results from non-linear interaction between different levels in hierarchical spatio-temporal structure of physical phenomena [30]. The possibility of using the particle model in multiple-scales enables the cross-scale modeling [24] thus helping to understand the nature of complexity.

Acknowledgments

We thank M.Sc. R. Wcisło and Mr. G. Popiela for discussions and contributions and Mr. T. Chedyński and P. Ukowski for supporting us with

LBG simulation results. The work has been supported by the geosciences program of the US Department of Energy and partially by the Polish Committee for Scientific Research (KBN) Grant No. 8T11C00615.

References

[1] Sloot, P. M. A., Kaandrop, J. A. and Schoneveld, A. (1995). *Dynamic Complex Systems (DCS): A New Approach to Parallel Computing in Physics.* Technical Report of Department of Computer Science, University of Amsterdam, CS-95-08.
[2] Dzwinel, W. (1997). Virtual Particles and Search for Global Minimum. *Future Generation Computer Systems*, **12**, 371–389.
[3] In: *Gather/Scatter Newsletter*, **10**, 3 (1994).
[4] Haile, J. M. (1992). *Molecular Dynamics Simulation*, John Wiley and Sons Inc., New York.
[5] Maitland, G. C., Rigby, M., Smith, E. B. and Wakeham, W. A., *Intermolecular forces. Their Origin and Determination*, Clarendon Press, Oxford, 1981.
[6] Beazley, D. M., Lomdahl, P. S., Jensen, N. G., Giles, R. and Tomayo, P. (1995). Parallel Algorithms for Short-range Molecular Dynamics. *World Scientifics Annual Reviews in Computational Physics*, **3**.
[7] Rapaport, D. C., Molecular Dynamics Simulation. *Computing in Science and Engineering*, January–February, 1999, p. 70.
[8] Rapaport, D. C. (1987). Microscale Hydrodynamics: Discrete-particle Simulation of Evolving Flow Patterns. *Phys. Rev.*, **A36**(7), 3288.
[9] Cui, S. T. and Evans, D. J. (1992). Molecular Dynamics Simulation of Two Dimensional Flow Past a Plate. *Molecular Simulation*, **9**, 179.
[10] Abraham, F. F., Broughton, J. Q., Bernstein, N. and Kaxiras, E. (1998). Spanning the Length Scales in Dynamic Simulation. *Computers in Physics*, **12**(6).
[11] Nakano, A., Bachlechner, M., Campbell, T., Kalia, R., Omeltchenko, A., Tsuruta, K., Vashishta, P., Ogata, S., Ebbsjo, I. and Madhukar, A. (1998). Atomistic simulation of nanostructured materials. *IEEE Computational Science and Engineering*, **5**(4), 68.
[12] Dzwinel, W., Alda, W., Pogoda, M. and Yuen, D. (1999). Turbulent mixing in the microscale: A 2D molecular dynamics simulation. *Physica D* (in print).
[13] Alda, W., Dzwinel, W., Kitowski, J., Mościński, J., Pogoda, M. and Yuen, D. A. (1998). Complex Fluid-dynamical Phenomena Modeled by Large-scale Molecular Dynamics Simulations. *Computers in Physics*, **12**(6), 595.
[14] Koelman, J. M. V. A. and Hoogerbrugge, P. V. (1993). Dynamic simulation of hard-sphere suspensions under steady shear. *Europhysics Lett.*, **21**, 363.
[15] Marsh, C., Backx, G. and Ernst, M. H. (1997). Static and dynamic properties of dissipative particle dynamics. *Physical Review E*, **56**, 1976.
[16] Español, P. (1998). Fluid particle model. *Physical Review E*, **57**(3), 2930–2948.
[17] Español, P., Serrano, M. and Zuniga, I. (1996). Coarse-graining of a fluid and its relation with dissipative particle dynamics and smoothed particle dynamics. *International Journal of Modern Physics C (Physics and Computers)*, **8**, 899.
[18] Rothman, D. H. and Zaleski, S. (1994). Lattice gas models of phase separation: interfaces, phase transitions, and multiphase flow. *Rev. of Mod. Phys.*, **66**(4), 1417–1479.
[19] Novik, K. E. and Coveney, P. V. (1997). Using Dissipative Particle Dynamics to Model Binary Immiscible Fluids. *International Journal of Modern Physics C*, **8**, 909–915.
[20] Dzwinel, W. and Yuen, D. A. (1999). Matching Macroscopic Properties of Binary Fluids to the Interactions of Dissipative Particle Dynamics. *International Journal of Modern Physics C.* (in print).
[21] Herrmann, H. J., The Importance of Computer Simulations of Granular Flow. *Computing in Science and Engineering*, January–February, 1999, p. 72.
[22] Schwarzer, S. (1995). Sedimentation and Fluid Flow through Porous Media: Simulating Dynamically Coupled Discrete and Continuum Phases. *Phys. Rev. E*, **52**(6), 6461–6475.
[23] Hadjiconstantinou, N. G. and Petera, A. T. (1997). Heterogeneous atomistic-continuum representation for dense fluid systems. *Int. Journal of Modern Physics C*, **8**(4), 967–976.
[24] Dzwinel, W., Alda, W. and Yuen, D. A. (1999). Cross-scale numerical simulations using discrete particle models. *Molecular Simulation*, **22**, 397–418.

[25] Peschek, A. G. and Libersky, L. D. (1993). Cylindrical Smoothed Particle Hydrodynamics. *Journal of Computational Physics*, **109**(1), 76.
[26] Wang, B., Wu, Z., Sun, Q. and Yuen, M. M. F., A deformation model of thin flexible surfaces. *6 Int. Conf. in Cent. Eur. on Computer Graphics and Visualization*, Plzen, Czech Republic, February 9–13, 1998.
[27] Wcisło, R., Dzwinel, W., Kitowski, J., Mościński, J. (1993). *Molecular Dynamics for Real World Phenomenon Animation*. CCP5 Information Quarterly, Darresbury Laboratory, Warrington, U.K., August, 1993, **38**, 25.
[28] Wcisło, R., Dzwinel, W., Kitowski, J. and Mościński, J. (1994). Real-time Animation Using Molecular Dynamics Methods. *Machine Graphics and Vision*, **3**(1/2), 203–210.
[29] Wcisło, R., Kitowski, J. and Mościński, J. (1996). Parallelization of a code for animation of multi-object system. In: Wasniewski, J., Dongarra, J., Madsen, K. and Olesen D. (Eds.), *Applied Parallel Computing – Industrial Computation and Optimization*. Lecture Notes in Computer Science 1184, 697, Springer.
[30] Badii, R. and Politi, A. (1997). Complexity. *Hierarchical Structures and Scaling in Physics*, Cambridge University Press, U.K.