

## STUDY OF INTEGRATION ALGORITHM AND TIME STEP ON MOLECULAR DYNAMIC SIMULATION

Janusz Bytnar, Anna Kucaba-Piętal

**Abstract:** A simulation is reliable when the simulation time is much longer than the relaxation time of the quantities in question. The aim of this work is to address the question when Molecular Dynamics (MD) simulation is reliable and how it depends on the integration algorithms and optimal time step. There were certain problems related to the choice of integration algorithms on Molecular Dynamics simulations. The effect of time step on convergence to equilibrium in Molecular Dynamics simulation has been studied.

**Keywords:** Molecular Dynamics, computer simulations, integration algorithms

**ACM Classification Keywords:** A.0 General Literature - Conference proceedings

---

### Introduction

---

After obtaining the results of a research, each scientist needs to consider verification and validation of those results. Computer simulation of molecular systems is playing an ever growing role in academic and industrial research. In areas ranging from materials science and chemistry to pharmacy and molecular biology, computer simulation is already a part of daily practice. The behavior of a variety of molecular systems can be studied by using the Molecular Dynamics (MD) simulation method. These include liquids, solutions, electrolytes, polymers such as proteins, DNA, and polysaccharides, as well as membranes, liquid crystals, crystals, and zeolites [Allen, 1987], [Bicout, 1996].

Computer simulation of molecular systems requires software to calculate the interatomic interactions and to integrate the equations of motion [Griebel, 2007].

Many models, for example in materials science or in astrophysics, contain large number of interacting bodies (called particles), as for example stars and galaxies or atoms and molecules. In many cases the number of particles can reach several millions or more. For instance, every cubic meter of gas under normal conditions (i.e., at temperature of 273.15 Kelvin and pressure of 101.325 kilopascal) contains  $2.68678 \times 10^{25}$  atoms (Loschmidt constant). 12 grams of the carbon isotope C12 contain  $6.02214 \times 10^{23}$  atoms (Avogadro constant).

These are some of the reasons why computer simulation has recently emerged as a third method in science besides experimental and theoretical approaches. Over the past years, computer simulation has become an indispensable tool for the investigation and prediction of physical and chemical processes. In this context, computer simulation means the mathematical prediction of technical or physical processes on modern computer systems [Griebel, 2007].

The deterministic method of Molecular Dynamics (MD) simulation, although theoretically valid for the whole range of densities, is employed mainly for liquids and solids [Allen, 1987]. The long flight paths between collisions of gas molecules make the method of Molecular Dynamics prohibitively expensive, while other methods, like e.g. Direct Monte-Carlo Simulation, can give satisfactory results at much lower computational cost. Molecules in liquids are densely packed and remain in constant contact with the neighbours. Under such conditions Molecular Dynamics seems to be the most accurate and, at the same time, the most efficient simulation method.

Molecular Dynamics requires the description of the molecules and the forces acting between them. Perhaps the most often, to describe the Van-der-Waals forces, the Lennard-Jones potential is used. It assumes that the molecules are spherically symmetric, repelling one another at close and attracting at far distances.

---

### Procedure of Molecular Dynamic

---

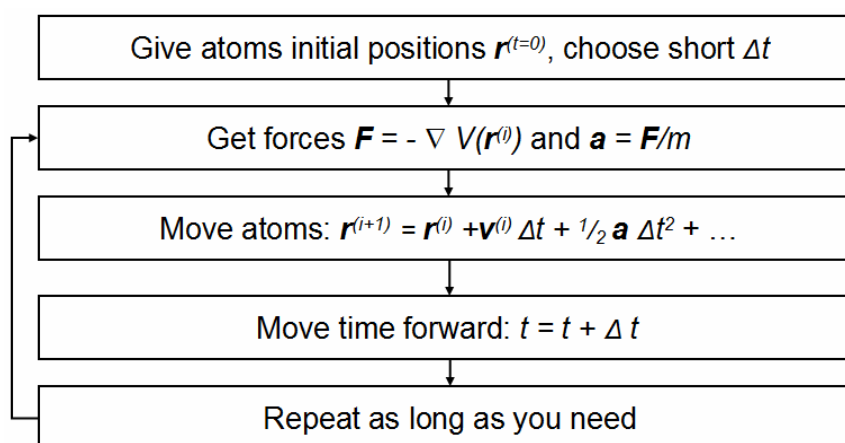
Molecular Dynamics (MD) is a computer simulation technique:

- the time evolution of interacting atoms is followed by integrating their equations of motion.
- the laws of classical mechanics are followed, and most notably Newton's law:

$$F_i = m_i a_i \quad (1)$$

$$a_i = d^2 r_i = dt^2 \quad (2)$$

- The Molecular Dynamics (MD) procedure can be written as follows:



- Given the initial set of coordinates and velocities, the subsequent time evolution is in principle completely determined.
- Atoms and molecules will 'move' in the computer, bumping into each other, vibrating about a mean position (if constrained), or wandering around (if the system is fluid), in a way similar to what real atoms and molecules do.
- The computer calculates a trajectory of the system
- 6N-dimensional phase space (3N positions and 3N moments).

The orientation of the molecules can be represented in several ways, however the use of quaternions [Refson, 2001] seems to be the most advisable. The most important advantage of quaternions is the fact, that they lead to equations of motion free of singularities (which is not the case for e.g. Euler angles). This, in turn, leads to good numerical stability of the simulation.

Integration algorithms used in Molecular Dynamics simulation are based on finite difference methods, with discretized time and the time step equal to  $\Delta t$ . Knowing the positions and some of their time derivatives at time  $t$  (the exact details depend on the type of algorithm), the integration scheme gives the same quantities at a later time  $(t + \Delta t)$ . With such procedure the evolution of the system can be followed for long times [Allen, 1987].

### Stages of simulation:

**Initiation:** placing the molecules of water and the copper atoms in the knots of crystalline mesh. After that the velocities of the molecules are initialized. Their values are sampled at random from the Maxwell – Boltzmann distribution for the assumed temperature.

**Balancing:** after initiation the positions of molecules are far from equilibrium. The whole ensemble is allowed to move freely for some time to attain equilibrium positions. This is always connected with decreasing the potential and increasing the kinetic energy of the molecules, i.e. increasing the temperature of the medium. This excess temperature must be removed by a suitable “thermostat”.

**Actual simulation:** after attaining equilibrium, the simulation starts. The required data (specified in advance) are accumulated in “dump-files” in preselected time intervals. Any property of interest.

---

### Integration Algorithms

---

The engine of a Molecular Dynamics program is the time integration algorithm, required to integrate the equation of motion of the interacting particles and follow their trajectory.

The integration scheme gives the possibility to find particle position at a later time  $t + \Delta t$ . By iterating the procedure, the time evolution of the system can be followed for long times.

Of course, these schemes are approximate and there are errors associated with them. In particular, one can distinguish between:

- Truncation errors, related to the accuracy of the finite difference method with respect to the true solution. Finite difference methods are usually based on a Taylor expansion truncated at some term, hence the name. These errors do not depend on the implementation: they are intrinsic to the algorithm.
- Round – off errors, related to errors associated to a particular implementation of the algorithm. For instance, to the finite number of digits used in computer arithmetic.

Both errors can be reduced by decreasing  $\Delta t$ . For large  $\Delta t$ , truncation errors dominate, but they decrease quickly as  $\Delta t$  is decreased. For instance, the Verlet algorithm has a truncation error proportional to  $\Delta t^4$  for each integration time step. Round – off errors decrease more slowly with decreasing  $\Delta t$ , and dominate in the small  $\Delta t$  limit [Ercolessi, 1997].

Using time integration techniques, it is possible to determine the velocity and position of a particle from its acceleration. There is a variety of different numerical methods available, however the nature of Molecular Dynamics simulations has narrowed down the field to a handful of methods. Methods which require more than one force calculation per time step are considered wasteful and can only be considered if the time step can be proportionally increased, while still maintaining the same accuracy.

Similarly, adaptive methods that change the time step dynamically are useless due to the rapidly changing neighbourhood of each atom. As a result, only two methods have become mainstream in Molecular Dynamics field, that is, the Verlet method and predictor-corrector method [Rapaport , 2004].

Both methods are based on finite difference techniques, derived from the Taylor expansion of the  $r(t)$ .

#### Basic Verlet Method

In Molecular Dynamics, time integration algorithm that is used very common is Verlet algorithm [Verlet, 1967]. The basic idea is to write two third-order Taylor expansions for the positions  $r(t)$ , one forward and one backward in time. The basic form of the Verlet method is defined by the equation:

$$r(t + \Delta t) = 2r(t) - r(t - \Delta t) + (\Delta t)^2 a(t) + O(\Delta t^4) \quad (3)$$

where  $a(t)$  is the acceleration. Via the combination of the force calculation with Newton's second law of motion, the acceleration is defined as

$$a(t) = -(1/m)\nabla U(r(t)) \quad (4)$$

While not required for computation, the velocity variable can be found by using the equation

$$v(t) = \frac{r(t + \Delta t) - r(t - \Delta t)}{2\Delta t} \quad (5)$$

The Verlet algorithm uses positions and accelerations at time  $t$  and positions from time  $t-\Delta t$  to calculate new positions at time  $t+\Delta t$ . The Verlet algorithm uses no explicit velocities. The advantages of the Verlet algorithm are:

- It is straightforward
- The storage requirements are modest

The disadvantage is that the algorithm is of moderate precision.

A problem with this version of the Verlet algorithm is that velocities are not directly generated. While they are not needed for the time evolution, their knowledge is sometimes necessary. Moreover, they are required to compute the kinetic energy  $E_K$ , whose evaluation is necessary to test conservation of the total energy  $E = E_K + E_P$ . We can also calculate temperature of the simulated molecular system from kinetic energy.

$$E_K = \frac{1}{2}k_B T \quad (6)$$

Where  $k_B$  is the Boltzman constant,  $T$  is the temperature.

This is one of the most important tests to verify that a Molecular Dynamics simulations of real processes is proceeding correctly.

However, the error associated to this expression is of order  $\Delta t^2$  rather than  $\Delta t^4$ . To overcome this difficulty, some variants of the Verlet algorithm have been development.

### Velocity Verlet Method

However more common algorithm is a related one, Velocity Verlet algorithm. Here the velocity, position and accelerations at time  $t+\Delta t$  are obtained from the same quantities at time  $t$  [Verlet, 1967]. This uses a similar approach but explicitly incorporates velocity, solving the first-time step problem in the Basic Verlet algorithm:

$$r(t + \Delta t) = r(t) + v(t)\Delta t + 1/2 a(t)\Delta t^2 \quad (7)$$

$$v(t + \Delta t) = v(t) + 1/2[a(t) + a(t + \Delta t)]\Delta t^2 \quad (8)$$

### Beeman Method

This algorithm is also closely related to the Verlet algorithm

$$r(t + \Delta t) = r(t) + v(t)\Delta t + 2/3 a(t)\Delta t^2 - 1/6 a(t - \Delta t)\Delta t^2 \quad (9)$$

$$v(t + \Delta t) = v(t) + v(t)\Delta t + 1/3 a(t + \Delta t) + 5/6 a(t)\Delta t - 1/6 a(t - \Delta t)\Delta t \quad (10)$$

The advantage of this algorithm is that it provides a more accurate expression for the velocities and better energy conservation. The disadvantage is that the more complex expressions make the calculation more expensive [Beeman, 1976].

A variant of the Verlet method, called the velocity-Verlet method, addresses this problem by directly including the velocity in computation. As a result, particle velocities are known at the same time step as coordinates, and the high-order accuracy of the method is maintained. Additionally, particle velocities are necessary for kinetic energy calculations, which play a critical role in most Molecular Dynamics simulations [Haile, 1997].

### Time Step

Lennard – Jones potential is the most popular interaction potential used in Molecular Dynamics (MD) simulations to describe Van-der-Waal forces [Karniadakis, 2005]. The form of the Lennard – Jones potential is as follows:

$$V(r) = 4\varepsilon \left[ \left( \frac{\delta}{r} \right)^{12} - \left( \frac{\delta}{r} \right)^6 \right] \tag{11}$$

where  $\varepsilon$  and  $\delta$  are the Lennard – Jones parameters that depend on the atoms involved in the interaction. Note that:

- $\varepsilon$  is related to the interaction strength, and a higher  $\varepsilon$  corresponds to a higher interaction energy between the atoms
- $\delta$  corresponds to the distance at which the potential between the two atoms goes to zero, which can be approximately taken as the diameter of a fluid atom.

The term  $\sim 1/r^{12}$ , dominating at shorter distance, models the repulsion between atoms when they are brought very close to each other.

The term  $\sim 1/r^6$ , dominating at large distance, constitute the attractive part. This is the term which gives cohesion to the system. A  $1/r^6$  attraction is originated by van der Waals dispersion forces, originated by dipole – dipole interactions in turn due to fluctuating dipoles. These are rather weak interactions, which however dominate the bonding character of closed – shell systems, that is, rare gases such as Argon. Therefore, these are the materials that Lennard – Jones potential could mimic fairly well [Ercolessi, 1997]. The parameters  $\varepsilon$  and  $\delta$  are chosen to fit the physical properties of the material.

In the Molecular Dynamics (MD) simulation with Lennard – Jones interaction potentials, the time and the other physical quantities are represented and typically computed using reduced units. Table 1 summarizes the units for various quantities used in simulations for instance, length, temperature, and density. In the Table 1, symbols  $\varepsilon$  and  $\delta$  denote constants as defined in equation (10),  $k_B$  is the Boltzman constant, and  $m$  is the mass of a atom.

Table 1. Units for various quantities in Lennard – Jones fluids [Griebel, 2007], [Karniadakis, 2005]

Length	$\delta$	Velocity	$(\varepsilon / m)^{1/2}$
Mass	$m$	Shear rate	$(\varepsilon / m \delta^2)^{1/2}$
Energy	$\varepsilon$	Stress	$\varepsilon / \delta^3$
Time	$(m \delta^2 / \varepsilon)^{1/2}$	Viscosity	$(m \varepsilon)^{1/2} \delta^2$
Number density	$\delta^{-3}$	Diffusivity	$\delta (\varepsilon / m)^{1/2}$
Temperature	$\varepsilon / k_B$		

In many publications [Griebel, 2007], [Karniadakis, 2005] authors calculate time step from the formula:

$$\Delta t = 0,001 * \sqrt{\frac{m\delta^2}{\varepsilon}} \quad (12)$$

We intended to use as large a time step as possible so that we can explore more of the phase space of the system. However, since we truncate the Taylor's series expansions, the time step needs to be small enough so that the expansion can provide a reliable estimate of the atomic positions and velocities at the end of the time step (see Fig. 1). For typical algorithms with a time accuracy of order three, one uses a time step that is a fraction of the period of the highest-frequency motion in the system [Karniadakis, 2005]. For a typical simulation of water transport, where the O – H bond length is fixed, a time step size of 1.0 to 2.0 fs is commonly used.

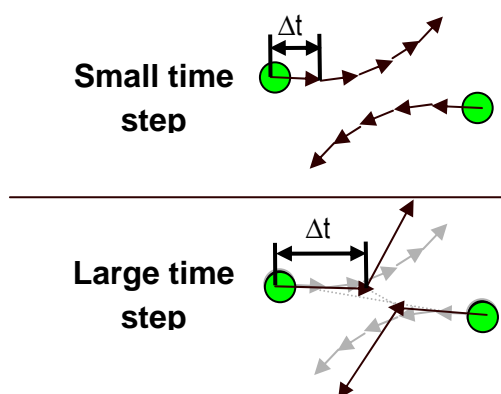


Fig. 1. The effect of the small and large temporary step

## Results

The aim of the research was to analyze the influence of integration algorithm and time step on the computational time and memory complexity. Also influence of the size of molecular systems on efficiency integration algorithms was studied.

The simulations were carried out for one molecular model of water TIP4P and three integration algorithms were applied: Velocity Verlet, Beeman and Beeman algorithm with Predictor – Corrector modifications.

We use for all algorithms different time steps  $\Delta t = 0,00001$ ,  $\Delta t = 0,00002$ ,  $\Delta t = 0,00005$ ,  $\Delta t = 0,0001$  and  $\Delta t = 0,0005$  picosecond long. The calculations were carried out over 100 000 time steps.

The program MOLLY [Refson, 2001], suitably modified, was used for this purpose. Moldy is free software; which may redistribute it and/or modify it under the terms of the GNU.

The physical properties of materials and their electrostatic interactions were taken into account. The number of water molecules was equal to 500 and 20000. The periodic boundary conditions were applied. The Lennard-Jones potential was assumed for interactions between water molecules [Kucaba – Pietal, 2004], [Bytnar, 2008].

Molecular Dynamics is always an approximate science approach, the longer the time step, the less accurate the results. In the worst case scenario, the time step will allow atoms to move too far between single iterations, allowing atoms to get closer together than they ever could in a real liquid. This usually causes an incorrect “chain reaction”, whereby two close particles repel at a much faster speed than normal causing them to bump even closer other atoms, which are repelled at an even greater velocity. This effect compounds until all atoms are moving at unrealistic speeds and eventually arithmetic overflows will occur. When we attempted to use larger time step (1 fs) the program crashed.

If we only consider figures 2-5 then we can deduct that ideally time step should be as small as possible.

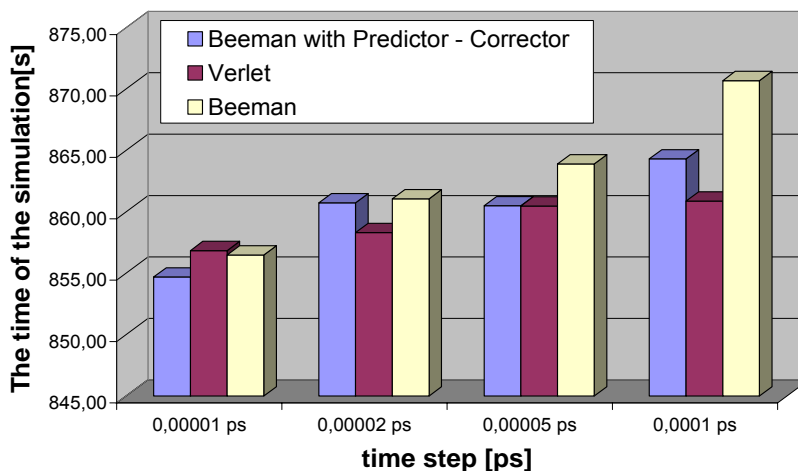


Fig. 2. The time of the whole simulation (500 molecules of water) – Integration algorithms with various time steps

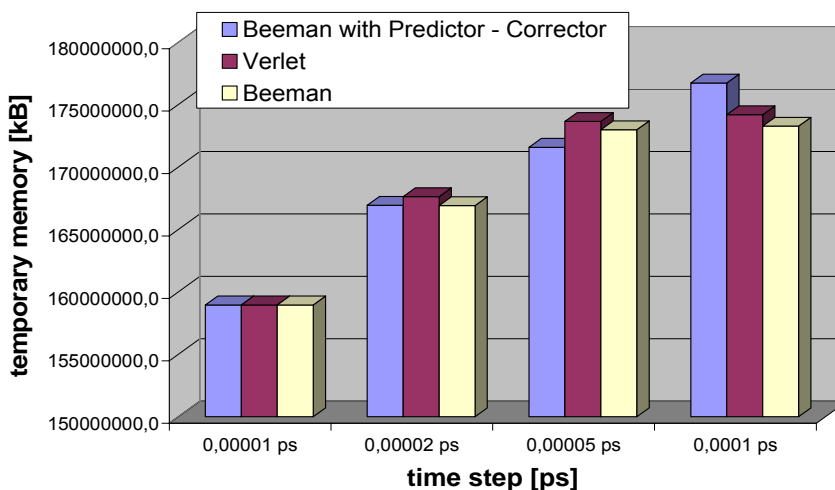


Fig. 3. The reservation of the temporary memory (500 molecules of water) – Integration algorithms with various time steps

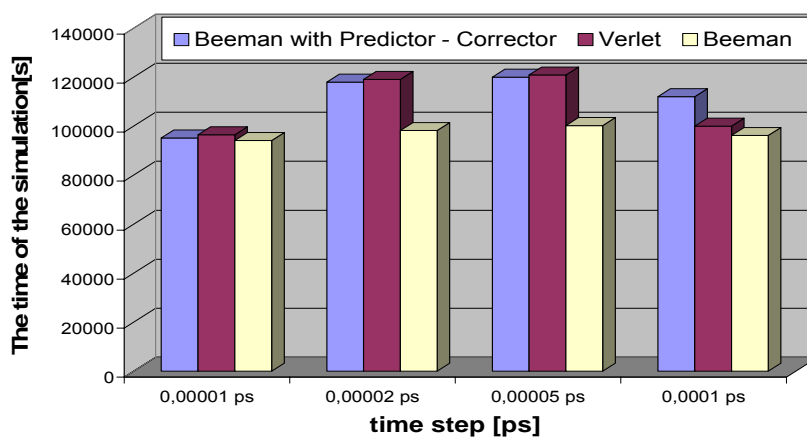


Fig. 4. The time of the whole simulation (20000 molecules of water) – Integration algorithms with various time steps

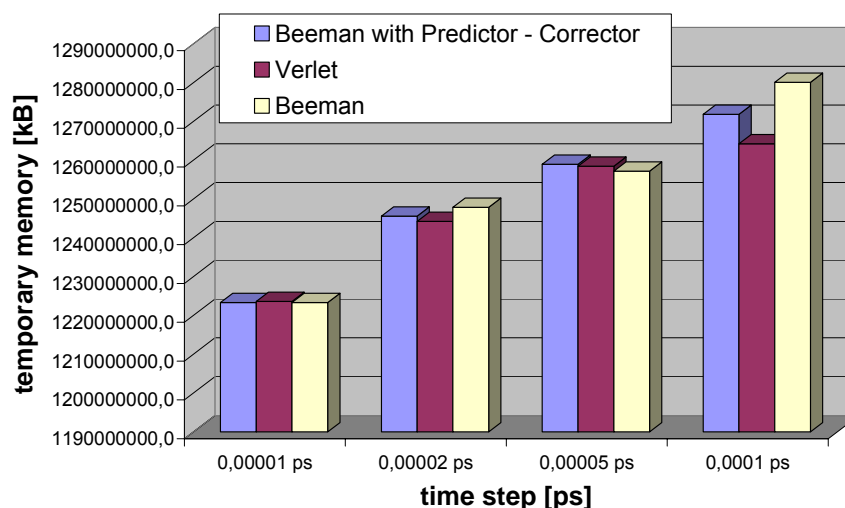


Fig. 5. The reservation of the temporal memory (20000 molecules of water) – Integration algorithms with various time steps

In figures 2-5, several different time step simulations were run in different integration algorithm. Using a small time step of 0,00001 ps gives good results connected with the reservation of the memory and the time of the whole simulation. The increment of the time step forces the use of the greater spaces of the memory and time of the whole simulations.

## Conclusions

In Molecular Dynamics simulation, there exist several algorithms to realize equations of motion of molecular systems.

From the presented diagrams we can see that the choice of these algorithms with correct time steps decides about the time of the whole simulations and also about the necessary reservation of the memory.

The choice of the time step in integration algorithm from equation (11) can be good for abstract molecular models. However, this type of simulations does not take into account many factors, which are very important for simulations of real materials. In Molecular Dynamics simulations of real processes not only speed of calculations and reservation of the memory are important. In this case what is very important is the thermodynamics of the simulated molecular configuration during the whole simulation as also the obtained results of physical properties simulated materials which in confrontation with the results of these materials in the larger scale will be satisfactory.

The influence of the correct choice of the time step was also present in paper [Bytnar, 2010]. From the presented diagrams (considered in paper [Bytnar, 2010]) it is clear, that for the problems considered, i.e. real flows in nanochannels, the Gaussian thermostat is much more efficient if the time step  $\Delta t$  is well chosen.

If we consider only time of the whole simulations and required memory (see Figures 2-5) we can say that the best performance and accuracy is if the time step is as small as possible also for small and large molecular systems.

Additionally we can see that the performance of all integration algorithms (Velocity Verlet, Beeman and Beeman with Predictor – Corrector modifications) is very similar also for small (500 molecules of water) and large (20000 molecules of water) molecular systems. This means that the choice of the integration algorithm does not have large signification for Molecular Dynamics simulations.



The smaller time step effects enlargement of the time of the whole simulation and the required memory, however the larger temporary step can effect generating the inaccurate trajectory of motion. Clearly, the larger time step, the less accurately our solution will follow the correct classical (see Fig 1).

From the presented figures in this paper it can be deduced that a good way of checking whether the time step is satisfactory is to run an equilibrium simulation of small molecular system because results for small (500 molecules of water) and large (20000 molecules of water) molecular system are very similar.

In summary, the choice of time step has a big impact on accuracy of simulations. It is recommended that, to avoid the incorrect "chain reaction" phenomena, if two atoms get unrealistically close, the user should be warned that the time step should be decreased and be given the option to terminate the program since the results are already effectively useless. A more advanced program might provide warnings if the user enters an unrealistically large time step before the simulation is allowed to start.

The physical explanation of other peculiarities of the presented diagrams, particularly the thermodynamics properties molecular systems, requires further investigation.

The study of Verification and Validation methods gives the possibility of the construction of the correct computer model to the description of the studied phenomenon, how also receipt of the exact and authentic results of computer simulations.

Influence on the results of the simulation in the Molecular Dynamics method has e.g. Integration algorithms and time step, molecular model of water or another material, mechanisms to control the temperature of the system (thermostats). Therefore, the study of the methods of Verification and Validation will be closely connected with the above mentioned factors.

---

## **Acknowledgment**

---

Calculations were made in the Interdisciplinary Centre for Mathematical and Computational Modeling (ICM) University of Warsaw (grant no. G44-9):

<https://granty.icm.edu.pl/lcmGrants/displayGrant/showGrants.jsp>

We also want to thank for the possibility of the realization of calculations in the Institute of Fundamental Technological Research PAN, Department of Mechanics and Physics of Fluids in Warsaw.

---

## **Bibliography**

---

- [Allen, 1987] M. P. Allen and D. J. Tildesley: Computer Simulation of Liquids, Clarendon Press, Oxford, 1987,
- [Bicout , 1996] D. Bicout and M. Field: Quantum Mechanical Simulation Methods for Studying Biological Systems, Springer, Berlin, 1996,
- [Griebel, 2007] M. Griebel S. Knapek, G. Zumbusch: Numerical Simulation in Molecular Dynamics, Numerics, Algorithms, Parallelization, Applications, Springer-Verlag Berlin Heidelberg, 2007,
- [Ercolessi, 1997] F. Ercolessi: A molecular dynamics primer, Spring College in Computational Physics, ICTP, Trieste, Italy, 1997,
- [Rapaport , 2004] D. C. Rapaport: The Art of Molecular Dynamics Simulation. Cambridge University Press, 2004,
- [Verlet, 1967] L. Verlet: Computer `experiments' on classical fluids. I. thermodynamical properties of Lennard-Jones molecules, Phys. Rev. 165, 201-214, 1967,
- [Beeman, 1976] D. Beeman: Some multistep methods for use in molecular dynamics calculations, J. Comp. Phys. 20 130-139, 1976,
- [Haile, 1997] J. Haile: Molecular Dynamics Simulation: Elementary Methods. New York: John Wiley and Sons Inc., 1997,

- 
- [Karniadakis, 2005] G. Karniadakis, A. Beskok, N. Aluru: Microflows and Nanoflows – Fundamentals and Simulation, Interdisciplinary Applied Mathematics, Springer, 2005,
- [Refson, 2001] K. Refson: Moldy User's Manual. Chapter II, <ftp://ftp.earth.ox.ac.uk/pub>,
- [Bytnar, 2010] J. Bytnar, A. Kucaba-Piętal, Z. Walenta: Verification and Validation of Molecular Dynamics Simulation, Publications S.H.F. Second European Conference in Microfluidics, ISBN 978-2-906831-85-8, Session P4-3-Liquid Microflows, Toulouse, France, 2010,
- [Bytnar, 2008] J. Bytnar, A. Kucaba-Piętal, Z. Walenta: Influence of molecular models of water on computer simulations of water nanoflows - Proceedings of International Conference on Computer Science and Information Technology, Volume 3, ISSN 1896-7094 pages 269 – 275, IEEE CS Press, Los Alamitos, CA, 2008
- [Kucaba – Piętal, 2004] A. Kucaba-Piętal: Microflows modelling by use micropolar fluid model, OW RUT, Rzeszow, 2004;

---

### Authors' Information

---

**Janusz Bytnar** –*Technical and Economical State School of Higher Education in Jaroslaw, Institute of Technical Engineering, ul. Czarnieckiego 16, 37-500 Jaroslaw, Poland; e-mail: [janusz.bytnar@pwste.edu.pl](mailto:janusz.bytnar@pwste.edu.pl)*

*Major Fields of Scientific Research: Software technologies, Molecular Dynamics, Computer simulations, Integration algorithms research*

**Anna Kucaba - Piętal** – *Technical and Economical State School of Higher Education in Jaroslaw, Institute of Technical Engineering, ul. Czarnieckiego 16, 37-500 Jaroslaw, Poland;*

*Rzeszow University of Technology, The Faculty of Mechanical Engineering and Aeronautics Powstancow Warszawy 8, 35-959 Rzeszow, Poland; e-mail: [anpietal@prz.edu.pl](mailto:anpietal@prz.edu.pl)*

*Major Fields of Scientific Research: Scientific Calculations, Computational Mechanics,, Molecular Dynamics, Computer simulations, Nano and Micromechanics*