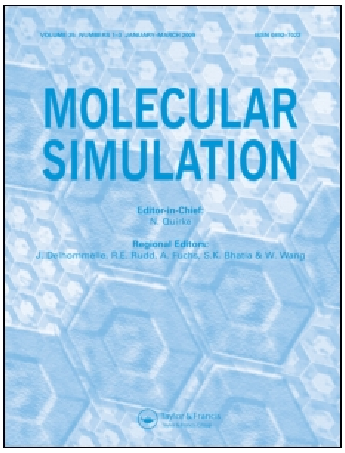

Molecular Simulation
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713644482

On Molecular Dynamics Algorithms

Marek Litniewski[a]
[a] Institute of Physical Chemistry of the Polish Academy of Sciences, Warsaw, Poland

Online publication date: 26 October 2010

To cite this Article Litniewski, Marek(2003) 'On Molecular Dynamics Algorithms', Molecular Simulation, 29: 3, 223 — 229
To link to this Article: DOI: 10.1080/0892702031000089687
URL: http://dx.doi.org/10.1080/0892702031000089687

Molecular Simulation, 2003 Vol. 29 (3), pp. 223–229

On Molecular Dynamics Algorithms

MAREK LITNIEWSKI

Institute of Physical Chemistry of the Polish Academy of Sciences, Kasprzaka 44/52, Warsaw 01-224, Poland

(Received August 2001; In final form November 2001)

The step errors (the local errors, called also the truncation errors) of algorithms used in molecular dynamics simulations result in errors of thermodynamic properties of simulated systems. The simulations on the Lennard–Jones (L–J) liquid showed that in the case of the Verlet algorithm the values of the errors are noticeable even if we apply Beeman's technique. For the time step $\Delta t = 0.01$ at the temperature $k_{\mathrm{B}}T \approx 1.26$ and the density $\rho = 0.75$, Beeman's technique "shifts" the pressure by about 0.009 (all in L–J units). The "shift" is proportional to Δt^2 and strongly decreases with a decrease of ρ. For the Gear 4th order predictor–corrector method, the shift, at $\Delta t = 0.0125$, is only a few times lower than that for Beeman's technique. The effect is due to scaling of velocity (necessary to eliminate the energy drift), but not to the step errors and quickly vanishes with $\Delta t \rightarrow 0$. In order to eliminate the effect, the Cowell–Numerov implicit method in which the nonlinear equations for space variables (equivalent to the equations of the Gear 4th order implicit method) are approximately solved by iteration applied only to the subset of the most rapidly varying forces is proposed. The method is stable, accurate, and efficient. At the given state point for $\Delta t = 0.0125$ the "shift" of pressure, for momentums evaluated from the 5th order formula, amounts to about 0.0002. This value is approximately equal to the result for Calvo and Sanz-Serena's 4th order method (CSS). The proposed method is more efficient than CSS, especially at a high cut-off distance (R_{C}). For $R_{\mathrm{C}} \rightarrow \infty$, the time of performing single time step approaches to that of the Verlet method.

Keywords: Molecular dynamics; Algorithms; Symplectic; Error; Verlet; Beeman; Gear; Cowell; Numerov

INTRODUCTION

Perhaps the most widely used method of integrating the equations of motion is the so called Verlet algorithm [1–3]. The method is very simple and also has some important properties. The algorithm is time-reversible and symplectic [4]. The method is very stable and has excellent energy conserving properties [2], but as will be shown, it has an important disadvantage. Because of the high step error (local error) of Eq. (1), the Verlet method generates noticeable errors in properties of simulating systems.

The following formulas are used in the Verlet algorithm for evaluating the $3N$ dimension vector of the coordinates ($\mathbf{q}$) and momentums ($\mathbf{p}$) of the N particle system at time $t + \Delta t$:

$$\mathbf{q}(t+\Delta t) = 2\mathbf{q}(t) - \mathbf{q}(t-\Delta t) + \frac{\Delta t^2}{m}\mathbf{F}(t) + O(\Delta t^4) \quad (1)$$

$$\mathbf{p}(t+\Delta t) = \frac{m}{\Delta t}[\mathbf{q}(t+\Delta t) - \mathbf{q}(t)] + \frac{\Delta t}{2}\mathbf{F}(t+\Delta t) + O(\Delta t^2) \quad (2)$$

where $\mathbf{F}(t)$ denotes the force and m—the mass.

The above form is equivalent to the form of the so called velocity Verlet algorithm. If $\mathbf{F}$ is independent of $\mathbf{p}$, Eq. (1) can be solved independently and Eq. (2) may be treated as a definition of momentum. But the definition omits higher terms in Δt:

$$\mathbf{Err}^{\mathrm{V}}(\mathbf{p}) = -\frac{1}{6}\frac{\mathrm{d}\mathbf{F}}{\mathrm{d}t}\Delta t^2 + O(\Delta t^3) \quad (3)$$

The average: $\langle \mathbf{p}_i \mathrm{d}\mathbf{F}_i/\mathrm{d}t \rangle$ $(i = 1, N)$ is not zero. As a result, individual errors of $\mathbf{p}_i$ accumulate shifting the average value of the total kinetic energy (E_{k}) by:

$$\Delta E_{\mathrm{k}} \cong \left\langle \sum_{i=1}^{N} \mathbf{p}_i \mathbf{Err}(\mathbf{p}_i)/m_i \right\rangle \quad (4)$$

According to Eq. (3), ΔE_{k} depends on the time step as Δt^2. For sufficiently high Δt, there should appear a shift ($\propto \Delta t^2$) in the thermodynamic properties if we

*Fax: +48-3912-0238. E-mail: mark@ichf.edu.pl

ISSN 0892-7022 print/ISSN 1029-0435 online © 2003 Taylor & Francis Ltd
DOI: 10.1080/0892702031000089687

simulate at constant energy or temperature. The effect is known [5] and the way to improve the results is to replace Eq. (2) by a better approximation, for example:

$$\mathbf{p}(t+\Delta t)=\frac{m}{\Delta t}[\mathbf{q}(t+\Delta t)-\mathbf{q}(t)]+\frac{\Delta t}{6}[2\mathbf{F}(t+\Delta t)+\mathbf{F}(t)]+O(\Delta t^3) \tag{5}$$

Equations (1) and (5) are equivalent to so called Beeman's technique [6], which is recommended [5,7] as the most accurate of all the Verlet-equivalent integration schemes. The term omitted in Eq. (5) is:

$$\mathbf{Err}^{\mathrm{G4}}(\mathbf{p})=-\frac{1}{24}\frac{\mathrm{d}^2\mathbf{F}}{\mathrm{d}t^2}\Delta t^3+\frac{7}{360}\frac{\mathrm{d}^3\mathbf{F}}{\mathrm{d}t^3}\Delta t^4+O(\Delta t^5) \tag{6}$$

which is usually much lower than Eq. (3). $\langle \mathbf{p}_i \mathrm{d}^2\mathbf{F}_i/\mathrm{d}t^2\rangle = 0$ (the second derivative is an even function of $\mathbf{p}$) so ΔE_{k} for Beeman's technique is of the order of Δt^4.

The error of Eq. (1) is $O(\Delta t^4)$ and propagates as Δt^2. As a result, the influence of the step error on the measured averages is still $O(\Delta t^2)$. Therefore, Beeman's technique is more accurate than the Verlet velocity method only if the influence of the local error of Eq. (1) can be neglected when compared to the influence of the error of Eq. (2). It will be shown in the next section that at high densities this condition is not fulfilled.

The Verlet algorithm is time-reversible. This important property is fulfilled also by the Gear 4th order method [2,3] treated as an implicit one, if $\mathbf{F}$ is independent of momentum. This is seen if we rearrange the final (corrected) formulas of the method into the fully equivalent form of: Eq. (5) and the Cowell–Numerov (C–N) formula [8]:

$$u=\begin{cases} 4\epsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12}-\left(\frac{\sigma}{r_{ij}}\right)^{6}\right]+A & \text{for} \quad r_{ij}/\sigma < R_{\mathrm{S}}=1.60 \\ B(r_{ij}/\sigma-R_{\mathrm{C}})^3[1+C(r_{ij}/\sigma-R_{\mathrm{C}})] & \text{for} \quad R_{\mathrm{S}}\le r_{ij}/\sigma\le R_{\mathrm{C}} \\ 0 & \text{for} \quad r_{ij}/\sigma > R_{\mathrm{C}}=2.0 \end{cases} \tag{9}$$

$$\mathbf{q}(t+\Delta t)=2\mathbf{q}(t)-\mathbf{q}(t-\Delta t)+\frac{\Delta t^2}{12m}[\mathbf{F}(t+\Delta t)+10\mathbf{F}(t)+\mathbf{F}(t-\Delta t)]+O(\Delta t^6) \tag{7}$$

Recently, it was shown [9] that the Cowell–Numerov method is symplectic, also.

At typical computer simulations, $\mathbf{F}(t)=\mathbf{F}(\mathbf{q}(t))$ and, the time-reversibility requires iterative solving of Eq. (7). An exact solving of Eq. (7) is a very time consuming. The procedure of approximate solving for potentials of the L–J type is presented in the Appendix. In this work, we will show that an application of the procedure makes the method of solving of Eq. (7) more efficient than Calvo and Sanz-Serena's 4th order method (CSS) [10,11], which is recommended [11] to be more efficient than the other explicit symplectic integrators.

Equation (7) is independent of momentums. Therefore, as in the case of Eq. (1), we can use much more accurate formula to define the momentum. For example:

$$\mathbf{p}(t)=\frac{m}{\Delta t}[\mathbf{q}(t+\Delta t)-\mathbf{q}(t)]-\frac{\Delta t}{360}[38\mathbf{F}(t+\Delta t)+171\mathbf{F}(t)-36\mathbf{F}(t-\Delta t)+7\mathbf{F}(t-2\Delta t)]+O(\Delta t^5) \tag{8}$$

The error of Eq. (8) $(-(17/1440)\Delta t^5\mathrm{d}^4\mathbf{F}/\mathrm{d}t^4+(37/10080)\Delta t^6\mathrm{d}^5\mathbf{F}/\mathrm{d}t^5)$ is typically much lower than Eq. (6), but the formula give the momentum at time t and not $t+\Delta t$ (like Eq. (5)), and so the total energy can be evaluated not before the next step is performed.

COMPUTER SIMULATIONS, RESULTS AND DISCUSSION

The computer simulations were performed using standard molecular dynamics (MD) constant volume and energy (NVE) method [2]. The simulated system consisted of $N = 32768$ identical particles interacting with shifted and specially truncated L–J potential:

where ϵ and σ are the L–J energy and length parameters, respectively, and the A, B, and C constants were adjusted so the second derivative of u was a continuous function for all interparticle distances, r_{ij}. All numerical values presented further are in reduced L–J units (i.e. for: $\sigma=\epsilon=m=1.$).

The short cut-off distance of Eq. (9) markedly decreases the time of computations. The way of truncation was not a typical one because we wanted the non-analyticity of the potential not to influence the results of the simulations performed by the 4th order methods. But, as a consequence, Eq. (9) is quite far from the non-truncated L–J potential [2]. Here, this does not influence the investigated effect because the dominant contribution to the step errors results from very close interparticle interactions. The shape of the potential for higher r_{ij} influences the effect very low.

The equations of motion were solved using the following methods: the velocity Verlet (Eqs. (1) and (2), denoted as VerV), Beeman's technique (Eqs. (1) and (5), denoted as VerB), the Gear 4th order predictor–corrector [2,3] (G4PC), the Gear 4th order implicit (approximate iteration of Eq. (7) + velocities from Eq. (5), denoted as G4IT), Calvo and Sanz-Serena's 4th order (CSS). The iteration of Eq. (7) (see the Appendix) was restricted only to the close neighbors ($r_{ij} < 1.25$), which saved the time of computation but resulted in a drift of energy, E. The drift of E/N per L–J time unit (denoted by $\delta\epsilon$) was many orders of magnitude lower than that for G4PC. In order to eliminate the drift (both for G4PC and G4IT) the particle velocities were scaled one per n_{corr} time steps.

Results of Simulations

An overwhelming majority of the simulations was performed at the state point, called the main point, of the numerical density $\rho = N/V = 0.75$ and the energy density $E/N = -1.17620 \pm 0.00002$. Three extra simulations were performed at $\rho = 0.50$ and $E/N = -0.18692 \pm 0.00001$. All the state points had approximately the same temperature $k_{\mathrm{B}}T \approx 1.26$. The results of all the simulations—the potential energy, U, the virial function, $\phi = \sum \mathbf{r}_{ij}(\partial U/\partial \mathbf{r}_{ij})$, and the parameters: ΔE_{k}, $\delta\epsilon$, n_{corr} are presented in Table I. ΔE_{k} presented in Table I was evaluated from: (a) for VerV, the difference in kinetic energies obtained from Eqs. (2) and (5); (b) for G4PC and G4IT, the difference in kinetic energies obtained from Eqs. (5) and (8); (c) for G4IT*, from Eq. (4) and the error of Eq. (8) evaluated during the evolutions performed by G4IT (the derivatives estimated to $O(\Delta t^2)$). The errors (in brackets) were estimated from the mean square deviations evaluated by dividing each evolution into 10 and 20 intervals. The first column of Table I presents the denotation of the "experimental point"—the sequence number preceded by the symbol of the method: VerV, VerB, G4PC, G4IT, and CSS. The "points" VerV*1–7 and G4IT* 3 and 4 were not simulated, but extrapolated from the corresponding values (X_{S} in Eq. (10)) of VerV and G4IT by using:

$$X = X_{\mathrm{S}} - \Delta E_{\mathrm{k}}(\partial X/\partial E)_{\mathrm{V}} \quad \text{for } X = U, \phi \tag{10}$$

The derivatives were obtained from fluctuations [12]:

$$(\partial U/\partial E)_{\mathrm{V}} = 0.3221(12), \quad (\partial \phi/\partial E)_{\mathrm{V}} = -5.47(3) \tag{11}$$

According to Eq. (10) and the way of evaluation of ΔE_{k}, VerV* corresponds to results' of Beeman's technique and G4IT* to results of G4IT with Eq. (5)

TABLE I Results of the simulations for various time steps (Δt) obtained by: the velocity Verlet method (VerV), Beeman's technique (VerB), the Gear 4th order predictor–corrector (G4PC), the Gear 4th order implicit (G4IT), and Calvo and Sanz-Serena's fourth-order method (CSS)

	Δt	ρ	E/N	U/N	$\phi/3V$	$\Delta E_{\mathrm{k}}/N$	$\delta\epsilon$	n_{corr}
VerV1	0.0015	0.75	− 1.17620	− 3.07576(7)	− 1.98344(35)	0.00038	–	–
VerV2	0.0020	0.75	− 1.17620	− 3.07562(6)	− 1.98444(29)	0.00068	–	–
VerV3	0.0030	0.75	− 1.17620	− 3.07551(7)	− 1.98466(32)	0.00154	–	–
VerV4	0.0040	0.75	− 1.17621	− 3.07514(5)	− 1.98626(35)	0.00273	–	–
VerV5	0.0060	0.75	− 1.17619	− 3.07440(4)	− 1.98859(24)	0.00615	–	–
VerV6	0.0080	0.75	− 1.17619	− 3.07328(5)	− 1.99248(35)	0.01091	–	–
VerV7	0.0100	0.75	− 1.17620	− 3.07167(6)	− 1.99831(33)	0.01702	–	–
VerV*1	0.0015	0.75	− 1.17620	− 3.07588(7)	− 1.98292(35)			
VerV*2	0.0020	0.75	− 1.17620	− 3.07584(6)	− 1.98351(29)			
VerV*3	0.0030	0.75	− 1.17620	− 3.07601(7)	− 1.98255(32)			
VerV*4	0.0040	0.75	− 1.17621	− 3.07602(5)	− 1.98253(35)			
VerV*5	0.0060	0.75	− 1.17619	− 3.07638(4)	− 1.98018(25)			
VerV*6	0.0080	0.75	− 1.17619	− 3.07679(5)	− 1.97756(36)			
VerV*7	0.0100	0.75	− 1.17620	− 3.07715(7)	− 1.97503(36)			
VerB7	0.0100	0.75	− 1.17618	− 3.07723(6)	− 1.97450(29)	0.00021	–	–
G4PC1	0.0040	0.75	− 1.17620	− 3.07578(6)	− 1.98370(32)	6.0×10^{-6}	-1.7×10^{-4}	80
G4PC2	0.0080	0.75	− 1.17622	− 3.07600(7)	− 1.98316(35)	0.00009	-9.5×10^{-3}	1
G4PC3	0.0125	0.75	− 1.17621	− 3.07636(6)	− 1.98266(33)	0.00049	-7.3×10^{-2}	1
G4IT1	0.0040	0.75	− 1.17620	− 3.07584(5)	− 1.98346(33)	6.0×10^{-6}	-9.0×10^{-9}	∞
G4IT2	0.0080	0.75	− 1.17620	− 3.07582(7)	− 1.98370(37)	0.00009	-1.9×10^{-7}	10000
G4IT3	0.0100	0.75	− 1.17621	− 3.07576(5)	− 1.98387(30)	0.00021	-1.3×10^{-6}	5000
G4IT4	0.0125	0.75	− 1.17619	− 3.07570(4)	− 1.98399(17)	0.00050	-1.4×10^{-5}	1000
G4IT*3	0.0100	0.75	− 1.17621	− 3.07583(5)	− 1.98358(30)	-1.4×10^{-5}		
G4IT*4	0.0125	0.75	− 1.17619	− 3.07586(4)	− 1.98331(17)	-4.6×10^{-5}		
CSS4	0.0125	0.75	− 1.17620	− 3.07579(4)	− 1.98368(15)		$<10^{-8}$	∞
G4IT5	0.0080	0.50	− 0.18693	− 2.08905(9)	− 0.10095(17)	0.00004	-3.5×10^{-7}	10000
VerV8	0.0100	0.50	− 0.18691	− 2.08739(8)	− 0.10326(18)	0.00828	–	–
VerB8	0.0100	0.50	− 0.18692	− 2.08911(7)	− 0.09927(17)	0.00009	–	–

U—the potential energy, $\phi = \sum \mathbf{r}_{ij}(\partial U/\partial \mathbf{r}_{ij})$—the Virial function, $\delta\epsilon$—the drift of the energy density, E/N, per L–J time unit, n_{corr}—typical number of the time steps performed without scaling velocities (∞ means that the velocity were not scaled at all, – means that $\delta\epsilon$ was not measured). The denotation of the method together with the sequence number is presented in the first column. The asterisk means that the results are obtained from extrapolation by using Eq. (10). The figure in parenthesis is the error in units of the last digit of the corresponding value.

replaced by Eq. (8). The possible error of this approximation is much lower than the investigated effects. The error is proportional to $(\Delta E_{\rm k})^2$ and for the highest $\Delta E_{\rm k}$ (for $\Delta t = 0.01$), the differences between VerV*7 and VerB7 are in the range of estimated errors. These differences are over 40 times lower than $\Delta E_{\rm k}(\partial X/\partial E)_{\rm V}$. Therefore, the possible extrapolation error for G4IT* is completely negligible when compared to the errors of G4IT. $\Delta E_{\rm k}$ for G4IT* were evaluated during the corresponding evolutions of G4IT.

All the simulations except of G4IT4 and CSS4 were performed for the total time 648.0–672.0 (L–J units). In the case of the two simulations, higher accuracy was required and the time amounted to 2560.0. For better presentation, the values of U/N obtained at the main point for various Δt are presented in Fig. 1.

U and ϕ for VerV1–VerV7 were fitted by the least square method to the following relation:

$$X = X_0 + A_X(\Delta t)^2 \quad \text{for } X = U, \phi \tag{12}$$

As a result we obtained:

$$\begin{aligned} &U_0/N = -3.07584 \quad A_U/N = 41.0 \quad \chi_U^2 = 1.1 \\ &\phi_0/3V = -1.98350 \quad A_\phi/3V = -146. \\ &\chi_\phi^2 = 1.3 \end{aligned} \tag{13}$$

In further considerations, U_0 and ϕ_0 of Eq. (13) will be treated as the reference values equal to the real values of U and ϕ of the main point. This assumption fully agrees with the results obtained by the Gear method (see Table I and Fig. 1).

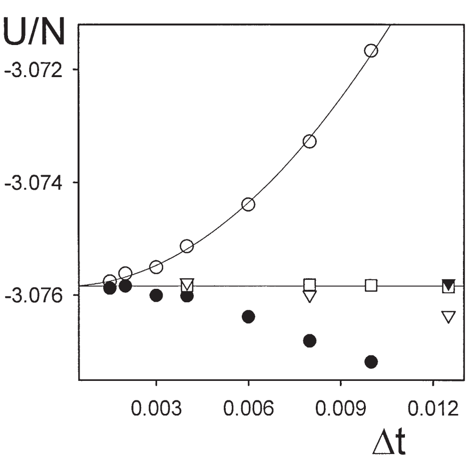

FIGURE 1 Potential energy density, U/N, as a function of Δt for various algorithms: empty circles—VerV, filled circles—VerV*, empty triangles—G4PC, empty squares—G4IT (for $\Delta t = 0.01$ and 0.0125, G4IT*), filled triangle—CSS. The curve presents Eq. (12) for $X = U$. The horizontal line corresponds to U_0/N from Eq. (13). The radius of the circles correspond to 1.2×10^{-4}, which is the doubled value of typical error from Table I.

The Verlet Method

The dependence of the results obtained by the Verlet velocity algorithm on Δt is evidently seen in Fig. 1 as well as in Table I. The deviations both of U and ϕ are very well described by Eq. (12) (low values of χ^2 of Eq. (13)). The kinetic energy of the state points obtained from VerV was evaluated from Eq. (2) so the deviations should be proportional to $\Delta E_{\rm k} \propto \Delta t^2$ (Ref. [5] and "Introduction" section of the paper). This is in agreement with the curve (12), but only qualitatively. The quantitative results show that it must be an additional effect because the deviations of U and ϕ do not correspond to the same value of $\Delta E_{\rm k}$. The ratio of $(\partial\phi/\partial E)_{\rm V}$ to $(\partial U/\partial E)_{\rm V}$ (from Eq. (11)) is not equal, but nearly 20% lower than the ratio of A_ϕ to A_U (from Eq. (13)). For VerV7 (see Table I), $(U - U_0)/(\partial U/\partial E)_{\rm V}$ exceeds $(\phi - \phi_0)/(\partial\phi/\partial E)_{\rm V}$ by nearly seven cumulative errors. The cause is the step error of Eq. (1), which results in a non-physical contribution to U and ϕ.

The influence of the step error on the obtained results is clearly seen for VerV*. $\Delta E_{\rm k}$ for VerV is nearly exactly proportional to Δt^2 (see Table I). As a result, U and ϕ of VerV*1–7 can be described by Eq. (12) with U_0, ϕ_0, and χ^2 very close to Eq. (13). The least square method gives, for VerV*, $A_U/N = -13.9$ and $A_\phi/3V = 87.2$, which are the values comparable with Eq. (13) (the replacement of VerV7* by VerB7 has no significant influence on the results of the minimization). The value of $\phi - \phi_0$ for VerB7 (Beeman's technique at $\Delta t = 0.01$) corresponds to the shift of pressure fully comparable with that for VerV7 (the Verlet velocity method) (−0.009 and 0.015, respectively). Therefore, at this thermodynamic conditions ($\rho = 0.75$) the influence of the step error of Eq. (1) is so strong that the use of Eq. (5) instead of Eq. (2) is aimless.

The effect quickly decreases with a decrease of density. For VerB8 and G4IT5 ($\rho = 0.50$ and the temperature approximately equal to that of the main point), the difference in U/N is lower than the single error. The difference in $\phi/3V$ is still non-negligible (about 0.0017), but it is over five times lower than the deviation for $\rho = 0.75$. For a comparison, $\Delta E_{\rm k}$ of VerV8 is only two times lower than that of VerV7. As a result, the difference in U/N between VerV7 and G4IT5 is still considerable. Therefore, the influence of the step error of Eq. (1) vanishes quicker with a decrease of ρ than the influence of $\Delta E_{\rm k}$. This confirms Beeman's technique to be more accurate than the Verlet velocity algorithm, but only at moderate or low densities.

Additional Test

The Verlet formula (1), the Cowell–Numerov (Eq. (7)), and the CSS method were additionally tested by performing series of central collisions of a pair of the L–J particles (Eq. (9)). The equation of motion was solved numerically by applying the three methods for various time steps. U and ϕ were measured at the same time intervals always equal to the maximal time step $\Delta t_{\max} = 0.01$. In order to eliminate an influence of initial conditions, the results were averaged over $n_s = 100$ collisions, each for the initial distance $r_0^i = R_C + (i-1)|v_0|\Delta t_{\max}/n_s$ for $i = 1, n_s$, where v_0—the initial velocity. The test was performed for $v_0 = -3.0$ and -5.0 (for the main point, $v_0^2/k_B T \approx 12.7$).

The results are presented in Table II. They confirm the results of MD simulations described in "The Verlet Method" section. The deviations of U and ϕ obtained by applying Eq. (1) are qualitatively in full agreement (sign, dependence $\propto (\Delta t)^2$) with that from Table I.

The Cowell–Numerov and the Gear Fourth Order Method

Table II shows great advantage of the Cowell–Numerov formula (7) over the Verlet one (Eq. (1)). The deviations of the measured averages are $O(\Delta t^4)$ and the value for $\Delta t = 0.01$ is about two orders of magnitude lower than in the case of Eq. (1). This is in full agreement with the results from Table I. For G4IT, the deviations of U and ϕ from U_0 and ϕ_0 are comparable with the estimated errors even for $\Delta t = 0.0125$. For G4IT4, the deviations are about three times higher than the error (which is especially low), but the main reason is the error of Eq. (5) and not of Eq. (7). The deviations for G4IT*4 are nearly three times lower than that of G4IT4 and the value of ΔE_k for Eq. (8) is negligible even for $\Delta t = 0.0125$ (G4IT*4). This shows that, especially if we are interested in a high accuracy, Eq. (5) should be replaced by Eq. (8). The advantage of Eq. (8) over Eq. (5) should increase with a decrease of ρ, like in the case of Beeman's and Verlet's methods ("The Verlet Method" section).

The advantage of formula (7) over formula (1) is so high because of the step error. The value of the error of Eq. (7) ($-(\Delta t^6/240m)\mathrm{d}^4\mathbf{F}/\mathrm{d}t^4$) is typically much lower than that of Eq. (1) ($(\Delta t^4/12m)\mathrm{d}^2\mathbf{F}/\mathrm{d}t^2$). As an example, for the central collision at $v_0 = -4.0$ the maximal absolute value of the step error for $\Delta t = 0.01$ amounts: 1.9×10^{-5} for Eq. (7) and 3.6×10^{-4} for Eq. (1). The other cause is that the fourth time derivative changes the sign more frequently than the second one. This decreases the accumulation of the step errors.

The step error of G4PC is formally very close to that of G4IT, but the method evaluates $\mathbf{F}_i$ only one per step. This leads to a considerably inconsistency between the left and the right hand side of Eq. (7) and, as a result, to a strong drift of the energy. The drift was eliminated by scaling the velocities. Table I and Fig. 1 clearly show that the scaling has an influence on U and ϕ. For G4PC3 ($\Delta t = 0.0125$), the velocities were multiplied each step by about 1.00025. As a result, the deviations of U and ϕ from U_0 and ϕ_0 are only a few times lower than that for the VerB7. The energy drift, $\delta\epsilon$, quickly vanishes with $\Delta t \rightarrow 0$. For $\Delta t = 0.008$ (G4PC2), $\delta\epsilon$ is nearly eight times lower than that for $\Delta t = 0.0125$ and, the deviations of U and ϕ are comparable with the errors.

The values of $\delta\epsilon$ for G4IT (see Table I) are not equal to zero. In practice, such small values of $\delta\epsilon$ are completely unimportant. The value of $\delta\epsilon$ for G4PCl is over 10 times higher than that for G4IT4, but its possible influence is still many times lower than the error. The main reason of the energy drift for G4IT is the approximated form of solving of Eq. (7) (see the Appendix). In fact, Eq. (7) is more stable than Eq. (1). The exact solving of Eq. (7) resulted, at the main point for $\Delta t = 0.0125$, in $\delta\epsilon = 5 \times 10^{-7}$. The Verlet method for this Δt gave $\delta\epsilon = 3 \times 10^{-6}$

TABLE II The deviations of averages of u and $r\partial u/\partial r$ as a function of Δt for the central collision of a pair of the L–J particles (Eq. (9)) for the initial velocities $v_0 = -3.0$ and -5.0

		$v_0 = -3.0$		$v_0 = -5.0$	
Method	Δt	$\delta\langle u\rangle$	$\delta\langle r\partial u/\partial r\rangle$	$\delta\langle u\rangle$	$\delta\langle r\partial u/\partial r\rangle$
Verlet	0.00125	1.3×10^{-5}	3.8×10^{-5}	3.3×10^{-5}	2.1×10^{-4}
	0.0025	5.5×10^{-5}	1.6×10^{-4}	1.4×10^{-4}	8.9×10^{-4}
	0.0050	2.2×10^{-4}	6.5×10^{-4}	5.6×10^{-4}	3.6×10^{-3}
	0.010	9.0×10^{-4}	2.6×10^{-3}	2.3×10^{-3}	1.5×10^{-2}
C–N	0.00125	1.5×10^{-9}	6.8×10^{-9}	8.1×10^{-9}	8.5×10^{-8}
	0.0025	2.4×10^{-8}	1.1×10^{-7}	1.3×10^{-7}	1.4×10^{-7}
	0.0050	3.8×10^{-7}	1.8×10^{-6}	2.1×10^{-6}	2.2×10^{-5}
	0.010	6.2×10^{-6}	2.8×10^{-5}	3.4×10^{-5}	3.6×10^{-4}
CSS	0.00125	1.1×10^{-9}	-2.1×10^{-8}	1.7×10^{-8}	-2.6×10^{-7}
	0.0025	1.8×10^{-8}	-3.4×10^{-7}	2.7×10^{-7}	-4.1×10^{-6}
	0.0050	3.0×10^{-7}	-5.4×10^{-6}	4.3×10^{-6}	-6.6×10^{-5}
	0.010	4.8×10^{-6}	-8.8×10^{-5}	7.0×10^{-5}	-1.1×10^{-3}

The Newton equation was solved numerically by using: (a) Eq. (1) − Verlet; (b) Eq. (7) + the procedure from the Appendix-C–N; (c) Calvo and Sanz-Serna's fourth-order explicit method-CSS. $\delta\langle x\rangle$ is the deviation of the average from the reference value obtained by using $\Delta t = 0.0003125$.

(which is the value equal to $\delta\epsilon$ for the exact solving of Eq. (7) for $\Delta t = 0.01375$). The exact solution of Eq. (7) was obtained by using the procedure from the Appendix with the point (c) performed three times.

According to Table I the deviations of the points CSS4 and G4IT*4 from U_0 and ϕ_0 are approximately the same. The approximation is very rough because the deviations are comparable with the estimated errors. Table II shows that, especially at strong collisions ($v_0 = -5.0$), the accuracy of C–N should be even slightly higher than that of CSS. But the accuracy of the MD results is markedly too low to confirm this possible advantage. Summarizing, the only we can say is that the accuracies of G4IT with Eq. (5) replaced by Eq. (8) and the CSS method are approximately the same.

The main disadvantage of the higher order symplectic methods [11] is the efficiency of computations. The CSS method requires four evaluation of total force to perform a single time step. Therefore, the ratio of the total time of performing single time step to the time of evaluation of total force amounts, for CSS, to about four. In the case of solving of Eq. (7), the ratio slightly depends on liquid structure (because the ratio of the number of close neighbors to the number of total neighbors depends on) but first off all it depends on R_C. The ratio quickly approaches to 1.0 with $R_C \rightarrow \infty$. For the simulations presented in this work, the value amounted to about 1.6. An extra (short) simulation at the main point for $R_C = 3.0$ gave the ratio of about 1.2. Therefore, if a large R_C is required the use of the proposed method can be even four times more efficient than that of CSS.

CONCLUDING REMARKS

It is obvious that in every case of numerical solving the equations of motion, the obtained properties are burdened with some errors caused by the step error. In this paper, the attention has been focused on the influence of the truncation error of the basic formulas for $\mathbf{q}(t + \Delta t)$ of the Verlet method (1) and the Gear 4th order method (7). The proposed method (approximate solving of Eq. (7), velocities from Eq. (8)) was also compared with Calvo and Sanz-Serena's method. The results of the simulations (the L–J liquid, mostly at $\rho = 0.75$ and $k_B T \approx 1.26$) can be summarized as follows: (a) for the Verlet algorithm the obtained values of potential energy and pressure noticeably depend on Δt (the pressure error of about 0.009 at the main point for Beeman's technique with $\Delta t = 0.01$). The effect is $O(\Delta t^2)$ and strongly depends on ρ. For $\rho = 0.50$, the pressure error is about five times lower than for $\rho = 0.75$; (b) for the Gear 4th order predictor–corrector method, the deviations of measured averages are mainly due not to the step errors but to the scaling of velocity, which is necessary to eliminate the energy drift. For $\Delta t = 0.0125$, this effect is only a few times weaker than that for the Verlet method but it quickly vanishes with $\Delta t \rightarrow 0$; (c) for the proposed method the effect is $O(\Delta t^4)$ and the amplitude for $\Delta t = 0.01$ is two orders of magnitude lower than that for the Verlet method for the same Δt. The accuracy of the method is approximately equal to that of CSS, but the proposed method is much faster (to four times for $R_C \rightarrow \infty$).

Summarizing, the influence of the step errors is not very high even in the case of the Verlet formula. The errors, however noticeable, should not have a significant influence as far as we are only interested in qualitative results. But, in the case of quantitative simulations, the influence of the step errors must not be neglected. For such simulations, the Verlet method is extremely non-effective. In order to attain the accuracy of the proposed method for $\Delta t = 0.0125$, the Verlet method requires $\Delta t \approx 0.002$.

The method of iterative solving of Eq. (7) is very accurate and stable. This predisposes the method to quantitative simulations. The problem is that the iteration required a few evaluations of the force per single time step, which is very time-consuming. Fortunately, for most kinds of intermolecular interactions, the problem can be resolved by restricting the iteration only to a close vicinity of the molecules. The example of such a procedure for interactions of the $\mathbf{r}^{-6}$ type is presented in the Appendix. The idea can be also adapted to a much more complex cases as for example water. However, dipole–dipole interaction is much stronger than the L–J one, but the general situation is similar. The interaction for the low distances (lower than for example 3.5–4 Å) is much stronger and much more rapidly varying than that for the higher distances.

The quantitative simulations require, usually, very long time simulations to attain the desired accuracy. Such simulations also require extra tests to estimate the actual accuracy level. For example, we have to estimate the effect described in this paper. As a result, the time of making and implementing the iteration procedure is very short when compared to the total time devoted to the problem. Therefore, the application of the implicit method based on the optimized iteration scheme seems to be a much better idea than, for example, the use of the original (without iteration) predictor–corrector method. Quantitative simulations may be also performed using a higher order simplectic method. The methods are simpler and easier to implement than the proposed implicit method, but especially if a high cut-off distance is required the advantage of this method is evident.

For a very wide range of thermodynamic conditions, the parameters used in the iteration procedure depend on the intermolecular potential only. The shape of the procedure does not depend either on the conditions or on the potential, subject to the condition that the general form of the potential is not changed. Therefore, the computer program, once written, can be easily modified and G4IT (preferably with Eq. (8) instead of Eq. (5)) can be used as a standard also for the qualitative simulations, especially that in many cases it is not sure if the performed simulations may be treated as qualitative only.

APPENDIX—THE PROCEDURE OF APPROXIMATED SOLVING OF EQ. (7)

The neighbor list [2] used in the procedure is divided into four sub-lists: for $r_{ij} < 0.96$, 1.10, 1.25, and R_C. The steps of the procedure starting at time t are as follows:

(a) evaluate, from Eq. (1), the predicted placement at time $t + \Delta t$ (denoted by $\mathbf{q}^0$);

(b) using the sub-list (for time t) for $r_{ij} < 1.1$, evaluate the force (denoted by $\mathbf{F}^R$) at time $t - \Delta t$ and $t + \Delta t$ (from $\mathbf{q}^0$) and improve the prediction of (a) by evaluating:

$$\mathbf{q}^1(t+\Delta t) = \mathbf{q}^0(t+\Delta t) + \frac{\Delta t^2}{12m}[\mathbf{F}^R(\mathbf{q}^0(t+\Delta t)) - 2\mathbf{F}^R(t) + \mathbf{F}^R(t-\Delta t)] \quad \text{(A1)}$$

(c) using $\mathbf{q}^1(t + \Delta t)$ construct the new sub-lists, evaluate $\mathbf{F}(\mathbf{q}^1(t + \Delta t))$ for $r_{ij} < R_C$ than evaluate $\mathbf{q}(t + \Delta t)$ from Eq. (7). Evaluate the force for $r_{ij} < 1.25$ and improve $\mathbf{q}(t + \Delta t)$;

(d) improve the result of (c) evaluating the force for $r_{ij} < 1.10$ and than for $r_{ij} < 0.96$. Save $\mathbf{F}^R(t + \Delta t)$ for further application (to evaluate $\mathbf{q}^1(t + 2\Delta t)$ by Eq. (A1));

References

[1] Verlet, L. (1967) "Computer experiments on classical fluids. I. Thermodynamical properties of Lennard–Jones molecules", *Phys. Rev.* **159**, 98.

[2] Allen, M.P. and Tildesley, D.J. (1987) Computer Simulations of Liquids (Clarendon Press, Oxford).

[3] Berendsen, J.C. and van Gunsteren, W.F. (1986) In: Ciccotti, G. and Hoover, W.G., eds, Molecular Dynamics Simulation of Statistical Mechanical Systems, Proceedings of the Enrico Fermi Summer School, Varenna 1985 (North Holland, New York).

[4] Isbister, D.J., Searles, D.J. and Evans, D.J. (1997) "Symplectic properties of algorithms and simulation methods", *Physica* **A240**, 105.

[5] MacGowan, D. and Heyes, D.M. (1988) "Large timesteps in molecular dynamics simulations", *Mol. Simul.* **1**, 277.

[6] Beeman, D. (1976) "Some multistep methods for use in molecular dynamics calculations", *J. Comput. Phys.* **20**, 130.

[7] Rodger, P.M. (1989) "On the accuracy of some common molecular dynamics algorithms", *Mol. Simul.* **3**, 263.

[8] Dahlquist, G. and Bjorck, A. (1974) Numerical Methods (Prentice-Hall, NJ).

[9] Skeel, R.D., Zhang, G. and Schlick, T. (1997) "A family of symplectic integrators: stability, accuracy, and molecular dynamics applications", *SIAM J. Sci. Comput.* **18**, 203.

[10] Okunbor, D.I. and Skeel, R.D. (1992) "Canonical numerical methods for molecular dynamics simulations", *J. Comput. Chem.* **15**, 72.

[11] Gray, S.K., Noid, D.W. and Sumpter, B.G. (1994) "Symplectic integrators for large-scale molecular dynamics simulations: a comparison of several explicit methods", *J. Chem. Phys.* **101**, 4062.

[12] Litniewski, M. (1990) "The microcanonical weight function: application to molecular dynamics simulations", *J. Phys. Chem.* **94**, 6472.