Condensed Matter Physics, 2002, Vol. 5, No. 3(31), pp. 369–390

CONDENSED MATTER PHYSICS

New optimized algorithms for molecular dynamics simulations

I.P.Omelyan [1,2], I.M.Mryglod [1,2], R.Folk [2]

[1] Institute for Condensed Matter Physics of the National Academy of Sciences of Ukraine, 1 Svientsitskii Str., 79011 Lviv, Ukraine

[2] Institute for Theoretical Physics, University of Linz, A-4040 Linz, Austria

Received March 29, 2002

The method of molecular dynamics (MD) is a powerful tool for the prediction and investigation of various phenomena in physics, chemistry and biology. The development of efficient MD algorithms for integration of the equations of motion in classical and quantum many-body systems should therefore impact a lot of fields of fundamental research. In the present study it is shown that most of the existing MD integrators are far from being ideal and further significant improvement in the efficiency of the calculations can be reached. As a result, we propose new optimized algorithms which allow to reduce the numerical uncertainties to a minimum with the same overall computational costs. The optimization is performed within the well recognized decomposition approach and concerns the widely used symplectic Verlet-, Forest-Ruth-, Suzuki- as well as force-gradient-based schemes. It is concluded that the efficiency of the new algorithms can be achieved better with respect to the original integrators in factors from 3 to 1000 for orders from 2 to 12. This conclusion is confirmed in our MD simulations of a Lennard-Jones fluid for a particular case of second- and fourth-order integration schemes.

Key words: *molecular dynamics, symplectic integration, decomposition method, splitting method, order conditions, numerical schemes*

PACS: *02.60.Cb, 02.70.Ns, 05.10.-a, 95.10.Ce, 95.75.Pq*

1. Introduction

Modelling various physical and chemical processes in molecular dynamics (MD) simulations, we come to the necessity of integrating the equations of motion for a many-body system of interacting particles. A lot of numerical algorithms have been devised and implemented over the years to perform such an integration. The traditional high-order explicit Runge-Kutta (RK) and implicit predictor-corrector (PC) schemes [1,2] were applied in early investigations. It was soon realized that the extra orders obtained in these schemes are not relevant since the truncation errors accumulate too rapidly on MD scales of time [3]. This high instability restricts the application of RK and PC integrators in long-term MD simulations to very small time steps only, and, thus, reduces significantly the efficiency of the computations. In addition, the RK and PC algorithms produce solutions which, unlike exact phase trajectories, are neither symplectic nor time reversible.

In 1990, a new approach to the integration of motion in classical and quantum systems was proposed [4,5]. Within this approach the exponential operator of time evolution is decomposed into analytically solvable parts. As a result, the symplectic map of the flow of particles and time reversibility can be conserved exactly, despite an approximate character of the produced trajectories. The preservation of symplecticity and reversibility appears to be very important because, as is now well established, this closely relates to the stability of an algorithm [6]. The stability means that the numerical errors arising during the decomposition integration are bounded even for large sizes of the time step. This is contrary to the RK and PC schemes, where the truncation uncertainties increase linearly with the increase of the time of integration [3,7–10]. The decomposition algorithms are, therefore, ideal for long-duration evaluations in MD [6] and in astrophysical [11] simulations.

The construction of MD algorithms within the decomposition approach has been the subject of many investigations [4,5,7,12–20]. However, much of the previous work has been of a heuristic nature. Moreover, any freedom in the choice of schemes has been eliminated by restricting attention to particular schemes, mainly to those which lead to a minimal number of decomposition stages at a given order. The most notorious example is the widely used Verlet algorithm [21,22], which corresponds to a three-stages decomposition scheme of the second order. The fourth-order algorithm by Forest and Ruth [5] relates to a scheme with seven single-exponential stages. The sixth-order propagations are available [13,16] beginning from such fifteen stages. Higher-order integrators can be obtained by composing schemes of lower orders [4,12,14,16,23–26].

Recently, a more consequent analysis of the exponential factorization process has allowed to extend the family of analytically integrable decomposition algorithms by new members from the so-called force-gradient class [15,17,18]. But again, as in the non-gradient case, the study has been restricted to particular solutions, actually to the fourth order only. For example, Chin [18] introduced a set of five- and seven-stages of such force-gradient solutions. In order to derive the algorithms of higher orders, it has been proposed to apply the crude iterative procedure [27]. Advantages of the force-gradient integrators over their standard non-gradient (including Verlet and Forest-Ruth) counterparts have been demonstrated in celestial [18,27], stochastic [28,29], and quantum [30] dynamics simulations.

The question of how to improve the efficiency of the decomposition integration for atomic systems with long-range interactions has also been considered. As a result, the so-called multiple time scale integrators have been introduced [22,31]. In these integrators, the slow subdynamics is treated in a specific way using larger step sizes in view of the weakness of long-range forces. The faster motion, caused by the interactions at short interparticle distances, remains to be integrated using the original decomposition algorithms, such as the Verlet integrator. In addition, it has been shown that the decomposition approach can be adapted to integrate not only the translational motion in atomic liquids, but also to simulate more complicated molecular and spin systems at the presence of orientational degrees of freedom [7, 19,20].

Despite these advances, there has been very little work done on the systematic construction of decomposition integrators. Some results in this direction have been obtained by Li [16], but he restricted ourselves to the most obvious solutions. In addition, his consideration was limited to the non-gradient class exclusively, omitting the case of force-gradient algorithms. Note that a complete systematization of decomposition integrators presents not only a methodological interest but may have an important practical significance, because the new solutions found can be more efficient than the already known ones. In particular, as has been shown quite recently [32–34], the decomposition propagators with minimal number of stages do not necessarily lead to an optimal performance.

The primary goal of this work is to provide a comprehensive study of a whole family of symplectic self-adjoint decomposition integrators. Note that the self-adjointness should be fulfilled to reproduce the property of time reversibility in the solutions for non-dissipative systems (in the case of stochastic dynamics the same is needed to recover the principle of detailed balance). In this context, we should mention another family of symplectic self-adjoint schemes developed within an extended Runge-Kutta approach [35]. These schemes, however, are implicit and require the systems of nonlinear equations to be solved by expensive iterations at each step of the integration process. Since such equations cannot be solved exactly (because the number of the iterations is finite), the symplecticity and time reversibility will be violated. These disadvantages are absent in the present decomposition method, where the time propagation is performed explicitly by a set of canonical transformations. Another benefit of our approach is that we work with the most general solutions and, thus, are able to easily carry out an optimization with respect to their efficiency. The optimization is reached by imposing additional constraints, such as minimizing the principal error coefficients, and thus it allows us to choose the best integrator for each given order.

2. Theoretical background

2.1. Basic equations of motion in classical and quantum systems

For classical systems, the equations of motion can be cast in the following compact form

$$\frac{\mathrm{d}\boldsymbol{\rho}}{\mathrm{d}t} = [\boldsymbol{\rho} \circ H] \equiv L\boldsymbol{\rho}(t)\,, \tag{1}$$

where $\boldsymbol{\rho}$ denotes the set of phase variables, $[\,\circ\,]$ is the Poisson bracket, and H represents the Hamiltonian function. In the case of N classical particles interacting through the pair-wise potential $\varphi(r_{ij}) \equiv \varphi(|\mathbf{r}_i - \mathbf{r}_j|)$, the explicit expression for the Hamiltonian is

$$H = \sum_{i=1}^{N} \frac{m\mathbf{v}_i{}^2}{2} + \frac{1}{2}\sum_{i\neq j}^{N} \varphi(r_{ij}) \equiv T + U\,. \tag{2}$$

Here $\mathbf{r}_i$ denotes the position of particle i ($i = 1, 2, \ldots, N$) moving with the velocity $\mathbf{v}_i = \mathrm{d}\mathbf{r}_i/\mathrm{dt}$ and carrying mass m, so that T and U are the total kinetic and potential energies, respectively. Then $\boldsymbol{\rho} = \{\mathbf{r}_i, \mathbf{v}_i\} \equiv \{\mathbf{r}, \mathbf{v}\}$, and the Liouville operator of the system takes the form

$$L = \sum_{i=1}^{N} \Big(\mathbf{v}_i \cdot \frac{\partial}{\partial \mathbf{r}_i} + \frac{\mathbf{f}_i}{m} \cdot \frac{\partial}{\partial \mathbf{v}_i} \Big)\,, \tag{3}$$

where $\mathbf{f}_i = -\sum_{j(j\neq i)}^{N} \varphi'(r_{ij})\mathbf{r}_{ij}/r_{ij}$ are the forces acting on the particles due to the interactions.

For quantum systems, the state evolution can be described by the time-dependent Schrödinger equation

$$\mathrm{i}\hbar \frac{\partial \psi}{\partial t} = \mathcal{H}(\mathbf{r})\psi \equiv \Big(\mathcal{T} + \mathcal{U}(\mathbf{r})\Big)\psi(t)\,, \tag{4}$$

where $\mathcal{T} = -\frac{1}{2}\sum_{i=1}^{N} \hbar^2 \boldsymbol{\nabla}_i^2/m$ and $\mathcal{U}$ are the kinetic and potential energy operators, respectively, and ψ is the wave function. The stochastic [28,29] and hybrid quantum-classical [36] dynamics models can also be introduced (in the latter case we come to a coupled system of Newtonian (1) and Schrödinger (4) equations). But, for simplicity, we will deal only with the above purely classic and quantum considerations.

If an initial configuration $\boldsymbol{\rho}(0)$ or $\psi(0)$ is specified, the unique solution to equation (1) or (4) can formally be presented as

$$\boldsymbol{\mathcal{R}}(t) = \mathrm{e}^{\mathcal{L}t}\boldsymbol{\mathcal{R}}(0) \equiv \Big(\mathrm{e}^{\mathcal{L}\Delta t}\Big)^{l}\boldsymbol{\mathcal{R}}(0)\,, \tag{5}$$

where Δt is the size of the time step, $l = t/\Delta t$ is the total number of steps, $\boldsymbol{\mathcal{R}}$ denotes either $\boldsymbol{\rho}$ or ψ, and $\mathcal{L}$ corresponds to L or $-\mathrm{i}\mathcal{H}/\hbar$. It is well known that the time evolution of many-particle systems ($N > 2$) cannot be carried out exactly. So that the only way of handling the problem (5) is to perform the calculations by numerical methods.

2.2. Symplectic self-adjoint decomposition schemes

The basic idea of a decomposition approach is to factor out the exponential propagator $\mathrm{e}^{\mathcal{L}\Delta t}$ on such suboperators which allow to be presented analytically or at least in quadratures. This is achieved by splitting the operator $\mathcal{L} = \mathcal{A} + \mathcal{B}$ into its kinetic $\mathcal{A}$ and potential $\mathcal{B}$ parts, where $\mathcal{A} = \mathbf{v}\cdot\partial/\partial\mathbf{r}$ or $\mathcal{A} = -\mathrm{i}\mathcal{T}/\hbar$ and $\mathcal{B} = \mathbf{a}\cdot\partial/\partial\mathbf{v}$ with $\mathbf{a} \equiv \{\mathbf{a}_i\} = \{\mathbf{f}_i/m\}$ being the acceleration or $\mathcal{B} = -\mathrm{i}\mathcal{U}/\hbar$ for the cases of classical or quantum mechanics, respectively. Then the total propagator can be decomposed as

$$\mathrm{e}^{(\mathcal{A}+\mathcal{B})\Delta t+\mathcal{O}(\Delta t^{K+1})} = \prod_{p=1}^{P} \mathrm{e}^{\mathcal{A}a_p\Delta t}\mathrm{e}^{\mathcal{B}b_p\Delta t+\mathcal{C}c_p\Delta t^3}, \tag{6}$$

where $\mathcal{C} \equiv [\mathcal{B},[\mathcal{A},\mathcal{B}]]$ is the auxiliary operator and [,] denotes the commutator of two operators. The coefficients a_p, b_p, and c_p have to be chosen in such a way as to provide the highest possible value for the order $K \geqslant 1$ at a given integer number $P \geqslant 1$. As a result, taking into account the smallness of Δt, the integration (5) can be performed approximately using the equation (6) by neglecting truncation terms $\mathcal{O}(\Delta t^{K+1})$. The precision of the integration will improve with the increase of the order K and with the decrease of the size Δt of the time step. Formula (6) represents the decomposition approach in its most general form [34]. For $c_p \equiv 0$, equation (6) reduces to the usual non-gradient factorization [4,5,12,14,15].

In order to show that the exponential subpropagators arising on the right-hand-side of equation (6) are indeed analytically integrable, let us first consider more in detail the structure of the third-order operator $\mathcal{C}$. In view of the expressions for operators $\mathcal{A}$ and $\mathcal{B}$, it can be readily obtained in the case of classical systems that

$$\mathcal{C} \equiv [\mathcal{B},[\mathcal{A},\mathcal{B}]] = \sum_{i=1}^{N} \frac{\mathbf{g}_i}{m}\cdot\frac{\partial}{\partial \mathbf{v}_i} \equiv \mathbf{G}\cdot\frac{\partial}{\partial \mathbf{v}}\,, \tag{7}$$

where $\mathbf{g}_{i\alpha} = 2\sum_{j\beta}(\mathbf{f}_{j\beta}/m)\ \partial\mathbf{f}_{i\alpha}/\partial\mathbf{r}_{j\beta}$. The force-gradient evaluations $\partial\mathbf{f}_{i\alpha}/\partial\mathbf{r}_{j\beta}$ can also be explicitly represented because of $\mathbf{f}_{i\alpha} = -\sum_{j(j\neq i)}\varphi'(r_{ij})(\mathbf{r}_{i\alpha}-\mathbf{r}_{j\alpha})/r_{ij}$. This results in

$$\begin{aligned}\mathbf{g}_i &= -2\sum_{j(j\neq i)}^{N}\left[\Big(\mathbf{a}_i-\mathbf{a}_j\Big)\frac{\varphi'_{ij}}{r_{ij}} + \frac{\mathbf{r}_{ij}}{r_{ij}^3}\Big(r_{ij}\varphi''_{ij}-\varphi'_{ij}\Big)\Big(\mathbf{r}_{ij}\cdot(\mathbf{a}_i-\mathbf{a}_j)\Big)\right] \\ &\equiv \sum_{j(j\neq i)}^{N}\mathbf{g}(\mathbf{r}_{ij}) = \mathbf{g}_i(\mathbf{r})\,,\end{aligned} \tag{8}$$

where $\varphi'_{ij} \equiv \varphi'(r_{ij}) = \mathrm{d}\varphi(r_{ij})/\mathrm{d}r_{ij}$ and $\varphi''_{ij} = \mathrm{d}\varphi'_{ij}/\mathrm{d}r_{ij}$. As can be seen, the function $\mathbf{G}$ like $\mathbf{a}$ does not depend on velocity and, thus, the operator $\mathcal{C}$ will commute with $\mathcal{B}$. Then, taking also into account the independence of $\mathbf{v}$ on $\mathbf{r}$, yields the following two equalities

$$\begin{aligned}\mathrm{e}^{\mathcal{A}a_p\Delta t}\{\mathbf{r},\mathbf{v}\} &= \{\mathbf{r}+a_p\mathbf{v}\Delta t,\mathbf{v}\}\,, \\ \mathrm{e}^{\mathcal{B}b_p\Delta t+\mathcal{C}c_p\Delta t^3}\{\mathbf{r},\mathbf{v}\} &= \{\mathbf{r},\mathbf{v}+b_p\mathbf{a}\Delta t+c_p\mathbf{G}\Delta t^3\}\end{aligned} \tag{9}$$

that represent simple shifts in position and velocity spaces, respectively [34]. For quantum systems, where $\mathcal{C} = \mathrm{i}\sum_i|\boldsymbol{\nabla}_i\mathcal{U}|^2/(\hbar m)$, the calculation of $\mathrm{e}^{\mathcal{B}b_p\Delta t+\mathcal{C}c_p\Delta t^3}$ also presents no difficulties (at least for particles in external fields), because this requires only the knowledge of the potential and its gradient. The kinetic part $\mathrm{e}^{\mathcal{A}a_p\Delta t}$ will require carrying out two, one direct and one inverse, spatial Fourier transforms [30].

An important feature of the decomposition integration (6) is its exact conservation of the symplectic map of the flow of particles in the phase space. This follows from the fact that separate shifts (9) of positions and velocities do not change the phase volume. The property $\mathcal{S}^{-1}(t) = \mathcal{S}(-t)$ of time reversibility of evolution operators $\mathcal{S}(t) = \mathrm{e}^{\mathcal{L}t}$ can also be reproduced perfectly by imposing self-adjoint constraints on the coefficients a_p, b_p, and c_p. There are two kinds of such constraints [33], namely, $a_1 = 0$, $a_{p+1} = a_{P-p+1}$, $b_p = b_{P-p+1}$, $c_p = c_{P-p+1}$, as well as $a_p = a_{P-p+1}$, $b_p = b_{P-p}$, $c_p = c_{P-p}$ with $b_P = 0$ and $c_P = 0$. Then single-exponential subpropagators will enter symmetrically into the decomposition (6), providing automatically the required reversibility. The existence of the two kinds of constraints is caused by the two kinds of single-exponential operators (9) and the fact that the symmetry is invariant with respect to the replacement of these operators between themselves. Note also that within the self-adjoint schemes, the total number of single-exponential operators (stages) in equation (6) is actually equal to $S = 2P - 1$, i.e. it takes only odd values (because one of the boundary set of coefficients is set to zero, $a_1 = 0$ or $b_P = c_P = 0$).

2.3. Order conditions

The above symmetry will result in the automatic disappearing of all even-order terms in the error function of decomposition transformation (6), i.e.,

$$\mathcal{O}(\Delta t^{K+1}) = \mathcal{O}_1 \Delta t + \mathcal{O}_3 \Delta t^3 + \mathcal{O}_5 \Delta t^5 + \mathcal{O}_7 \Delta t^7 + \ldots + \mathcal{O}_{K+1} \Delta t^{K+1} + \ldots . \tag{10}$$

We see, therefore, that the order K of self-adjoint algorithms may accept only even numbers ($K = 2, 4, 6, 8, \ldots$). The cancellation of the remaining odd-order terms (up to $\mathcal{O}_{K-1}\Delta t^{K-1}$ for the order K) has to be provided by fulfilling a set of basic conditions for a_p, b_p, and c_p. Let us write down explicit expressions for the functions $\mathcal{O}_1$, $\mathcal{O}_3$, $\mathcal{O}_5$, and $\mathcal{O}_7$ (this will be enough to derive algorithms up to the eighth order). Expanding both sides of equation (6) into Taylor's series, and collecting the terms with the same powers of Δt one finds:

$$\mathcal{O}_1 = (\nu - 1)\mathcal{A} + (\sigma - 1)\mathcal{B}\,, \qquad \mathcal{O}_3 = \alpha[\mathcal{A},[\mathcal{A},\mathcal{B}]] + \beta[\mathcal{B},[\mathcal{A},\mathcal{B}]]\,, \tag{11}$$

$$\begin{aligned} \mathcal{O}_5 \;=\; & \gamma_1[\mathcal{A},[\mathcal{A},[\mathcal{A},[\mathcal{A},\mathcal{B}]]]] + \gamma_2[\mathcal{A},[\mathcal{A},[\mathcal{B},[\mathcal{A},\mathcal{B}]]]] \\ & + \gamma_3[\mathcal{B},[\mathcal{A},[\mathcal{A},[\mathcal{A},\mathcal{B}]]]] + \gamma_4[\mathcal{B},[\mathcal{B},[\mathcal{A},[\mathcal{A},\mathcal{B}]]]]\,, \end{aligned} \tag{12}$$

$$\begin{aligned} \mathcal{O}_7 \;=\; & \zeta_1[\mathcal{B},[\mathcal{B},[\mathcal{A},[\mathcal{B},[\mathcal{A},[\mathcal{B},\mathcal{A}]]]]]] + \zeta_2[\mathcal{B},[\mathcal{B},[\mathcal{B},[\mathcal{A},[\mathcal{A},[\mathcal{B},\mathcal{A}]]]]]] \\ & + \zeta_3[\mathcal{B},[\mathcal{B},[\mathcal{A},[\mathcal{A},[\mathcal{A},[\mathcal{B},\mathcal{A}]]]]]] + \zeta_4[\mathcal{B},[\mathcal{A},[\mathcal{B},[\mathcal{A},[\mathcal{A},[\mathcal{B},\mathcal{A}]]]]]] \\ & + \zeta_5[\mathcal{A},[\mathcal{B},[\mathcal{B},[\mathcal{A},[\mathcal{A},[\mathcal{B},\mathcal{A}]]]]]] + \zeta_6[\mathcal{A},[\mathcal{B},[\mathcal{A},[\mathcal{B},[\mathcal{A},[\mathcal{B},\mathcal{A}]]]]]] \\ & + \zeta_7[\mathcal{B},[\mathcal{A},[\mathcal{A},[\mathcal{A},[\mathcal{A},[\mathcal{B},\mathcal{A}]]]]]] + \zeta_8[\mathcal{A},[\mathcal{B},[\mathcal{A},[\mathcal{A},[\mathcal{A},[\mathcal{B},\mathcal{A}]]]]]] \\ & + \zeta_9[\mathcal{A},[\mathcal{A},[\mathcal{B},[\mathcal{A},[\mathcal{A},[\mathcal{B},\mathcal{A}]]]]]] + \zeta_{10}[\mathcal{A},[\mathcal{A},[\mathcal{A},[\mathcal{A},[\mathcal{A},[\mathcal{B},\mathcal{A}]]]]]]\,. \end{aligned} \tag{13}$$

Here we take into account the equality $[\mathcal{B},\mathcal{C}] = 0$ following from the commutation of operators $\mathcal{B}$ and $\mathcal{C}$, so that any occurrence of combinations containing the chain

$[\mathcal{B}, [\mathcal{B}, [\mathcal{A}, \mathcal{B}]]]$ has been ignored (in particular, for the fifth-order truncation term $\mathcal{O}_5$, this has allowed us to exclude the two zero-valued commutators $[\mathcal{B}, [\mathcal{B}, [\mathcal{B}, [\mathcal{A}, \mathcal{B}]]]]$ and $[\mathcal{A}, [\mathcal{B}, [\mathcal{B}, [\mathcal{A}, \mathcal{B}]]]]$). The multipliers ν, σ, α, β, γ_{1-4}, and ζ_{1-10}, arising in equations (11)–(13), will depend on the coefficients a_p, b_p, and c_p, where $p = 1, 2, \ldots, P$. The forms of this dependency for the first two multipliers are particularly simple and look like $\nu = \sum_{p=1}^{P} a_p$ and $\sigma = \sum_{p=1}^{P} b_p$. Explicit expressions for α, β, γ_{1-4}, and ζ_{1-10} as functions of $\{a_p, b_p, c_p\}$ can be found in our recent work [34].

So that putting $\nu = 1$ and $\sigma = 1$ will cancel the first-order truncation uncertainties, $\mathcal{O}_1 = 0$. The next multipliers α, β, γ_{1-4}, ζ_{1-10} should be set to zero to kill higher-order truncation terms, namely, $\mathcal{O}_3$, $\mathcal{O}_5$, and $\mathcal{O}_7$ (see equations (11)–(13)). In such a way, we come to a set of the required order conditions. They present, in fact, a system of non-linear equations which should be solved with respect to the time coefficients a_p, b_p, and c_p.

3. Optimization of decomposition schemes

The main idea of our approach lies in the fact that for each given order, we can always introduce an extended set of single-exponential time coefficients with one or more free parameters. Then, imposing an additional constraint related to the norm of the principal error coefficient, the best solutions can be chosen by minimizing that norm with respect to these parameters. In this section we will first describe in detail the optimization of the most known non-gradient Verlet [21,22] and Forest-Ruth [5] integrators of orders 2 and 4, respectively. The final results on the optimization of the schemes with higher orders and stage numbers, including the case of force-gradient integrators, will also be presented.

3.1. Verlet-like algorithms of the second order

The method just highlighted is quite general to build numerical integrators of arbitrary orders. In particular, the second-order ($K = 2$) velocity Verlet (VV) algorithm [21,22]

$$\mathrm{e}^{(\mathcal{A}+\mathcal{B})\Delta t+\mathcal{O}(\Delta t^3)} = \mathrm{e}^{\mathcal{B}\frac{\Delta t}{2}}\mathrm{e}^{\mathcal{A}\Delta t}\mathrm{e}^{\mathcal{B}\frac{\Delta t}{2}} \tag{14}$$

is immediately reproduced from equation (6) at $P = 2$ and $a_1 = 0$, $b_1 = b_2 = 1/2$, $a_2 = 1$ as well as $c_1 = c_2 = 0$. The case when the operators $\mathcal{A}$ and $\mathcal{B}$ are replaced by each other ($\mathcal{A} \leftrightarrow \mathcal{B}$) is also possible, and we come to the so-called position-Verlet (PV) algorithm [22], $\mathrm{e}^{(\mathcal{A}+\mathcal{B})\Delta t+\mathcal{O}(\Delta t^3)} = \mathrm{e}^{\mathcal{A}\Delta t/2}\mathrm{e}^{\mathcal{B}\Delta t}\mathrm{e}^{\mathcal{A}\Delta t/2}$, corresponding to the choice $a_1 = a_2 = 1/2$, $b_1 = 1$, and $b_2 = 0$.

Despite the fact that the method of construction of time-reversible integrators using symplectic decompositions is not new, some important cases have never been considered and have been completely ignored in the literature. This concerns, in particular, the following question. Are the above Verlet algorithms optimal in view of the time efficiency among all possible basic (i.e. with single splitting of the Liouville operator) decomposition integrators of the second-order? We can say only that the Verlet algorithms do minimize the number of force evaluations per time step.

However, as will be shown below, this does not guarantee the optimization with respect to the overall number of force recalculations (the most time-consuming part of MD simulations), which are necessary to perform during a fixed observation time in order to achieve a given precision in solutions.

It can be readily seen that the Verlet algorithms ($P = 2$) require only one ($P-1 = 1$) force evaluation per time step Δt, whereas the fourth- and higher-order schemes ($P \geqslant 4$) need three or more such evaluations (see the next subsections 3.2 and 3.3). Let us consider now the intermediate case $P = 3$ which leads to an extended self-adjoint propagation in the form

$$\mathrm{e}^{(\mathcal{A}+\mathcal{B})\Delta t+\mathcal{O}_3\Delta t^3+\mathcal{O}(\Delta t^5)} = \mathrm{e}^{\mathcal{A}\xi\Delta t}\mathrm{e}^{\mathcal{B}\frac{\Delta t}{2}}\mathrm{e}^{\mathcal{A}(1-2\xi)\Delta t}\mathrm{e}^{\mathcal{B}\frac{\Delta t}{2}}\mathrm{e}^{\mathcal{A}\xi\Delta t} \tag{15}$$

following from equation (6) at $a_1 = a_3 \equiv \xi$, $a_2 = 1-2\xi$, $b_1 = b_2 = 1/2$, and $b_3 = 0$ (with $c_1 = c_2 = c_3 = 0$). Again, the propagation with $\mathcal{A} \leftrightarrow \mathcal{B}$ is also acceptable (then $a_1 = 0$, $b_1 = b_3 \equiv \xi$, $b_2 = 1-2\xi$ and $a_2 = a_3 = 1/2$). Formula (15) represents a whole family of symplectic time-reversible integrators of the second-order in which a particular member can be extracted by choosing a value for a free parameter ξ. For $\xi = 0$, equation (15) reduces to the VV (see equation (14)) or PV (at $\mathcal{A} \leftrightarrow \mathcal{B}$) algorithm. The extended (when $\xi \neq 0$) propagation requires already two, instead of one, force recalculation per time step. For this reason, we can come to an incorrect conclusion that such a propagation has no advantage over the Verlet algorithms.

In order to prove that the above conclusion is indeed incorrect, let us analyze more in detail the influence of truncation errors $\mathcal{O}_3\Delta t^3$ on the result. The Taylor's Δt-expansion of equation (15) yields

$$\alpha(\xi) = \frac{1-6\xi+6\xi^2}{12}\,, \qquad \beta(\xi) = \frac{1-6\xi}{24}\,, \tag{16}$$

so that the norm of $\mathcal{O}_3$ with respect to the third-order commutators $[\mathcal{A},[\mathcal{A},\mathcal{B}]]$ and $[\mathcal{B},[\mathcal{A},\mathcal{B}]]$ (see equation (11)) is

$$d(\xi) = \sqrt{\alpha^2(\xi)+\beta^2(\xi)}\,. \tag{17}$$

Then, the norm of local uncertainties appearing during a single-step propagation can be expressed in terms of d and Δt as $g = d\Delta t^3$. During a whole integration over a fixed time interval t, the total number l of such single steps is proportional to Δt^{-1} (see equation (5)). As a result, the local third-order uncertainties will accumulate step by step leading, at $t \gg \Delta t$, to the second-order global errors $\Gamma_2 = g\Delta t^{-1}$, i.e.,

$$\Gamma_2(\xi,\Delta t) = d(\xi)\Delta t^2\,. \tag{18}$$

Extended propagation (15) can now be optimized with respect to ξ by minimizing the function $d(\xi)$. As can be readily verified, the minimum of $d(\xi)$ is achieved at $\xi = \xi_0$, where

$$\begin{aligned}\xi_0 &= \frac{1}{2} - \frac{(2\sqrt{326}+36)^{1/3}}{12} + \frac{1}{6(2\sqrt{326}+36)^{1/3}} \\ &\approx 0.1931833275037836\,,\end{aligned} \tag{19}$$

with $d_{\min} \equiv d(\xi_0) \approx 0.00855$. On the other hand, the value $d(0)$ of d corresponding to the Verlet algorithms (when $\xi = 0$) is equal to $d(0) \approx 0.0932$, i.e., it increases in $d(0)/d(\xi_0) \approx 11$ times. Remembering that the extended propagation requires two force evaluation per time step Δt, it should be performed with double step size $2\Delta t$ with respect to that of the Verlet algorithms, in order to provide the same number of total force recalculations during the integration over the same time interval. Therefore, the extended propagation will be more efficient if the following inequality $\Gamma_2(\xi, 2\Delta t) < \Gamma_2(\xi = 0, \Delta t)$ takes place. Taking into account equation (18), such an inequality can be rewritten as $d(0)/d(\xi) > 4$, and thus it is fulfilled completely in the optimization regime. In particular,

$$\frac{\Gamma_2(\xi_0, 2\Delta t)}{\Gamma_2(\xi = 0, \Delta t)} \approx 0.367\,, \tag{20}$$

indicating that *the optimized propagation, being applied even with double sizes of the time step, will reduce the global errors approximately three times.*

In view of equations (9) and (15), more explicit expressions for the single-step propagation of position and velocity from time t to $t + \Delta t$ within the optimized VV-like algorithm are:

$$\begin{aligned}
\mathbf{r}_{\mathrm{I}} &= \mathbf{r}(t) + \xi \mathbf{v}(t)\Delta t\,,\\
\mathbf{v}_{\mathrm{I}} &= \mathbf{v}(t) + \mathbf{f}[\mathbf{r}_{\mathrm{I}}]\Delta t/(2m)\,,\\
\mathbf{r}_{\mathrm{II}} &= \mathbf{r}_{\mathrm{I}} + (1-2\xi)\mathbf{v}_{\mathrm{I}}\Delta t\,,\\
\mathbf{v}(t+\Delta t) &= \mathbf{v}_{\mathrm{I}} + \mathbf{f}[\mathbf{r}_{\mathrm{II}}]\Delta t/(2m)\,,\\
\mathbf{r}(t+\Delta t) &= \mathbf{r}_{\mathrm{II}} + \xi \mathbf{v}(t+\Delta t)\Delta t\,,
\end{aligned} \tag{21}$$

whereas the optimized PV-like algorithm (when $\mathcal{A} \leftrightarrow \mathcal{B}$ in equation (15)) reads:

$$\begin{aligned}
\mathbf{v}_{\mathrm{I}} &= \mathbf{v}(t) + \xi \mathbf{f}[\mathbf{r}(t)]\Delta t/m\,,\\
\mathbf{r}_{\mathrm{I}} &= \mathbf{r}(t) + \mathbf{v}_{\mathrm{I}}\Delta t/2\,,\\
\mathbf{v}_{\mathrm{II}} &= \mathbf{v}_{\mathrm{I}} + (1-2\xi)\mathbf{f}[\mathbf{r}_{\mathrm{I}}]\Delta t/m\,,\\
\mathbf{r}(t+\Delta t) &= \mathbf{r}_{\mathrm{I}} + \mathbf{v}_{\mathrm{II}}\Delta t/2\,,\\
\mathbf{v}(t+\Delta t) &= \mathbf{v}_{\mathrm{II}} + \xi \mathbf{f}[\mathbf{r}(t+\Delta t)]\Delta t/m\,,
\end{aligned} \tag{22}$$

where the parameter ξ should take its optimal value ξ_0 (see equation (19)), and $\mathbf{r}_{\mathrm{I}}$, $\mathbf{r}_{\mathrm{II}}$, $\mathbf{v}_{\mathrm{I}}$, and $\mathbf{v}_{\mathrm{II}}$ are the auxiliary quantities denoting positions and velocities of all particles in intermediate stages.

The algorithms are simple, require only slight modification with respect to the original Verlet versions, and can be easily implemented in the existing program codes.

3.2. Forest-Ruth-like algorithms of the fourth order

The fourth-order ($K = 4$) algorithm by Forest and Ruth (FR) [5] is reproduced from equation (6) at $P = 4$ as

$$\mathrm{e}^{(\mathcal{A}+\mathcal{B})\Delta t + \mathcal{O}(\Delta t^5)} = \mathrm{e}^{\mathcal{A}\theta\frac{\Delta t}{2}}\mathrm{e}^{\mathcal{B}\theta\Delta t}\mathrm{e}^{\mathcal{A}(1-\theta)\frac{\Delta t}{2}}\mathrm{e}^{\mathcal{B}(1-2\theta)\Delta t}\mathrm{e}^{\mathcal{A}(1-\theta)\frac{\Delta t}{2}}\mathrm{e}^{\mathcal{B}\theta\Delta t}\mathrm{e}^{\mathcal{A}\theta\frac{\Delta t}{2}} \tag{23}$$

with $a_1 = a_4 = \theta/2$, $a_2 = a_3 = (1-\theta)/2$, $b_1 = b_3 = \theta$, $b_2 = (1-2\theta)$, $b_4 = 0$, $c_1 = c_2 = c_3 = c_4 = 0$, and $\theta = 1/(2-\sqrt[3]{2}) \approx 1.3512$.

Let us consider now an extended decomposition scheme of the fourth order by allowing P to exceed the necessary minimum ($P = 4$) by unity, i.e., letting $P = 5$. Remember that we cannot choose the number P to be too big, because this results in a too large number, namely $P-1$, of expensive force recalculations. Choosing $P = 5$ we just hope to reduce the truncation errors significantly with a little additional computational cost, rather than to increase the order of the decomposition scheme (note that sixth-order integrators are derivable [4] beginning from $P = 8$).

For $P = 5$, the extended decomposition can be presented in the form

$$\begin{aligned} &\mathrm{e}^{(\mathcal{A}+\mathcal{B})\Delta t+\mathcal{O}_3\Delta t^3+\mathcal{O}_5\Delta t^5+\mathcal{O}(\Delta t^7)} = \\ &= \mathrm{e}^{\mathcal{B}\xi\Delta t}\mathrm{e}^{\mathcal{A}(1-2\lambda)\frac{\Delta t}{2}}\mathrm{e}^{\mathcal{B}\chi\Delta t}\mathrm{e}^{\mathcal{A}\lambda\Delta t}\mathrm{e}^{\mathcal{B}(1-2(\chi+\xi))\Delta t}\mathrm{e}^{\mathcal{A}\lambda\Delta t}\mathrm{e}^{\mathcal{B}\chi\Delta t}\mathrm{e}^{\mathcal{A}(1-2\lambda)\frac{\Delta t}{2}}\mathrm{e}^{\mathcal{B}\xi\Delta t} \end{aligned} \tag{24}$$

following from equation (6) with $a_1 = 0$, $b_1 = b_5 = \xi$, $a_2 = a_5 = (1-2\lambda)/2$, $b_2 = b_4 = \chi$, $a_3 = a_4 = \lambda$, $c_{1-5} = 0$, and $b_3 = 1-2(\chi+\xi)$. Here the symmetry of time coefficients and the condition $\sum_{p=1}^{P} a_p = \sum_{p=1}^{P} b_p = 1$ have already been taken into account. The fourth-order conditions now read

$$\begin{aligned} \alpha(\xi,\lambda,\chi) &= -\frac{1}{24} + \lambda^2\chi + \frac{\xi}{4} = 0\,, \\ \beta(\xi,\lambda,\chi) &= -\frac{1}{12} + \lambda\chi(1-\chi-2\xi) + \frac{\xi}{2} - \frac{\xi^2}{2} = 0\,. \end{aligned} \tag{25}$$

Equation (25) provides the cancellation ($\mathcal{O}_3 = 0$) of third-order truncation uncertainties (see equation (11)) and thus describes a whole family of symplectic time-reversible integrators of the fourth order.

A particular member of the above family can be obtained by choosing the corresponding values for ξ, λ, and χ. As far as there are three parameters and only two constraints (see equation (25)), one of these parameters, ξ say, can be treated as a free one. Then, for example, putting $\xi = 0$, equation (24) reduces to the original FR algorithm (23). The extended (when $\xi \neq 0$) propagation will require already four, instead of three, force recalculation per time step. However, having a room in varying ξ, we can overcompensate the increased computational efforts by minimizing the fifth-order truncation uncertainties $\mathcal{O}_5\Delta t^5$.

In order to show this, let us consider explicit expressions for the γ-multipliers. They are:

$$\begin{aligned} \gamma_1 &= \frac{7}{5760} - \frac{\lambda^2\chi}{12}\Big(\frac{1}{2}-\lambda^2\Big) - \frac{\xi}{192}\,, \\ \gamma_2 &= \frac{1}{480} - \frac{\lambda\chi}{2}\Big(\frac{1}{12} - \frac{\lambda}{6} + \lambda^2\chi + \lambda\xi - \frac{\chi}{12} - \frac{\xi}{6}\Big) - \frac{\xi^2}{24}\,, \\ \gamma_3 &= \frac{1}{360} - \lambda^2\chi\Big(\frac{1}{6} - \frac{\lambda}{6} - \frac{\lambda\chi}{3} + \frac{\lambda\xi}{3} - \frac{\xi}{2}\Big) - \frac{\xi}{48} + \frac{\xi^2}{24}\,, \\ \gamma_4 &= \frac{1}{120} - \lambda\chi\Big(\frac{1}{6} - \frac{\lambda}{2}\Big[\frac{1}{6} + \frac{\chi}{2} - \chi^2 - \chi\xi - \xi + \xi^2\Big] \\ &\quad -\frac{\chi}{6} + \frac{\chi\xi}{2} - \frac{5\xi}{6} + \xi^2\Big) - \frac{\xi}{16} + \frac{7\xi^2}{48} - \frac{\xi^3}{8}\,. \end{aligned}$$

Then the norm of $\mathcal{O}_5$ (see equation (12)) is

$$\gamma(\xi, \lambda, \chi) = \sqrt{\gamma_1^2 + \gamma_2^2 + \gamma_3^2 + \gamma_4^2}\,, \tag{26}$$

and the global errors will behave as

$$\Gamma_4(\xi, \lambda, \chi, \Delta t) = \gamma(\xi, \lambda, \chi)\Delta t^4\,. \tag{27}$$

The optimization of the extended propagation (24) with respect to time coefficients ξ, λ, and χ can be carried out by finding the global minimum for the function $\gamma(\xi, \lambda, \chi)$, provided $\alpha = 0$ and $\beta = 0$, i.e., solving the system of equations

$$\begin{cases} \alpha(\xi, \lambda, \chi) = 0\,, \\ \beta(\xi, \lambda, \chi) = 0\,, \\ \gamma(\xi, \lambda, \chi) = \min\,(\text{global})\,. \end{cases} \tag{28}$$

The result within sixteen significant digits is

$$\begin{aligned} \xi &= +0.1644986515575760\text{E}+00 \\ \lambda &= -0.2094333910398989\text{E}-01 \\ \chi &= +0.1235692651138917\text{E}+01 \end{aligned} \tag{29}$$

with the global minimum $\gamma_{\min}^{\text{EFRL}} \approx 0.00065$, where the abbreviation EFRL refers to the extended Forest-Ruth-like scheme (24). At the same time, the value of γ corresponding to the usual FR integration (23) is equal only to $\gamma^{\text{FR}} \approx 0.028$. We see, therefore, that applying the extended integration allows one to decrease the truncation errors approximately $\gamma^{\text{FR}}/\gamma_{\min}^{\text{EFRL}} \approx 43$ times. Taking into account that such a decrease has been achieved by increasing the number of force evaluations per step from three to four, the extended propagation must be performed with step sizes which are a factor of 4/3 higher than those of the FR algorithm, in order to provide the same number of total force recalculations during the fixed overall interval of integration. Thus, we will come to more efficient calculations if the inequality ${\Gamma_4}_{\min}^{\text{EFRL}}(4\Delta t/3) < {\Gamma_4}^{\text{FR}}(\Delta t)$ is fulfilled. Simple computations show

$$\frac{{\Gamma_4}_{\min}^{\text{EFRL}}(4\Delta t/3)}{{\Gamma_4}^{\text{FR}}(\Delta t)} \approx 0.073\,, \tag{30}$$

so that *the global errors can be reduced approximately 15 times with respect to the FR integration without spending any additional computational costs.*

3.3. Classification of schemes through 11 stages

The decomposition schemes of higher orders and with larger stage numbers can be derived and optimized in a similar way as the above. Some of them have been reported in our recent papers [33,34,37]. Herein, the optimization within the Suzuki composition approach [14] was considered as well [33]. In this paper we present

Table 1. The complete family of self-adjoint decomposition algorithms with up to eleven stages

Algorithm	Order	n_{f}	n_{g}	Err_3	Err_5	Err_7	Efficiency	Remarks	No.
BAB	2	1	0	0.0932	.00913	.00132	10.7**	[21,22]	1
ABA	2	1	0	0.0932	.00911	.00134	10.7**	[22]	2
CAC	2	1	1	0.0833	.0134	.00224	1.3	New	3
ACA	2	1	1	0.0417	.00648	.000725	2.7*	New	4
BABAB[(a)]	2	2	0	**0.00855**	.00103	.000110	**29.2*****	**New**	5
ABABA	2	2	0	**0.00855**	.00106	.000106	**29.2*****	**New**	6
CABAC	4	2	1	0	.00334	.000272	1.2*	New	7
BACAB	4	2	1	0	.000713	.0000630	5.5**	[15,18]	8
CACAC	4	2	2	0	.000595	.0000483	1.3*	New	9
ACACA	4	2	2	0	.000715	.0000559	1.1*	[15,18]	10
BABABAB	4	3	0	0	.0383	.0126	.32	[5]	11
ABABABA	4	3	0	0	.0283	.00630	.44	[5]	12
CABABAC	4	3	1	0	.000855[(b)]	.0000224[(b)]	1.9*[(b)]	[30]/New	13
ABACABA	4	3	1	0	.000141[(b)]	.0000104[(b)]	11.3**[(b)]	[18]/New	14
BACACAB	4	3	2	0	.0000443	.00000371	9.4**	New	15
ACABACA	4	3	2	0	.0000823	.00000912	5.1**	New	16
CACACAC	4	3	3	0	.0000167	.00000574	9.1**	New	17
ACACACA	4	3	3	0	.0000123	.00000491	12.4**	New	18
BABABABAB	4	4	0	0	.000654	.0000645	6.0**	New	19
ABABABABA	4	4	0	0	.000610	.0000456	6.4**	New	20
BABACABAB	4	4	1	0	.0000634	.00000511	12.2**	New	21
CABABABAC	4	4	1	0	.000294	.0000129	2.6*	New	22
CABACABAC	4	4	2	0	**.00000368**	.00000679	**66.3*****	**New**	23
BACABACAB	4	4	2	0	.00000649	.00000123	37.6***	New	24
ABACACABA	4	4	2	0	.0000323	.00000171	7.6**	New	25
ACABABACA	4	4	2	0	.0000464	.00000541	5.3**	New	26
CACABACAC	4	4	3	0	.00000605	.0000104	16.5**	New	27
CACACACAC[(c)] ≡ BACACACAB	6	4	3	0	0	.00150	.00067	New	28
ACACACACA	4	4	4	0	.00000312	.00000227	15.5**	New	29
BABABABABAB	4	5	0	0	**.0000270**	.00000272	**59.3*****	**New**	30
ABABABABABA	4	5	0	0	.0000518	.0000173	30.9***	New	31
CABABABABAC	4	5	1	0	.0000165	.00000631	25.2***	New	32
ABABACABABA	4	5	1	0	.0000121	.00000537	34.4***	New	33
BACABABACAB	4	5	2	0	.00000320	.00000156	47.6***	New	34
BABACACABAB	4	5	2	0	.0000132	.00000351	11.5**	New	35
ABACABACABA	4	5	2	0	**.00000127**	.00000224	**120.0*****	**New**	36
ACABABABACA	4	5	2	0	.0000117	.00000483	13.0**	New	37
CABACACABAC	6	5	3	0	0	.0000147	.0384**	New	38
CACABABACAC	6	5	3	0	0	.0000264	.0214*	New	39
ACABACABACA	6	5	3	0	0	.00000607	.0930***	New	40
ABACACACABA	6	5	3	0	0	.000146	.0039	New	41
BACACACACAB	6	5	4	0	0	.00000366	.0566**	New	42
ACACABACACA	6	5	4	0	0	.0000139	.0149*	New	43
CACACACACAC	6	5	5	0	0	.00000249	.0353**	New	44
ACACACACACA	6	5	5	0	0	.00000299	.0294*	New	45

[(a)] Particularly outstanding algorithms are shown by the bold font

[(b)] The values corresponding to new algorithms

[(c)] This nine-stages algorithm is, in fact, of the sixth order [34] only the final results with brief comments to the most important cases. All the decomposition algorithms, obtained by us within up to $S = 11$ stages, are collected in table 1.

In our classification, for each algorithm introduced we point out the order K of the integration as well as the numbers n_{f} and n_{g} of force and force-gradient evaluations per time step. The designations Err_3, Err_5, and Err_7 relate to the norms d, γ, and $\zeta = \sum_{i=1}^{10} {\zeta_i}^2$ of third-, fifth- and seventh-order truncation errors, respectively (see equations (11)–(13), (17), and (26)). Note that the number $n_{\mathrm{f}} = P - 1 = (S+1)/2 - 1$ of force recalculations is directly connected with the number $S = 2P - 1$ of stages. The maximal possible value of n_{g} is, thus, equal to n_{f}. The minimal value is equal to zero and can be achieved when all the coefficients $c_p = 0$ are equal to zero as well. The latter case will correspond to a scheme belonging to the non-gradient class. The force-gradient class will be filled up with algorithms having nonzero numbers n_{g}. Different numbers n_{g} will lead to different decomposition variants within the same value of S. In addition, we distinguished two versions, namely velocity and position ones, which are related to the first ($a_1 = 0$) and second ($b_P = c_P = 0$) type of boundary self-adjoint constraints.

The efficiency of the algorithms has been measured in terms of the quantity

$$\mathrm{Eff}^{(K)} = \frac{1}{(n_{\mathrm{f}} + G n_{\mathrm{g}})^K \mathrm{Err}_{K+1}} \tag{31}$$

at $G = 2$, where Err_{K+1} is the leading error coefficient (i.e. Err_3, Err_5, or Err_7, in dependence of the order $K = 2, 4$, or 6). The weight G reflects the fact that one gradient evaluation requires more (in factor G) processor time than that necessary for one force calculation. Each algorithm has been presented in an abbreviated form, so that the letters A and B correspond to the exponential operators $\exp(\mathcal{A} a_p \Delta t)$ and $\exp(\mathcal{B} b_p \Delta t)$, respectively, whereas the letter C denotes $\exp(\mathcal{B} b_p \Delta t + \mathcal{C} c_p \Delta t^3)$. The optimized values for time coefficients $\{a_p, b_p, c_p\}$ can be found in our quite recent paper [37]. They have been obtained using the computer algebra packages, Maple 6 and Mathematica 4 (the two packages have been applied to ensure the correctness of the results and to minimize transcription mistakes). Each group of algorithms corresponding to the same number ($S = 3, 5, 7, 9$, and 11) of stages is separated by horizontal lines. Within the same group, the algorithms are allocated in the order of increasing of computational efforts. At the same values of n_{f} and n_{g}, the velocity versions (where the letters C or B, but not A, appear first) are written first as well. When all these parameters coincide among themselves, the preference is given to more precise schemes (with smaller values for the leading error coefficient, i.e. with higher efficiency).

It can be seen that within 11 stages there are 46 possible decomposition variants which lead to 45 (10 non-gradient plus 35 force-gradient) different algorithms overall (it has been established that the scheme CACACACAC actually reduces to BACACACAB, because the cancellation of fifth-order truncation errors is reached only when $c_1 = 0$). Among them, 4 non- plus 4 force-gradient integrators were known previously (see the references in the remarks to table 1). All other 37 schemes are new. According to the efficiency, the algorithms have been categorized using a three-star classification (the worst with no star and the best with 3 stars). Some of the integrators are particularly outstanding (they are marked by the bold font) because they lead to the most efficient computational scheme within a given order. For instance, the five-stages non-gradient algorithms (equation (15)), labelled by No. 5 and 6 in table 1, are the best among second-order decomposition schemes. The well-known Verlet integrators (equation (14)) No. 1 and 2, which are used in the majority of molecular dynamics simulations, are obviously worse (see also subsection 3.1). An interesting situation is observed in the fourth-order decomposition schemes. As we can see, the previously known seven-stages integrator by Forest and Ruth (equation (23)) No. 12 results in the worst (with no star) integration. The optimized nine-stages EFRL scheme No. 19 (equation (24)), described in subsection 3.2, is much better (two stars) but not the best. The best integrator of the order four is No. 36 and belongs to the gradient class with eleven stages. It allows to reduce the numerical uncertainties up to $120/0.32 \approx 375$ times at the same overall computational costs. The algorithm No. 23 leads to a similar reduction of numerical errors and appears to be the best within a nine-stages group. The scheme No. 30 also exhibits an equivalence in the efficiency and should be considered as the best within the non-gradient class. Finally, for the order six, the optimal integration can be carried out using the eleven-stages algorithm under No. 40. Its efficiency is higher by a factor of $0.0930/0.00067 \approx 140$ with respect to the worst sixth-order algorithm under No. 28. We see, therefore, that the best algorithms at each order considered belong to extended groups (where the number of stages exceeds the necessary minimum). Moreover, *the efficiency of these new algorithms is higher with respect to the known schemes in factors from 3 up to 375 for orders 2 to 6.*

It should be pointed out that formula (31) can be used to compare the algorithms of the same orders only. In view of the structure of this formula it follows that the relative efficiency of two algorithms is determined at a given K as the ratio of precisions which can be achieved during the integration within the same computational cost. The precision is introduced as the inverse to numerical uncertainties, i.e. as $1/\mathrm{Err}$. The leading term of these uncertainties behaves like $\mathrm{Err}^{(K)} = C_K\, \mathrm{Err}_{K+1} \Delta t^K$, where C_K is the parameter which depends on the order K, on the system under consideration and on the measured quantity. For the same value of K, such a parameter is cancelled when determining the relative efficiency of one integrator with respect to another one. When considering algorithms of different orders, K and $K+M$ say, one obtains that the relative efficiency is $\frac{C_{K+M}}{C_K}\, \mathrm{Eff}^{(K)}/\mathrm{Eff}^{(K+M)}$. It depends not only on the norms Err_{K+1} and Err_{K+M+1} of numerical uncertainties but also on the factor C_{K+M}/C_K. The last factor is not universal and cannot be evaluated theoretically. It should be investigated directly in simulations for each concrete system, where the final decision concerning the preference of higher-order algorithms over lower-order ones or vise versa can be given.

With the increase of the number of stages above 11, the number of decomposition variants increases too rapidly, so that the performance of a complete classification at $S > 11$ becomes difficult. Moreover, sufficiently high stage numbers will result in very cumbersome systems of non-linear order conditions. So that we may come to an unresolvable technical problem when trying to handle such conditions. Our investigations [34] have shown that because of the restricted capabilities of modern computers, the maximal number of stages, which can still be handled within the direct decomposition approach, is limited to 23. This corresponds to force-gradient algorithms of the order 8. Combining the decomposition method with an advanced composition technique allows us to increase the overall order up to 16. Note that in the earlier studies such extremely high-order algorithms were derived using the crude iterative procedure [27]. As has been demonstrated [34], our composition technique is better in efficiency by factors 25 to 1000 for orders 8 to 12, respectively. For orders 14 and 16, the advanced composition schemes, at considerably smaller computational costs, permit to reduce the numerical uncertainties up to 100 000 times with respect to those of the standard iterative approach.

4. Application to molecular dynamics simulations

In order to verify our theoretical predictions presented in section 3, we have applied the optimized Verlet- and Forest-Ruth-like algorithms to MD simulations of a Lennard-Jones (LJ) fluid, and compared the results with those of the original approaches. The system under consideration was a collection of $N = 256$ particles interacting through a shifted LJ-like potential, $\varphi(r) = \Phi(r) - \Phi(r_{\rm c})$ for $r < r_{\rm c}$ and $\varphi(r) = 0$ otherwise, where $\Phi(r) = 4u[(\sigma/r)^{12} - (\sigma/r)^6]$ is the genuine LJ potential. The particles were placed in a basic cubic box of volume $V = L^3$, and the modification of $\Phi(r)$ with $r_{\rm c} = L/2 \approx 3.36\sigma$ as well as the periodic boundary conditions have been used to exclude the finite-size effects. The simulations were performed in a microcanonical ensemble at a reduced density of $n^* = \frac{N}{V}\sigma^3 = 0.845$ and at a reduced temperature of $T^* = k_{\rm B}T/u = 1.7$. All the runs were of the length in $l = 10\,000$ time steps and were started from an identical well equilibrated initial configuration $\boldsymbol{\rho}(0)$. The precision of the algorithms was measured in terms of the relative total energy fluctuations $\mathcal{E} = \langle (E - \langle E\rangle)^2\rangle^{1/2}/|\langle E\rangle|$, where $E = \frac{1}{2}\sum_{i=1}^{N} m\mathbf{v}_i{}^2 + \frac{1}{2}\sum_{i\neq j}^{N}\varphi(r_{ij})$ and $\langle\ \rangle$ denotes the microcanonical averaging. Note that if the equations of motion could be solved exactly, the above fluctuations should vanish, because in microcanonical ensembles the total energy is an integral of motion, $E(t) = E(0)$. So that during approximate MD integrations, smaller values of $\mathcal{E}$ will correspond to a better precision in evaluation of phase trajectories.

The total energy fluctuations obtained in the simulations at the end of the runs for four (fixed within each run) dimensionless time steps, $\Delta t^* = \Delta t(u/m\sigma^2)^{1/2} =$ 0.01, 0.005, 0.0025, and 0.001, are shown in figure 1 as depending on free parameter ξ. The subsets (a) and (b) of this figure correspond to the VV- (see equation (21)) and PV- (see equation (22)) like integration, respectively. As can be seen, all the dependencies $\mathcal{E}(\xi, \Delta t)$ have one minimum which locates at the same point $\xi \approx$ 0.19 independently of the size Δt of the time step. This point coincides completely with the minimum at ξ_0 (equation (19)) of function $d(\xi)$ (equation (17)) which is included in figure 1 as well (dashed curves in the subsets). Moreover, the energy fluctuations $\mathcal{E}(\xi, \Delta t)$ appear to be proportional to the norm $\Gamma_2(\xi, \Delta t)$ of global errors (see equation (18)), and the coefficient of this proportionality almost does not depend on ξ and Δt. In addition, at each step size considered the energy fluctuations decrease at the minimum more than ten times with respect to those at $\xi = 0$, that is in agreement with our predicted value $d(0)/d(\xi_0) \approx 11$.

The MD results, obtained for the total energy fluctuations by the optimized VV- and PV-like algorithms (at $\xi = \xi_0$), are presented in figure 2. The calculated values are functions of the length of the simulations and are plotted by curves marked as OVV and OPV, respectively. For the purpose of comparison, the results corresponding to the original VV and PV integrators are also drawn there (curves marked as VV and PV). Note that for the original integrators, the time step within each subset was chosen to be always twice smaller than that of the optimized versions (this condition is necessary to provide the same number of force recalculations during the same observation time), namely, $\Delta t^* = 0.005$, 0.0025, 0.00125, and 0.0005 (see subsets (a), (b), (c), and (d), respectively). Note also that within the original Verlet algorithms, the higher-orders truncation uncertainties become too big at step sizes $\Delta t^* > 0.005$. In particular, then the ratio of the total energy fluctuations to the fluctuations in potential energy (the standard ratio for estimating the precision of the calculations) appears to be more than a few per cent. For this reason, such large step sizes cannot be used in precise MD simulations and, thus, are not considered in the present study.

As we see in figure 2, both the original (VV and PV) and optimized (OVV and OPV) algorithms exhibit very good stability properties. No systematic deviations in the total energy fluctuations can be observed for all the integrators. Instead, in each of the cases considered the amplitude of these deviations tends to its own value which does not increase with the further increase of the length of the simulations. However, this value appears to be significantly larger for the original versions VV and PV. On the other hand, using the optimized OVV and OPV algorithms even with double sizes of the time step allows us to decrease the unphysical energy fluctuations approximately by a factor of three. This is in an excellent accord with our theoretical prediction (20). Note also that the OPV algorithm is slightly better in energy conservation than its OVV version (whereas the VV integrator is better with respect to the PV counterpart). Furthermore, in view of the structure of equations (21) and (22), the OPV algorithm is more convenient when averaging macroscopic quantities. In particular, then the interparticle potentials can be calculated at the end of time steps simultaneously with the interparticle forces within the same loop, thereby increasing the time efficiency of the computations.

A similar pattern has been observed within the fourth-order integration, where the equations of motion were solved using the extended decomposition scheme (24). The total energy fluctuations, obtained in this case at the end of the runs, are plotted in figure 3 as functions of the parameter ξ for three dimensionless time steps, $\Delta t^* = 0.00125$, 0.0025, and 0.005. Note that the two other time coefficients $\lambda(\xi)$ and $\chi(\xi)$, arising in the decomposition (24), are connected with ξ according to the order condition (25), so that $\gamma(\xi, \lambda(\xi), \chi(\xi)) \equiv \gamma(\xi)$ (see equation (26)).

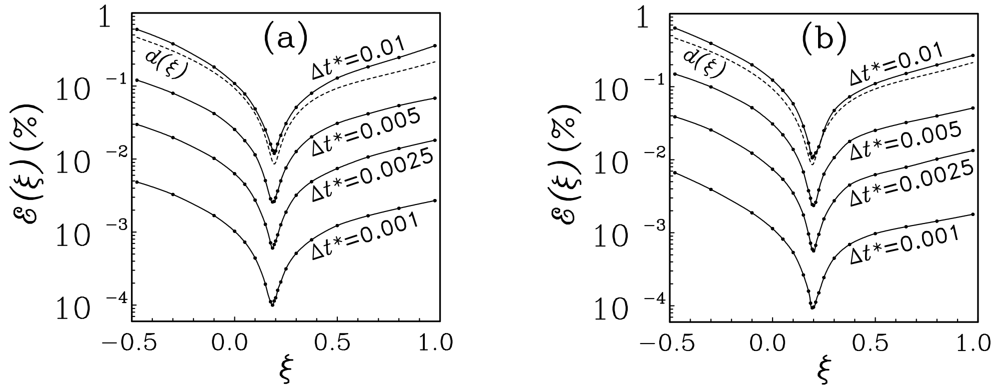

Figure 1. The total energy fluctuations obtained in the MD simulations for different values of free parameter ξ at four reduced time steps, $\Delta t^* = 0.01$, 0.005, 0.0025, and 0.001, using the VV- (subset (a)) and PV- (subset (b)) like integration (equations (21) and (22), respectively). The simulation results are presented by circles connected by the solid curves. The function $d(\xi)$ (see equation (17)) is plotted in both subsets by the dashed curve.

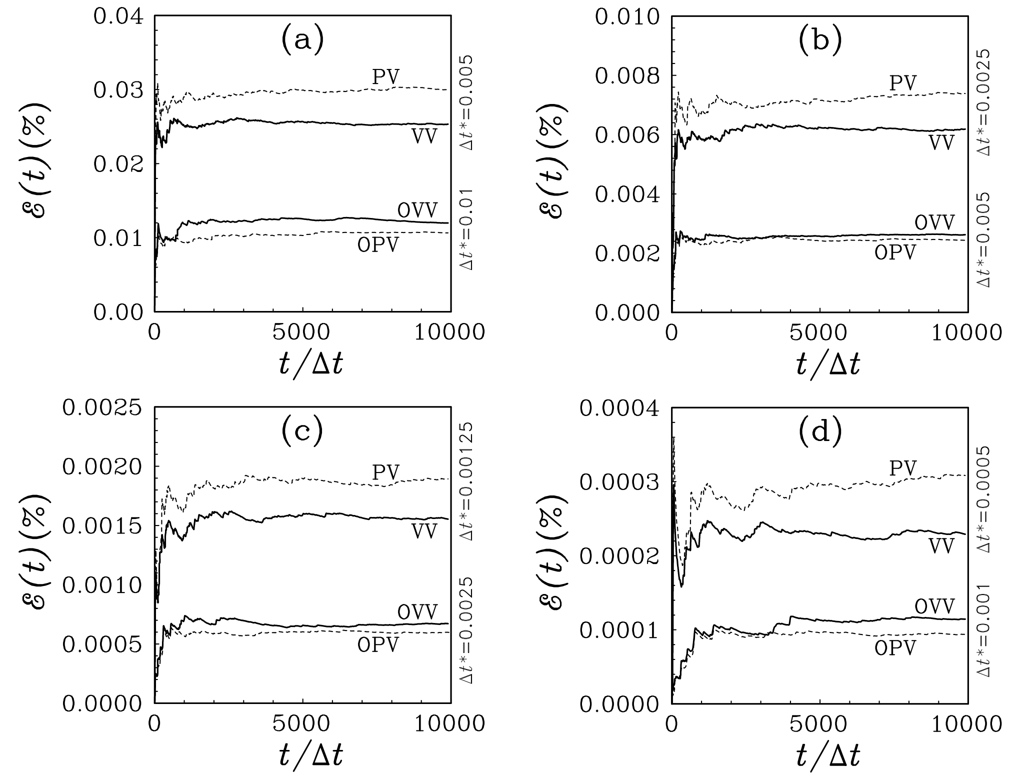

Figure 2. The total energy fluctuations as functions of the length of the MD simulations performed using the optimized VV- (solid curve marked as OVV) and PV- (dashed curve, OPV) like algorithms, as well as the original VV (solid curve, VV) and PV (dashed curve, PV) integrators. The results corresponding to different values of the time step, namely, $\Delta t^* = 0.01$ and 0.005, 0.005 and 0.0025, 0.0025 and 0.00125, as well as 0.001 and 0.0005 are presented in subsets (a), (b), (c), and (d), respectively.

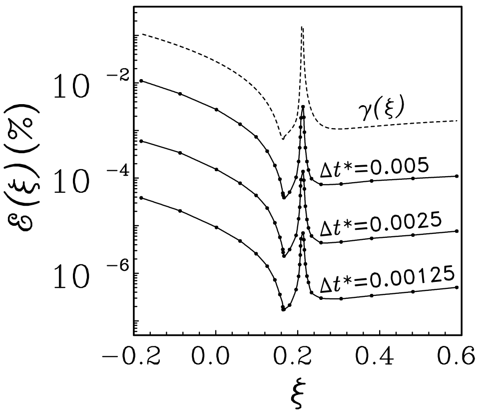

Figure 3. The total energy fluctuations obtained in the MD simulations with the EFRL integration (equation (24)) for different values of free parameter ξ at three fixed time steps, $\Delta t^* = 0.00125$, 0.0025, and 0.005. The simulation results are presented by circles connected by the solid curves. The function $\gamma(\xi)$ (see equation (26)) is plotted by the dashed curve.

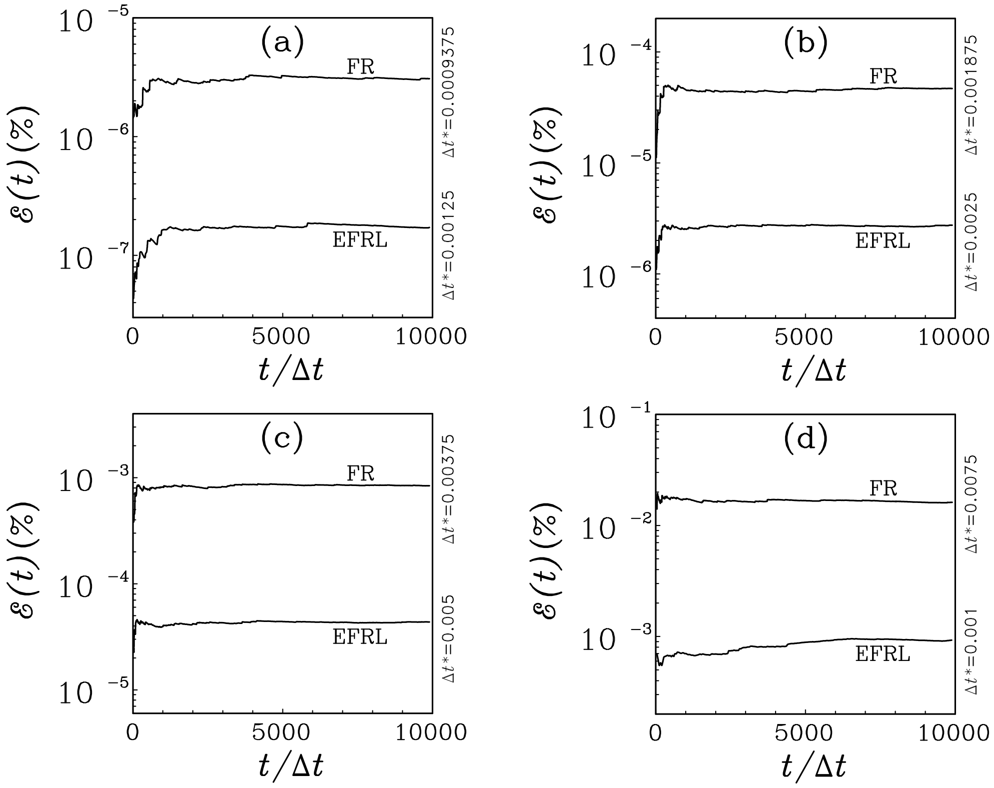

Figure 4. The total energy fluctuations as functions of the length of the MD simulations performed using the optimized EFRL algorithm (equations (24) and (29)) as well as the original FR integrator (equation (23)). The results corresponding to different values of the time step, namely, $\Delta t^* = 0.00125$ and 0.0009375, 0.0025 and 0.001875, 0.005 and 0.00375, as well as 0.01 and 0.0075 are presented in subsets (a), (b), (c), and (d), respectively.

Again, as in the case of second-order integrators (see figure 1), all the functions $\mathcal{E}(\xi, \Delta t)$ display the global minimum at the same point on the ξ-axis, namely, at $\xi \approx 0.164$, independently of Δt. Such a point coincides with the minimum given by equation (29) for the function $\gamma(\xi)$ (which is presented in figure 3 by the dashed curve). This is so because the fluctuations $\mathcal{E}(\xi, \Delta t)$ prove to be proportional to the norm $\Gamma_4 = \gamma \Delta t^4$ (equation (27)) of fourth-order global errors. In addition, these fluctuations decrease at least 40 to 50 times with respect to those at $\xi = 0$, that is close to our predicted value $\gamma^{\mathrm{FR}}/\gamma_{\mathrm{min}}^{\mathrm{EFRL}} \approx 43$.

The EFRL-energy fluctuations obtained in the optimization regime (29) are presented in subsets (a), (b), (c), and (d) of figure 4 as depending on the length of the simulations for the time steps $\Delta t^* = 0.00125$, 0.0025, 0.005, and 0.01, respectively. The corresponding dependencies related to the original FR integrator (see equation (23)) are shown in this figure as well. Within the FR integration, the time step for each subset was by a factor of 4/3 smaller than that of the optimized version, i.e., $\Delta t^* = 0.0009375$, 0.001875, 0.00375, and 0.0075 (then the total number of force recalculations during a given observation time $t \gg \Delta t$ is the same for the both approaches). As can be easily seen, the optimized EFRL algorithm leads to a much lower (15 to 25 times) level of numerical uncertainties than the original version, despite the usage of larger time steps. This is in a good agreement with our estimations obtained in subsection 3.2 (see equation (30)).

Some other fourth-order algorithms presented in table 1, including the force-gradient versions, have been applied to MD simulations as well [34,37]. The estimated and actual efficiencies obtained for such algorithms were also close between themselves as was found for the cases considered above.

5. Conclusion

In the present study, a complete classification of all analytically integrable self-adjoint decomposition algorithms which include up to 11 stages has been given. The algorithms were consequently derived and the optimization processes used were described. As a result, we have found 37 (6 non-gradient plus 31 force-gradient) new schemes in addition to 8 (4 non- and 4 force-gradient) previously known integrators. It has been predicted theoretically and confirmed in our MD simulations that some of the new algorithms are particularly outstanding. They may lead to a much superior integration of the equations of motion in comparison with decomposition schemes widely used in the literature, such as, for example, the well-known Verlet and Forest-Ruth integrators. In particular, it has been shown that the new Verlet-like algorithms proposed should be considered as the best ones with respect to time efficiency among all possible decomposition integrators (with single splitting of the Liouville operator) of the second order. All the algorithms introduced are explicit (i.e., do not require any iteration), simple in implementation, and produce stable solutions which (like exact phase trajectories) are symplectic and time reversible.

The approach presented can also be adapted for the optimization of the integration of motion in more complicated systems, where splitting of the Liouville operator into more than two parts may pay dividends. The examples are multi-component fluids and systems with long-range interactions, when characteristic time scales of the dynamic processes can differ by many orders from each other. In some cases, e.g. systems with orientational degrees of freedom, additional splitting may be necessary to obtain analytically integrable parts. These and related problems will be considered in our further investigations.

Part of this work was supported by the Fonds zur Förderung der wissenschaftlichen Forschung under Project No. P15247. I.M. and I.O. thank the Fundamental Researches State Fund of the Ministry of Education and Science of Ukraine for support under Project No. 02.07/00303.

References

1. Gear C.W. Numerical Initial Value Problems in Ordinary Differential Equations. Engelwood Cliffs, NJ, Prentice-Hall, 1971.
2. Burden R.L., Faires J.D. Numerical Analysis. 5th ed. Boston, PWS Publishing, 1993.
3. Allen M.P., Tildesley D.J. Computer Simulation of Liquids. Oxford, Clarendon, 1987.
4. Yoshida H. // Phys. Lett. A, 1990, vol. 150, p. 262.
5. Forest E., Ruth R.D. // Physica D, 1990, vol. 43, p. 105.
6. Frenkel D., Smit B. Understanding Molecular Simulation: from Algorithms to Applications. New York, Academic Press, 1996.
7. Omelyan I.P., Mryglod I.M., Folk R. // Phys. Rev. Lett., 2001, vol. 86, p. 898.
8. Omelyan I.P. // Phys. Rev. E, 1998, vol. 58, p. 1169.
9. Kinoshita H., Yoshida H., Nakai H. // Celest. Mech., 1991, vol. 50, p. 59.
10. Gladman B., Duncan M., Candy J. // Celest. Mech., 1991, vol. 52, p. 221.
11. Wisdom J., Holman M. // Astrophys. J., 1991, vol. 102, p. 1528.
12. Suzuki M. // Phys. Lett. A, 1992, vol. 165, p. 387.
13. Forest E. // J. Comput. Phys., 1992, vol. 99, p. 209.
14. Suzuki M., Umeno K. – In: Computer Simulation Studies in Condensed Matter Physics VI. Edited by D.P. Landau, K.K. Mon, and H.-B. Schüttler. Berlin, Springer, 1993.
15. Suzuki M. – In: Computer Simulation Studies in Condensed Matter Physics VIII. Edited by D.P. Landau, K.K. Mon, and H.-B. Schüttler. Berlin, Springer-Verlag, 1995.
16. Li R.-C. PhD thesis. Department of Mathematics, University of California at Berkeley, CA, 1995.
17. Suzuki M. // Phys. Lett. A, 1995, vol. 201, p. 425.
18. Chin S.A. // Phys. Lett. A, 1997, vol. 226, p. 344.
19. Dullweber A., Leimkuhler B., McLachlan R. // J. Chem. Phys., 1997, vol. 107, p. 5840.
20. Krech M., Bunker A., Landau D.P. // Comput. Phys. Commun., 1998, vol. 111, p. 1.
21. Swope W.C., Andersen H.C., Berens P.H., Wilson K.R. // J. Chem. Phys., 1982, vol. 76, p. 637.
22. Tuckerman M., Berne B.J., Martyna G.J. // J. Chem. Phys., 1992, vol. 97, p. 1990.
23. Qin M., Zhu W.J. // Computing, 1992, vol. 47, p. 309.
24. McLachlan R.I. // SIAM J. Sci. Comput., 1995, vol. 16, p. 151.
25. Kahan W., Li R.-C. // Mathematics of Computation, 1997, vol. 66, p. 1089.
26. Murua A., Sanz-Serna J.M. // Phil. Trans. Roy. Soc. A, 1999, vol. 357, p. 1079.

27. Chin S.A., Kidwell D.W. // Phys. Rev. E, 2000, vol. 62, p. 8746.
28. Drozdov A.N., Brey J.J. // Phys. Rev. E, 1998, vol. 57, p. 1284.
29. Forbert H.A., Chin S.A. // Phys. Rev. E, 2000, vol. 63, p. 016703.
30. Chin S.A., Chen C.R. // J. Chem. Phys., 2001, vol. 114, p. 7338.
31. Stuart S.J., Zhou R., Berne B.J. // J. Chem. Phys., 1996, vol. 105, p. 1426.
32. Omelyan I.P., Mryglod I.M., Folk R. // Phys. Rev. E, 2002, vol. 65, p. 056706.
33. Omelyan I.P., Mryglod I.M., Folk R. // Comput. Phys. Commun., 2002, vol. 146, p. 188.
34. Omelyan I.P., Mryglod I.M., Folk R. // Phys. Rev. E, 2002, vol. 66, p. 026701.
35. Sofroniou M., Oevel W. // SIAM J. Numer. Anal., 1997, vol. 34, p. 2063.
36. Bornemann F.A., Nettesheim P., Schütte Ch. // J. Chem. Phys., 1996, vol. 105, p. 1074.
37. Omelyan I.P., Mryglod I.M., Folk R. // Comput. Phys. Commun., 2002 (in press).

Нові оптимізовані алгоритми для моделювання методом молекулярної динаміки

І.П.Омелян [1,2], І.М.Мриглод [1,2], Р.Фольк [2]

[1] Інститут фізики конденсованих систем НАН України, 79011 Львів, вул. Свєнціцького, 1

[2] Інститут теоретичної фізики, Університет м. Лінц, Альтенбергштрассе 69, A-4040 Лінц, Австрія

Отримано 29 березня 2002 р.

Метод молекулярної динаміки (МД) є потужним знаряддям для передбачення і вивчення різноманітних явищ у фізиці, хімії та біології. Побудова ефективних МД алгоритмів для інтегрування рівнянь руху в класичних і квантових багаточастинкових системах повинна, отже, істотно вплинути на розвиток багатьох областей фундаментальних досліджень. У даному розгляді показано, що більшість існуючих МД інтеграторів далекі від ідеальних, і може бути досягнуто подальше значне покращення ефективності обчислень. Як результат, ми пропонуємо нові оптимізовані алгоритми, які дозволяють зменшити чисельні похибки до мінімуму при тих самих загальних обчислювальних затратах. Оптимізація здійснюється у рамках загально визнаного декомпозиційного підходу і стосується широко застосовуваних симплектичних схем Верле, Фореста-Рутха, Сузукі, а також схем, які базуються на обчисленні градієнтів сил. Ми приходимо до висновку, що ефективність нових алгоритмів може бути кращою порівняно з оригінальними інтеграторами від 3 до 1000 разів для порядків від 2 до 12. Цей висновок підтверджується у наших МД симуляціях Леннард-Джонсівської рідини для випадку схем інтегрування другого і четвертого порядку точності.

Ключові слова: *молекулярна динаміка, симплектичне інтегрування, декомпозиційний метод, умови порядків, чисельні схеми*

PACS: *02.60.Cb, 02.70.Ns, 05.10.-a, 95.10.Ce, 95.75.Pq*