Extrapolated gradientlike algorithms for molecular dynamics and celestial mechanics simulations

I. P. Omelyan
Institute for Condensed Matter Physics, 1 Svientsitskii Street, UA-79011 Lviv, Ukraine
and Institute for Theoretical Physics, Linz University, A-4040 Linz, Austria
(Received 29 March 2006; revised manuscript received 16 June 2006; published 19 September 2006)

A class of symplectic algorithms is introduced to integrate the equations of motion in many-body systems. The algorithms are derived on the basis of an advanced gradientlike decomposition approach. Its main advantage over the standard gradient scheme is the avoidance of time-consuming evaluations of force gradients by force extrapolation without any loss of precision. As a result, the efficiency of the integration improves significantly. The algorithms obtained are analyzed and optimized using an error-function theory. The best among them are tested in actual molecular dynamics and celestial mechanics simulations for comparison with well-known nongradient and gradient algorithms such as the Störmer-Verlet, Runge-Kutta, Cowell-Numerov, Forest-Ruth, Suzuki-Chin, and others. It is demonstrated that for moderate and high accuracy, the extrapolated algorithms should be considered as the most efficient for the integration of motion in molecular dynamics simulations.

DOI: 10.1103/PhysRevE.74.036703 PACS number(s): 02.70.Ns, 02.60.Cb, 05.10.−a

I. INTRODUCTION

The method of molecular dynamics (MD) is widely used for modeling of various processes in physics, chemistry, and biology. It allows one from first principles to predict macroscopic properties of many-body systems at a microscopic level of description from known interactions between the constituent atoms and molecules. Despite noticeable previous developments, the construction of more efficient MD algorithms remains a current problem in view of the restricted capabilities of even modern supercomputers. Note that the improvement in the efficiency makes it possible to consider larger systems and examine them during longer observation times, reducing the finite-size effects and statistical noise to a minimum.

Many algorithms have been proposed over the last several decades to solve the equations of motion in MD and celestial mechanics (CM) simulations. The traditional explicit Runge-Kutta (RK) as well as implicit Gear, Adams-Bashforth-Moulton, and other predictor-corrector (PC) schemes of different orders have been exploited in early investigations [1–3]. It was soon established that such schemes are very inefficient because they require too small time steps to avoid an instability in the generated solutions [2,4,5]. As is now well realized, the instability follows from the fact that these solutions do not reproduce certain properties of Hamiltonian systems, such as the conservation of volume in phase space and self-adjointness [6–8]. In other words, the RK and PC algorithms are neither symplectic nor time reversible and cannot be appropriate for long-term MD and CM simulations.

In 1990, a novel approach was proposed [9,10] for the integration of motion in many-particle systems. In this approach, the time evolution propagator is decomposed into a set of analytically solvable exponential operators [9–31]. As a result, the decomposition integrators appear to be exactly symplectic and self-adjoint for an arbitrary order in the time step. For this reason they exhibit exceptional stability properties and thus are ideal for long-duration MD and CM computations. For instance, the well-known velocity Verlet (VV) algorithm [13,32], which is employed in the great majority of MD simulations, can be derived by the decomposition method in terms of three exponential operators. This corresponds to the second-order propagation with one force evaluation per step. Extended versions of the VV algorithm with five exponentials and two force evaluations were also considered [28]. The decomposition with three or four force calculations represents fourth-order integrators [10,29]. Schemes with larger numbers of exponentials and higher orders are generally not considered in MD applications since they entail too large a number of costly force recalculations. These more complicated schemes can be applied to CM systems (such as the solar one) where it is important to compute the trajectories as precisely as possible.

In further studies, the decomposition approach was developed by including additional solvable parts associated with force-gradient contributions into the exponential propagators [33–35]. First force-gradient algorithms were introduced by Suzuki [33] and Chin [35] for order 4 in the context of small perturbations and few-body dynamics, respectively. The advantages of gradient schemes over their nongradient counterparts have been demonstrated in CM [35–37], stochastic [38,39], and quantum [40–45] dynamics simulations. Recently, the gradient approach has been adapted to simulate the dynamics of many-body fluid systems and the corresponding improved force-gradient algorithms have been devised [46,47]. It has been proven theoretically and shown in MD simulations that such algorithms may lead to a much more efficient integration with respect to nongradient ones, despite an extra computational effort spent on the calculation of the gradients. A complete classification and derivation of force and force-gradient decomposition algorithms up to order 6 was done in Ref. [47] in 2003.

It is worth emphasizing that the gradient decomposition approach, like the nongradient one, yields symplectic and self-adjoint algorithms which are explicit—i.e., do not involve iterative procedures. All other symplectic methods known are implicit. An example is the implicit Runge-Kutta (IRK) approach [48] in which expensive iterations should be carried out at each time step to reproduce the symplecticity and self-adjointness properties. Another important feature of the force-gradient approach is the possibility to construct

second- and fourth-order algorithms with decomposition coefficients which are all positive. This is contrary to the nongradient propagation, where beyond second order any scheme expressed in terms of only force exponentials results in some negative time coefficients [11]. Note that the negative time propagation is impossible in stochastic dynamics, nonequilibrium mechanics, quantum statistics, etc., because one cannot simulate stochastic or diffusion processes backward in time or sample configurations with negative temperatures.

As can be seen from the aforesaid, nowadays the gradient decomposition approach is the most powerful tool for the construction of high quality algorithms in different areas of MD applications. The only point concerning this approach is the fact that the direct evaluation of force gradients may present a difficulty for systems with complicated interaction potentials (in particular, for long-range Coulomb interactions, where the cumbersome Ewald summation technique is used to properly take into account the tail contribution). Moreover, such an evaluation is the most time-consuming part of the propagation even for relatively simple potentials. The calculation of force gradients may take the computational time which exceeds considerably the time needed for the calculation of forces themselves. Therefore, in order to further improve the efficiency of the decomposition method, it would be very desirable to develop an advanced approach allowing one to evaluate the force gradients in terms of forces exclusively, rather than to calculate the gradients directly. In this paper, studying in detail the structure of decomposition operators, we show that such an advanced gradientlike approach indeed can be realized.

The paper is organized as follows. The standard gradient decomposition method and its advanced reformulation are described in Sec. II. Classification and derivation of advanced gradientlike algorithms are presented Sec. III. Their applications to MD and CM simulations and comparison with previous integrators are made in Sec. IV. Concluding remarks are added in Sec. V.

II. THEORY

A. Equations of motion

Consider a classical N-body system described by the Hamiltonian

$$H=\sum_{i=1}^{N}\frac{m_i\mathbf{v}_i^2}{2}+\frac{1}{2}\sum_{i\neq j}^{N}\varphi(r_{ij})+\sum_{i=1}^{N}u(\mathbf{r}_i,t). \tag{1}$$

Here $\mathbf{r}_i$ and $\mathbf{v}_i=d\mathbf{r}_i/dt$ are correspondingly the position and velocity of particle i carrying mass m_i and interacting with all the rest bodies via the pairwise potential $\varphi(r_{ij})\equiv\varphi(|\mathbf{r}_i-\mathbf{r}_j|)$ in a spatially inhomogeneous time-dependent external field $u(\mathbf{r}_i,t)$. The Newtonian equations of motion for such a system can be cast in the following compact form:

$$\frac{d\boldsymbol{\rho}}{dt}=\{\boldsymbol{\rho}\odot H\}\equiv L(t)\boldsymbol{\rho}(t), \tag{2}$$

where $\boldsymbol{\rho}=(\mathbf{r}_1,\mathbf{v}_1;\ldots;\mathbf{r}_N,\mathbf{v}_N)\equiv(\mathbf{r};\mathbf{v})$ is the set of phase variables, $\{\odot\}$ denotes the Poisson bracket, and

$$L(t)=\sum_{i=1}^{N}\left(\mathbf{v}_i\cdot\frac{\partial}{\partial\mathbf{r}_i}+\mathbf{a}_i(\mathbf{r},t)\cdot\frac{\partial}{\partial\mathbf{v}_i}\right)\equiv\mathbf{v}\cdot\frac{\partial}{\partial\mathbf{r}}+\mathbf{a}(\mathbf{r},t)\cdot\frac{\partial}{\partial\mathbf{v}} \tag{3}$$

designates the Liouville operator. The set $\mathbf{a}\equiv(\mathbf{a}_1,\ldots,\mathbf{a}_N)$ of accelerations $\mathbf{a}_i(\mathbf{r},t)=\mathbf{f}_i(\mathbf{r},t)/m_i$ is determined by the forces

$$\mathbf{f}_i(\mathbf{r},t)=-\sum_{j(j\neq i)}^{N}\frac{\mathbf{r}_{ij}}{r_{ij}}\varphi'(r_{ij})-\frac{\partial u(\mathbf{r}_i,t)}{\partial\mathbf{r}_i} \tag{4}$$

acting on the particles due to the interactions.

If an initial configuration $\boldsymbol{\rho}(0)$ is specified, the unique solution to Eq. (2) can formally be presented for any time t as

$$\boldsymbol{\rho}(t)=\left(T\exp\left[\int_0^{\Delta t}L(s)ds\right]\right)^l\boldsymbol{\rho}(0), \tag{5}$$

where T is the time ordering operator, $\Delta t=t/l$ the size of the time step, and l the total number of the steps. Suzuki proved [49] that the ordered exponential can be expressed in terms of usual exponential propagators,

$$\boldsymbol{\rho}(t)=(\exp[(\mathcal{D}+L)\Delta t])^l\boldsymbol{\rho}(0)\equiv(e^{\mathcal{L}\Delta t})^l\boldsymbol{\rho}(0), \tag{6}$$

by introducing the time derivative operator $\mathcal{D}=\overleftarrow{\partial}/\partial t$ which acts on the left on time-dependent functions [in particular, $F(t)e^{\mathcal{D}\Delta t}=F(t+\Delta t)$ for arbitrary function F of t]. For conservative systems (then external fields are absent, $u\equiv 0$), the Liouville operator L does not depend explicitly on time, and thus $\mathcal{D}\equiv 0$.

In general Eq. (2) presents a very complicated set of highly nonlinear coupled equations of motion. As is well known, they can neither be solved exactly nor reduced to quadratures already at $N>2$ even for conservative systems. The only way of handling these equations is to perform the calculations by numerical methods. This is especially true in the case of MD simulations of many-body systems, where $N\gg 2$.

B. Standard gradient decomposition method

It has been mentioned in the Introduction that an efficient numerical approach follows from the gradient decomposition method. The main idea of this method is to factorize the exponential propagator $e^{\mathcal{L}\Delta t}$ on such subpropagators which allow to be evaluated analytically. This is achieved by splitting the full operator $\mathcal{L}=\mathcal{D}+L=A+B$ into its kinetic A and potential B parts,

$$A=\mathcal{D}+\mathbf{v}\cdot\frac{\partial}{\partial\mathbf{r}},\quad B=\mathbf{a}(\mathbf{r},t)\cdot\frac{\partial}{\partial\mathbf{v}}. \tag{7}$$

Then the total propagator $e^{\mathcal{L}\Delta t}$ can be decomposed as [46,47]

$$e^{(A+B)\Delta t+\mathcal{E}(\Delta t^{K+1})}=\prod_{p=1}^{P}e^{Aa_p\Delta t}e^{Bb_p\Delta t+Cc_p\Delta t^3}, \tag{8}$$

where $C=[B,[A,B]]$ and $[\cdot,\cdot]$ denotes the commutator of two operators. For a given number $P>1$ of stages, the coef-

ficients a_p, b_p, and c_p have to be chosen in such a way to provide the highest possible value for the order $K>1$ of precision of the decomposition. As a result, the integration (6) can be performed approximately using Eq. (8) and neglecting the truncation terms $\mathcal{E}(\Delta t^{K+1})$ in view of the smallness of the size $\Delta t=t/l$ of the time step (note that $l\gg 1$). The accuracy of the decomposition will increase with rising K and decreasing Δt. For $c_p=0$ at $p=1,2,\ldots,P$, Eq. (8) reduces to the usual nongradient factorization [9–11,15,29,33].

Taking into account explicit expressions [Eqs. (7)] for operators A and B, one obtains [47]

$$C=[B,[A,B]]=\sum_{i=1}^{N}\frac{\mathbf{g}_i(\mathbf{r},t)}{m_i}\cdot\frac{\partial}{\partial\mathbf{v}_i}\equiv\mathbf{G}(\mathbf{r},t)\cdot\frac{\partial}{\partial\mathbf{v}},\qquad(9)$$

where

$$\mathbf{g}_{i\alpha}(\mathbf{r},t)=2\sum_{j\beta}\frac{\partial\mathbf{f}_{i\alpha}(\mathbf{r},t)}{\partial\mathbf{r}_{j\beta}}\mathbf{a}_{j\beta}(\mathbf{r},t),\qquad(10)$$

α and β denote the Cartesian components of vectors, $\mathbf{G}\equiv(\mathbf{g}_1/m_1,\ldots,\mathbf{g}_N/m_N)$, and the property $[B,[\mathcal{D},B]]=0$ has been used. In view of Eq. (4), the force-gradient evaluations $\partial\mathbf{f}_{i\alpha}/\partial\mathbf{r}_{j\beta}$ can be carried out explicitly and the result is

$$\mathbf{g}_i(\mathbf{r},t)=-2\sum_{j(j\neq i)}^{N}\left[(\mathbf{w}_i-\mathbf{w}_j)\frac{\varphi'_{ij}}{r_{ij}}+[\mathbf{r}_{ij}\cdot(\mathbf{w}_i-\mathbf{w}_j)]\times\frac{\mathbf{r}_{ij}}{r_{ij}^3}(r_{ij}\varphi''_{ij}-\varphi'_{ij})\right]+\mathbf{h}_i(t),\qquad(11)$$

where $\mathbf{w}_i(\mathbf{r},t)=-\frac{1}{m_i}\Sigma_{j(j\neq i)}\mathbf{r}_{ij}\varphi'(r_{ij})/r_{ij}$ is the interparticle part of the full acceleration $\mathbf{a}_i(\mathbf{r},t)=\mathbf{w}_i(\mathbf{r},t)-\frac{1}{m_i}\partial u(\mathbf{r}_i,t)/\partial\mathbf{r}_i$, $\varphi'_{ij}\equiv\varphi'(r_{ij})=d\varphi(r_{ij})/dr_{ij}$, $\varphi''_{ij}=d\varphi'_{ij}/dr_{ij}$, and $\mathbf{h}_{i\alpha}(t)=\frac{2}{m_i}\Sigma_\beta\partial u(\mathbf{r}_i,t)/\partial\mathbf{r}_{i\beta}\,\partial^2u(\mathbf{r}_i,t)/\partial\mathbf{r}_{i\alpha}\partial\mathbf{r}_{i\beta}\equiv\mathbf{h}_\alpha(\mathbf{r}_i,t)$ denotes the external-field contribution.

As can be seen, the function $\mathbf{G}(\mathbf{r},t)$ like $\mathbf{a}(\mathbf{r},t)$ does not depend on velocity and, thus, the operator C commutes with B. Then, taking into account the independence of $\mathbf{v}$ on $\mathbf{r}$ and the commutation of $\mathcal{D}$ with $\mathbf{v}\cdot\partial/\partial\mathbf{r}$ yields

$$e^{Aa_p\Delta t}(\mathbf{r},\mathbf{v})=e^{\mathcal{D}a_p\Delta t}(\mathbf{r}+a_p\mathbf{v}\Delta t,\mathbf{v}),$$

$$e^{Bb_p\Delta t+Cc_p\Delta t^3}(\mathbf{r},\mathbf{v})=(\mathbf{r},\mathbf{v}+b_p\mathbf{a}\Delta t+c_p\mathbf{G}\Delta t^3),\qquad(12)$$

which represent simple shifts in position and velocity spaces, respectively [46,47]. Note also that the quantities $\mathbf{a}$ and $\mathbf{G}$ being functions of $\mathbf{r}$ may depend also explicitly on time t (if $u\neq 0$). Then any appearance of them on the left of the operator $e^{\mathcal{D}a_p\Delta t}$ will shift their time coordinates—i.e., $(\mathbf{a}(t),\mathbf{G}(\mathbf{r},t))e^{\mathcal{D}a_p\Delta t}=(\mathbf{a}(\mathbf{r},t+a_p\Delta t),\mathbf{G}(\mathbf{r},t+a_p\Delta t))$. Therefore, the exponential subpropagators arising on the right-hand side of Eq. (8) are indeed analytically integrable. This is a very important feature, reducing significantly the computational costs, because no iterations are required to produce the solutions.

Note that the existence of force-gradients terms ($c_p\neq 0$) lowers considerably the truncation errors $\mathcal{E}(\Delta t^{K+1})$ of the decomposition propagation [Eq. (8)] at each given order K of precision [47]. This compensates the increased computational effort spent on the calculation of force gradients and thus may lead to a more efficient propagation than in the case of the nongradient one.

C. Advanced gradientlike reformulation

For many-body systems ($N\gg 1$), the evaluation of force gradients [Eq. (11)] presents the most time-consuming part of the decomposition propagation [Eq. (8)]. Estimations show that, in dependence on the form of the interacting potential φ, this evaluation can take the computational time which is larger in a factor of $\theta=2-5$ with respect to that needed for the calculation of forces [notice that Eq. (4) is much simpler than Eq. (11)]. The factor θ can increase further in the case of long-range (Coulomb) interactions, where the calculation of force gradients is especially difficult because of a very complicated structure of Ewald expressions. For this reason, the question arises on how to avoid the direct evaluation of force gradients and, at the same time, maintain a high level of precision inherent in the gradient method. We will now show that the gradients can be expressed in terms of only force evaluations without breaking the exponential structure of decomposition propagators. This will further improve the efficiency of the decomposition method, because the same accuracy can be reached by lower computational costs.

First, it should be taken into account that the force acting on the ith particle ($i=1,\ldots,N$) depends on the positions of all the bodies [Eq. (4)]—i.e., $\mathbf{f}_i(\mathbf{r},t)\equiv\mathbf{f}_i(\mathbf{r}_1,\ldots,\mathbf{r}_N,t)\equiv\mathbf{f}_i(\{\mathbf{r}_j\},t)$. Second, one introduces modified forces $\mathbf{f}_i(\{\mathbf{r}_j+\xi\mathbf{a}_j(\mathbf{r},t)\Delta t^2\},t)\equiv\mathbf{f}_i(\mathbf{r}+\xi\mathbf{a}\Delta t^2,t)$, obtained by shifting the original positions $\mathbf{r}_j\equiv\mathbf{r}_j(t)$ on small values $\xi\mathbf{a}_j(\mathbf{r},t)\Delta t^2$, where ξ denotes a constant. Then expanding the modified forces into Taylor's series in the shifted positions one finds

$$\mathbf{f}_i(\mathbf{r}+\xi\mathbf{a}\Delta t^2,t)=\mathbf{f}_i(\mathbf{r},t)+\sum_{j=1}^{N}\frac{\partial\mathbf{f}_i(\mathbf{r},t)}{\partial\mathbf{r}_j}\mathbf{a}_j(\mathbf{r},t)\xi\Delta t^2+\mathbf{\Gamma}_i(\mathbf{r},t)=\mathbf{f}_i(\mathbf{r},t)+\mathbf{g}_i(\mathbf{r},t)\xi\frac{\Delta t^2}{2}+\mathbf{\Gamma}_i(\mathbf{r},t),\qquad(13)$$

where the definition [Eq. (10)] of $\mathbf{g}_i(\mathbf{r},t)$ has been used. The higher-order terms can be cast as

$$\mathbf{\Gamma}_i(\mathbf{r},t)=\sum_{k=2}^{\infty}\sum_{j_1,\ldots,j_k=1}^{N}\frac{\partial^k\mathbf{f}_i(\mathbf{r},t)}{\prod_{l=1}^{k}\partial\mathbf{r}_{j_l}}\prod_{l=1}^{k}\mathbf{a}_{j_l}(\mathbf{r},t)\frac{\xi^k\Delta t^{2k}}{k!}\equiv\sum_{k=2}^{\infty}\mathbf{\Gamma}_i^{(k)}(\mathbf{r},t)\xi^k\Delta t^{2k}.\qquad(14)$$

We see therefore that the force-gradient vector $\mathbf{g}_i$ actually appears as the first-order term ($k=1$) in $\xi^k\Delta t^{2k}/k!$ of expansion (13). To simplify the notation, such an expansion can be rewritten in the abbreviated acceleration form

$$\mathbf{a}(\mathbf{r}+\xi\mathbf{a}(\mathbf{r},t)\Delta t^2,t)=\mathbf{a}(\mathbf{r},t)+\mathbf{G}(\mathbf{r},t)\xi\frac{\Delta t^2}{2}+\sum_{k=2}^{\infty}\mathbf{\Gamma}^{(k)}(\mathbf{r},t)\xi^k\Delta t^{2k}, \quad (15)$$

where $\mathbf{G}(\mathbf{r},t)=2\mathbf{a}(\mathbf{r},t)\partial\mathbf{a}(\mathbf{r},t)/\partial\mathbf{r}$ [Eqs. (9) and (10)] and $\mathbf{\Gamma}^{(k)}(\mathbf{r},t)=\frac{1}{k!}\mathbf{a}^k(\mathbf{r},t)\partial^k\mathbf{a}(\mathbf{r},t)/\partial\mathbf{r}^k$ designates the set $(\mathbf{\Gamma}_1^{(k)}/m_1,\ldots,\mathbf{\Gamma}_N^{(k)}/m_N)$. Then the second line of Eq. (12) can be expressed in terms of the modified acceleration [Eq. (15)] as

$$e^{Bb_p\Delta t+Cc_p\Delta t^3}\mathbf{v}=\mathbf{v}+b_p\Delta t\left(\mathbf{a}(\mathbf{r},t)+\frac{c_p}{b_p}\mathbf{G}(\mathbf{r},t)\Delta t^2\right)=\mathbf{v}+b_p\Delta t\left(\mathbf{a}(\mathbf{r}+\xi\mathbf{a}(\mathbf{r},t)\Delta t^2,t)-\sum_{k=2}^{\infty}\mathbf{\Gamma}^{(k)}\xi^k\Delta t^{2k}\right), \quad (16)$$

with $\xi=2c_p/b_p$. Since $\mathbf{\Gamma}^{(k)}(\mathbf{r},t)$ like $\mathbf{a}(\mathbf{r},t)$ and $\mathbf{G}(\mathbf{r},t)$ does not depend on velocity, we can introduce the high-order gradient operators

$$D_k=\mathbf{\Gamma}^{(k)}(\mathbf{r},t)\cdot\frac{\partial}{\partial\mathbf{v}}\equiv\frac{\mathbf{a}^k(\mathbf{r},t)}{k!}\frac{\partial^k\mathbf{a}(\mathbf{r},t)}{\partial\mathbf{r}^k}\cdot\frac{\partial}{\partial\mathbf{v}} \quad (17)$$

(which commute with B and C; note that $C=2D_1$) and transfer the high-order terms of the right-hand side of the third line of Eq. (16) into the left-hand side of the first line under the exponential propagator. Then within the Kth order of precision one obtains

$$\exp\left[Bb_p\Delta t+Cc_p\Delta t^3+\sum_{k=2}^{K/2-1}D_kd_{pk}\Delta t^{2k+1}\right]\mathbf{v}=\mathbf{v}+b_p\Delta t\mathbf{a}\left(\mathbf{r}+\frac{2c_p}{b_p}\mathbf{a}(\mathbf{r},t)\Delta t^2,t\right)+O(\Delta t^{K+1}), \quad (18)$$

where $d_{pk}=2^kc_p^k/b_p^{k-1}$ and the higher-order terms $O(\Delta t^{K+1})$ have been neglected.

In view of Eq. (18), the time evolution propagator can now be decomposed as

$$e^{\mathcal{L}\Delta t+E(\Delta t^{K+1})}=e^{(A+B)\Delta t+E(\Delta t^{K+1})}=\prod_{p=1}^{P}e^{Aa_p\Delta t}\exp\left(Bb_p\Delta t+Cc_p\Delta t^3+\sum_{k=2}^{K/2-1}D_kd_{pk}\Delta t^{2k+1}\right), \quad (19)$$

where $E(\Delta t^{K+1})$ denotes the total truncation uncertainties [which include $O(\Delta t^{K+1})$]. Then the integration (6) can be carried out within an arbitrary order K of precision [so that the truncation terms $E(\Delta t^{K+1})$ are neglected] using the advanced decomposition [Eq. (19)]. The latter has a similar structure to that of the standard force-gradient scheme [Eq. (8)] but includes additional high-order gradient operators D_k. Such an inclusion presents no difficulties; rather, this simplifies the calculations significantly because no evaluation of the gradients is in fact required. According to Eq. (18), all the gradient operators up to a given order K of precision can be collected together to represent a modified (extrapolated) acceleration. Then at each current stage p of the integration process at $c_p\neq 0$, the advanced propagation reduces to the calculation [Eq. (4)] of only two forces corresponding to the genuine $\mathbf{a}(\mathbf{r},t)$ and extrapolated $\mathbf{a}(\mathbf{r}+2c_p\mathbf{a}(\mathbf{r},t)\Delta t^2/b_p,t)$ accelerations. This is contrary to the standard decomposition scheme [Eq. (8)], where besides $\mathbf{a}(\mathbf{r},t)$, the cumbersome force-gradient evaluation of $\mathbf{g}_i(\mathbf{r},t)$ [Eq. (11)] is needed. In the advanced decomposition scheme, such an evaluation is obviated by calculating the extrapolated term $\mathbf{a}(\mathbf{r}+2c_p\mathbf{a}(\mathbf{r},t)\Delta t^2/b_p,t)$ which includes implicitly the force gradients.

Another important feature of the decomposition integration (19) is that it conserves exactly the symplectic map of particle's flow in phase space. This follows from the fact that separate shifts of position [Eq. (12), first line] and velocity [Eq. (18)] do not change the phase volume. The time-reversibility (or self-adjointness) property $S^{-1}(t)=S(-t)$ of the evolution operator $S(t)=e^{Lt}$ for conservative systems ($\mathcal{D}=0$) can also be satisfied exactly by imposing a symmetry on the decomposition coefficients a_p, b_p, and c_p. There are two types of such conditions [47]. The first one reads $a_1=0$, $a_{p+1}=a_{P-p+1}$, $b_p=b_{P-p+1}$, and $c_p=c_{P-p+1}$, so that the velocity will be changed first and the corresponding algorithms will be referred to as of velocity type. The second one is $a_p=a_{P-p+1}$, $b_p=b_{P-p}$, and $c_p=c_{P-p}$ with $b_P=0$ and $c_P=0$, so that the position will be changed first and the corresponding algorithms will be referred to as of position type. According to these conditions, the single-exponential subpropagators will enter symmetrically into the decomposition (19), providing automatically the required self-adjointness. Note that the coefficients $d_{pk}=2^kc_p^k/b_p^{k-1}$ (where $d_{Pk}=0$ when $c_P=0$ and $b_P=0$) will be symmetric as well because they are completely defined by b_p and c_p.

D. Order conditions

For each number $P>1$ of stages, the coefficients a_p, b_p, and c_p of the decomposition [Eq. (19)] must be chosen in such a way to provide the best precision of the integration. This can be done by satisfying order conditions up to the highest possible order $K>1$ of precision. It can be shown that due to the imposed symmetry on decomposition coefficients, all the even-order terms in the error function of Eq. (19) will vanish—i.e.,

$$E(\Delta t^{K+1})=E_1\Delta t+E_3\Delta t^3+E_5\Delta t^5+E_7\Delta t^7+\cdots+E_{K+1}\Delta t^{K+1}+\cdots. \quad (20)$$

Therefore, the order K of the self-adjoint algorithms may accept only even numbers ($K=2,4,6,8,\ldots$). The cancellation of the remaining odd-order terms have to be provided by fulfilling the set of the order conditions $E_1=0$, $E_3=0$, and so on up to $E_{K-1}=0$ for order K.

Let us write down explicit expressions for the functions E_1, E_3, E_5, and E_7 (this will be enough to derive algorithms

up to the eighth order). Expanding both sides of Eq. (19) into a Taylor's series and collecting the terms with the same powers of Δt one finds

$$E_1 = (\nu - 1)A + (\sigma - 1)B,$$

$$E_3 = \alpha[A,[A,B]] + \beta[B,[A,B]], \tag{21}$$

$$E_5 = \gamma_1[A,[A,[A,[A,B]]]] + \gamma_2[A,[A,[B,[A,B]]]] + \gamma_3[B,[A,[A,[A,B]]]] + \gamma_4[B,[B,[A,[A,B]]]] + \eta_1 D_2, \tag{22}$$

$$\begin{aligned} E_7 = {} & \zeta_1[B,[B,[A,[B,[A,[B,A]]]]]] \\ & + \zeta_2[B,[B,[B,[A,[A,[B,A]]]]]] \\ & + \zeta_3[B,[B,[A,[A,[A,[B,A]]]]]] \\ & + \zeta_4[B,[A,[B,[A,[A,[B,A]]]]]] \\ & + \zeta_5[A,[B,[B,[A,[A,[B,A]]]]]] \\ & + \zeta_6[A,[B,[A,[B,[A,[B,A]]]]]] \\ & + \zeta_7[B,[A,[A,[A,[A,[B,A]]]]]] \\ & + \zeta_8[A,[B,[A,[A,[A,[B,A]]]]]] \\ & + \zeta_9[A,[A,[B,[A,[A,[B,A]]]]]] \\ & + \zeta_{10}[A,[A,[A,[A,[A,[B,A]]]]]] + \eta_2 D_3 \\ & + \eta_3[A,[A,D_2]], \end{aligned} \tag{23}$$

where the commutation equalities $[B,C]=[B,D_k]=[C,D_k]=0$ have been taken into account and the Baker-Campbell-Hausdorff (BCH) formula $e^{Ah}e^{Bh}=\exp\left[(A+B)h+\frac{h^2}{2}[A,B]+\frac{h^3}{12}([A,[A,B]]-[B,[A,B]])+\cdots\right]$ has been used. As can be seen from Eqs. (21)–(23), the structure of order conditions for the advanced decomposition scheme is similar to that of the standard gradient approach [46,47]. The only difference is the existence of additional terms with D_k $(k=2,3,\ldots)$ in the error functions E_{2k+1} beginning from the fifth order.

Explicit expressions for the generating functions ν, σ, α, β, γ_{1-4}, η_{1-3}, and ζ_{1-10} can be obtained using a recursive procedure. It is based on the fact that any self-adjoint decomposition scheme [Eq. (19)] can be reproduced by alternately applying $P-1$ times the following two types of symmetric transformations

$$e^{W^{(n+1)}} = e^{Aa^{(n)}\Delta t} e^{W^{(n)}} e^{Aa^{(n)}\Delta t} \tag{24}$$

and

$$\begin{aligned} e^{W^{(n+1)}} = {} & \exp\left(Bb^{(n)}\Delta t + Cc^{(n)}\Delta t^3 + \sum_{k=2}^{K/2-1} D_k d_k^{(n)} \Delta t^{2k+1} \right) \\ & \times e^{W^{(n)}} \exp\left(Bb^{(n)}\Delta t + Cc^{(n)}\Delta t^3 + \sum_{k=2}^{K/2-1} D_k d_k^{(n)} \Delta t^{2k+1} \right), \end{aligned} \tag{25}$$

where $n=0,1,2,\ldots,P-1$ and

$$W^{(n)} = (\nu^{(n)}A + \sigma^{(n)}B)\Delta t + E_3^{(n)}\Delta t^3 + E_5^{(n)}\Delta t^5 + \cdots .$$

The transformations should start from a central single-exponential operator $W^{(0)}$ which always can be selected, because the total number of single-exponential operators is actually equal to an odd value, $S=2P-1$ (note that either $a_1=0$ or $b_P=c_P=d_{Pk}=0$).

For velocity type of decompositions (when $a_1=0$) with even P or for position type (when $b_P=c_P=d_{kP}=0$) with odd P, the central operator is correspondingly $e^{Aa_{P/2+1}\Delta t}$ or $e^{Aa_{(P+1)/2}\Delta t}$. So that here we must put $\sigma^{(0)}=\alpha^{(0)}=\beta^{(0)}=\gamma_{1-4}^{(0)}=\eta_{1-3}^{(0)}=\zeta_{1-10}^{(0)}=0$ as well as either $\nu^{(0)}=a_{P/2+1}$ or $\nu^{(0)}=a_{(P+1)/2}$ on the very beginning $(n=0)$ of the recursive procedure. Moreover, we should start from the transformation (25) at $b^{(0)}=b_Q$, $c^{(0)}=c_Q$, and $d_k^{(0)}=d_{Qk}$, where $Q=P/2$ or $Q=(P+1)/2-1$ with further alternate decreasing the subscript $p=Q,Q-1,\ldots,1$ in $a^{(n)}\equiv a_{p+1}$ or $b^{(n)}\equiv b_p$, $c^{(n)}\equiv c_p$, and $d_k^{(n)}\equiv d_{pk}$ with increasing $n=1,2,\ldots,P-1$ during the transformations (24) or (25). For velocity type with odd P or position type with even P, the central operator is $e^{BbQ\Delta t+Cc_Q\Delta t^3+D_2d_{Q2}\Delta t^5+D_3d_{Q3}\Delta t^7+\cdots}$, where $Q=(P+1)/2$ or $Q=P/2$, which corresponds to $\sigma^{(0)}=b_Q$, $\beta^{(0)}=c_Q$, $\eta_1^{(0)}=d_{Q2}$, and $\eta_2^{(0)}=d_{Q3}$ with $\nu^{(0)}=0$ and $\alpha^{(0)}=\gamma_{1-4}^{(0)}=\zeta_{1-10}^{(0)}=\eta_3^{(0)}=0$. In this case, the procedure should start from the transformation (24) at $a^{(0)}=a_Q$ with alternate decreasing $p=Q,Q-1,\ldots,1$ in $b^{(n)}\equiv b_p$, $c^{(n)}\equiv c_p$, and $d_k^{(n)}\equiv d_{pk}$ or $a^{(n)}\equiv a_p$ at increasing n in Eq. (25) or (24).

Applying again the BCH formula within each elementary transformation, one finds recursive relations. For functions ν, σ, α, β, and η_{1-3} corresponding to transformation (24), they read

$$\begin{aligned} & \nu^{(n+1)} = \nu^{(n)} + 2a^{(n)}, \quad \sigma^{(n+1)} = \sigma^{(n)}, \\ & \alpha^{(n+1)} = \alpha^{(n)} - a^{(n)}\sigma^{(n)}(a^{(n)} + \nu^{(n)})/6, \\ & \beta^{(n+1)} = \beta^{(n)} - a^{(n)}\sigma^{(n)2}/6, \\ & \eta_1^{(n+1)} = \eta_1^{(n)}, \quad \eta_2^{(n+1)} = \eta_2^{(n)}, \\ & \eta_3^{(n+1)} = \eta_3^{(n)} - a^{(n)}\eta_1^{(n)}(a^{(n)} + \nu^{(n)})/6. \end{aligned} \tag{26}$$

For transformation (25) the relations are

$$\begin{aligned} & \nu^{(n+1)} = \nu^{(n)}, \quad \sigma^{(n+1)} = \sigma^{(n)} + 2b^{(n)}, \\ & \alpha^{(n+1)} = \alpha^{(n)} + b^{(n)}\nu^{(n)2}/6, \\ & \beta^{(n+1)} = \beta^{(n)} + [12c^{(n)} + b^{(n)}\nu^{(n)}(b^{(n)} + \sigma^{(n)})]/6, \\ & \eta_1^{(n+1)} = \eta_1^{(n)} + 2d_2^{(n)}, \quad \eta_2^{(n+1)} = \eta_2^{(n)} + 2d_3^{(n)}, \\ & \eta_3^{(n+1)} = \eta_3^{(n)} + d_2^{(n)}\nu^{(n)2}/6. \end{aligned} \tag{27}$$

The relations for γ_{1-4} and ζ_{1-10} are the same as in the case of the standard gradient approach and can be found in Ref. [47].

Therefore, at the end of the recursive procedure we come to the desired generating functions ν, σ, α, β, γ_{1-4}, η_{1-3}, and ζ_{1-10}, expressed explicitly in terms of decomposition coeffi-

cients a_p, b_p, and c_p with $p=1,2,\ldots,P$. Then according to the structure of Eqs. (21)–(23), the order conditions $E_{2k+1}=0$ (where $k=0,1,2,3,\ldots$) transform into a set of nonlinear equations $\nu=\sigma=1$, $\alpha=\beta=0$, $\gamma_{1-4}=0$, $\eta_{1-3}=0$, $\zeta_{1-10}=0$, which should be solved with respect to unknown decomposition coefficients a_p, b_p, and c_p. Some of the equations are particularly simple—namely, $\nu=\Sigma_{p=1}^{P}a_p=1$ and $\sigma=\Sigma_{p=1}^{P}b_p=1$ as well as

$$\eta_1=\sum_{p=1}^{P}d_{2p}=4\sum_{p=1}^{P}\frac{c_p^2}{b_p}=0,\quad \eta_2=\sum_{p=1}^{P}d_{3p}=8\sum_{p=1}^{P}\frac{c_p^3}{b_p^2}=0. \tag{28}$$

Putting $\nu=\sigma=1$ will cancel the first-order truncation uncertainties, $E_1=0$, and lead to second-order ($K=2$) algorithms. Fourth-order ($K=4$) schemes can be obtained by additionally letting $\alpha=0$ and $\beta=0$, destroying the third-order truncation term, $E_3=0$. The next functions γ_{1-4}, η_{1-3}, and ζ_{1-10} should be set to zero to destroy the higher-order error terms E_5 and E_7 for obtaining sixth- and eighth-order algorithms, respectively.

E. Higher-level acceleration scheme

All the commutators appearing at the generating functions in Eqs. (21)–(23) can be reduced to the Liouville-like form

$$\mathcal{F}=\mathbf{F}_\mathbf{r}(\mathbf{r},\mathbf{v},t)\cdot\frac{\partial}{\partial\mathbf{r}}+\mathbf{F}_\mathbf{v}(\mathbf{r},\mathbf{v},t)\cdot\frac{\partial}{\partial\mathbf{v}}, \tag{29}$$

where the vector functions $\mathbf{F}_\mathbf{r}(\mathbf{r},\mathbf{v},t)$ and $\mathbf{F}_\mathbf{v}(\mathbf{r},\mathbf{v},t)$ are different for different commutators. In general, these functions are very complicated (especially for commutators with large numbers of operators A and B) involving various combinations of velocity, acceleration, and force gradient tensors $\mathbf{v}^k$, $\mathbf{a}^k(\mathbf{r},t)$, and $\partial^k\mathbf{a}(\mathbf{r},t)/\partial\mathbf{r}^k$ of different orders k. There is a special group of commutators for which the functions $\mathbf{F}_\mathbf{r}$ are equal to zero, while the functions $\mathbf{F}_\mathbf{v}$ are independent of velocity. For each error function E_{2k+1} (with $k=1,2,3,\ldots$), this group includes such $(2k+1)$th-order commutators C_k in which the operator A appears k times. They are $C_1=[B,[A,B]]\equiv C$ [Eq. (9)],

$$C_2=[B,[B,[A,[A,B]]]]=4D_2+4\mathbf{\Gamma}^{(1)}(\mathbf{r},t)\frac{\partial\mathbf{a}(\mathbf{r},t)}{\partial\mathbf{r}}\cdot\frac{\partial}{\partial\mathbf{v}},$$

$$C_3^{\mathrm{I}}=[B,[B,[A,[B,[A,[B,A]]]]]]=-36D_3-36\mathbf{\Gamma}^{(2)}(\mathbf{r},t)\frac{\partial\mathbf{a}(\mathbf{r},t)}{\partial\mathbf{r}}\cdot\frac{\partial}{\partial\mathbf{v}},$$

$$C_3^{\mathrm{II}}=[B,[B,[B,[A,[A,[B,A]]]]]]=-24D_3-8\left[5\mathbf{\Gamma}^{(2)}\times(\mathbf{r},t)\frac{\partial\mathbf{a}(\mathbf{r},t)}{\partial\mathbf{r}}+\mathbf{\Gamma}^{(1)}(\mathbf{r},t)\left(\frac{\partial\mathbf{a}(\mathbf{r},t)}{\partial\mathbf{r}}\right)^2\right]\cdot\frac{\partial}{\partial\mathbf{v}}, \tag{30}$$

and so on for higher k. The operators D_k [as well as their commutators with A; see Eq. (17)] also belong to the group and commute with C_k and B. Because within the group the function $\mathbf{F}_\mathbf{v}\equiv\mathbf{F}_\mathbf{v}(\mathbf{r},t)$ does not depend on $\mathbf{v}$, the exponential propagators e^{C_kh} with operators C_k [Eq. (30)] allow an analytic representation [like the propagators with B, C, and D_k; see Eq. (18)]. Such a representation can be found as follows.

Denoting the genuine and extrapolated accelerations by $\mathbf{a}^{(0)}(\mathbf{r},t)\equiv\mathbf{a}(\mathbf{r},t)$ and $\mathbf{a}^{(1)}(\mathbf{r},t,\Delta t)\equiv\mathbf{a}(\mathbf{r}+\xi\Delta t^2\mathbf{a}^{(0)},t)$, where $\xi=2c_p/b_p$, we can develop the recursive scheme $\mathbf{a}^{(s+1)}(\mathbf{r},t,\Delta t)=\mathbf{a}(\mathbf{r}+\xi\Delta t^2\mathbf{a}^{(s)},t)$, where $s=0,1,2,\ldots$, to introduce the extrapolation $\mathbf{a}^{(s)}$ of the sth level. The latter presents, in fact, an sth-order approximation to the solution of the self-consistent equation

$$\bar{\mathbf{a}}(\mathbf{r},t,\Delta t)=\mathbf{a}\left(\mathbf{r}+\frac{2c_p}{b_p}\Delta t^2\bar{\mathbf{a}}(\mathbf{r},t,\Delta t),t\right), \tag{31}$$

where the exact solution can be cast as $\bar{\mathbf{a}}(\mathbf{r},t,\Delta t)=\lim_{s\to\infty}\mathbf{a}^{(s)}(\mathbf{r},t,\Delta t)$. It can be shown after cumbersome algebra that this solution satisfies the formula

$$\exp\left[Bb_p\Delta t+Cc_p\Delta t^3+\frac{c_p^2}{b_p}C_2\Delta t^5+\frac{c_p^3}{b_p^2}\times\left(4C_3^{\mathrm{II}}-9C_3^{\mathrm{I}}\right)\frac{\Delta t^7}{9}+\cdots\right]\mathbf{v}=\mathbf{v}+b_p\Delta t\bar{\mathbf{a}}(\mathbf{r},t,\Delta t), \tag{32}$$

where the higher-order commutators of A and B belonging to the above group have been omitted. This formula is exact but requires the infinite number of C_k terms to be included and the infinite number of iterations to be performed to obtain $\bar{\mathbf{a}}(\mathbf{r},t,\Delta t)$. In practical applications, we can use finite counterparts of Eq. (32) up to a given order K of accuracy by neglecting the higher-order truncation uncertainties $O(\Delta t^{K+1})$. Then for the second-level description one finds

$$\exp\left[Bb_p\Delta t+Cc_p\Delta t^3+\frac{c_p^2}{b_p}C_2\Delta t^5+\sum_{k=3}^{K/2-1}\frac{c_p^k}{b_p^{k-1}}\times\Psi_k\Delta t^{2k+1}\right]\mathbf{v}=\mathbf{v}+b_p\Delta t\mathbf{a}^{(2)}(\mathbf{r},t,\Delta t)+O(\Delta t^{K+1}), \tag{33}$$

where Ψ_k are the high-order operators which like D_k [Eqs. (18) and (30)] contain portions of the full commutators C_k.

We can now build the second-level decomposition scheme replacing in Eq. (19) the velocity propagator of Eq. (18) by that of Eq. (33). The difference with respect to the first-level acceleration description [Eq. (18)] is that such a scheme involves under the exponential propagator the full second-order commutator C_2 instead of its shortened counterpart D_2 [Eqs. (17) and (30)], as well as more complete (than D_k) higher-order parts Ψ_k of C_k, where $k=3,\ldots,K/2-1$. For this reason, the second-level acceleration description may lead to a more precise propagation than the first-level one. However, its disadvantage is that it requires three force evaluations $(\mathbf{a}^{(0)},\mathbf{a}^{(1)},\mathbf{a}^{(2)})$ per single-exponential propagation [Eq. (33)] instead of two $(\mathbf{a}^{(0)},\mathbf{a}^{(1)})$ during the advanced propagation [Eq. (18)]. The third- and higher-level descriptions ($s\geq 3$)

will require too large numbers (namely, $s+1$) of force evaluations to map the gradient and thus can be excluded from the consideration.

Within the second-level description [Eq. (33)], the error functions E_5 and E_7 [Eqs. (22) and (23)] should be modified replacing D_2 by C_2 and D_3 by Ψ_3. Then the last term on the right-hand side of Eq. (22) can be collected together with the next to last one resulting in $(\gamma_4+\eta_1)C_2 \equiv \bar{\gamma}_4[B,[B,[A,[A,B]]]]$. Because of $[A,[A,C_2]]=-[A,[B,[A,[B,[A,[B,A]]]]]]$, the last term on the right-hand side of Eq. (23) can be combined with the ζ_6 term leading to $-(\zeta_6-\eta_3)[A,[A,C_2]] \equiv \bar{\zeta}_6[A,[B,[A,[B,[A,[B,A]]]]]]$. As a result, the set of generating functions reduces to ν, σ, α, β, γ_{1-3}, $\bar{\gamma}_4=\gamma_4+\eta_1$, η_2, ζ_{1-5}, $\bar{\zeta}_6=\zeta_6-\eta_3$, and ζ_{7-10}.

The advanced gradientlike approach [Eqs. (18), (19), and (30)] like the standard gradient method [Eqs. (8) and (12)] leads to positive decomposition coefficients $a_p>0$ and $b_p>0$ with $p=1,2,\ldots,P$ for orders $K=2$ and $K=4$, since the commutator $C \equiv C_1$ is included into the exponential operator (the importance of the positiveness was highlighted in the Introduction). The inclusion of the higher-order commutator C_2 or D_2 does not allow one, however, to construct positive decompositions of order $K=6$ (and higher). Indeed, according to the condition $\eta_1=0$ [Eq. (28)], the sum of nonzero terms c_p^2/b_p can be equal to zero when some of them are positive and some negative. Since $c_p^2>0$, it follows immediately that at least one coefficient b_p should be negative. Note that the commutators C_k (as well as their shortened counterparts D_k and Ψ_k) are the only ones which being included under the exponential propagator can be represented analytically. In general, no other commutators of A and B allow such a representation, because then the functions $\mathbf{F}_\mathbf{r}$ and $\mathbf{F}_\mathbf{v}$ will depend on both position and velocity.

Quite recently, Chin [50] has shown that in order to obtain all positive coefficients a_p and b_p for order six, the fifth-order commutator $U=[B,[A,[A,[A,B]]]]$ must be included. But it does not belong to the introduced above analytically integrable commutator class. The only way to handle the exponential propagator $e^{U\Delta t^5}$ is to carry out numerical iterations. However, taking into account a very complicated structure of functions $\mathbf{F}_\mathbf{r}(\mathbf{r},\mathbf{v},t)$ and $\mathbf{F}_\mathbf{v}(\mathbf{r},\mathbf{v},t)$ corresponding to U, this requires the direct evaluation of force gradients up to the third order in position (operator A appears 3 times under U). The latter prevents the implementation of positive sixth-order decomposition algorithms for many-body systems in practice, because then the computational efforts increase dramatically. Moreover, the dependence of $\mathbf{F}_\mathbf{r}$ on $\mathbf{r}$ and $\mathbf{F}_\mathbf{v}$ on $\mathbf{v}$ destroys the symplecticity of the integration, while the iterative character of the propagation will violate the time reversibility of the solutions.

F. Runge-Kutta-Nyström-like representation

In view of Eqs. (12) and (18), the advanced exponential propagation [Eqs. (6) and (19)] presents consecutive position and velocity shifts. Summing these shifts one finds explicit values for position and velocity at the end of the qth intermediate stage,

$$\mathbf{r}^{(q)} = \mathbf{r}(t) + \sum_{p=1}^{q} a_p \mathbf{v}^{(p-1)} \Delta t,$$

$$\mathbf{v}^{(q)} = \mathbf{v}(t) + \sum_{p=1}^{q} b_p \mathbf{a}\left(\mathbf{r}^{(p)} + \frac{2c_p}{b_p}\mathbf{a}(\mathbf{r}^{(p)},t_p)\Delta t^2, t_p\right)\Delta t, \quad (34)$$

where $q=1,2,\ldots,P$, with $\mathbf{r}^{(0)}=\mathbf{r}(t)$ and $\mathbf{v}^{(0)}=\mathbf{v}(t)$. The action of operator $\mathcal{D}$ on time-dependent acceleration [see the text just below Eq. (6)] results in the time coefficients $t_p=t+\tau_p\Delta t$ with $\tau_p=1-\Sigma_{q=1}^{p}a_q \equiv 1-\lambda_p$. Substituting the second line of Eq. (34) into the first one and taking into account that $\mathbf{r}(t+\Delta t) \equiv \mathbf{r}^{(P)}$ and $\mathbf{v}(t+\Delta t) \equiv \mathbf{v}^{(P)}$ yields the following equivalent representation of the advanced decomposition propagation:

$$\mathbf{r}(t+\Delta t) = \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \Delta t^2 \sum_{p=1}^{P-1} \tilde{b}_p \mathbf{a}(\mathbf{r}^{(p)},t_p),$$

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t) + \Delta t \sum_{p=1}^{P} b_p \mathbf{a}(\mathbf{r}^{(p)},t_p), \quad (35)$$

where

$$\mathbf{r}^{(p)} = \mathbf{r}(t) + \lambda_p \mathbf{v}(t)\Delta t + \Delta t^2 \sum_{q=1}^{p-1} \tilde{a}_{pq} \mathbf{a}(\mathbf{r}^{(q)} + \xi_q \mathbf{a}(\mathbf{r}^{(q)},t_q)\Delta t^2, t_q), \quad (36)$$

as well as $\xi_q=2c_q/b_q$ and $\tilde{a}_{pq}=\Sigma_{k=1}^{p}a_{kq}$ with $a_{kq}=a_k b_q$ for $q<k$ and $a_{kq}=0$ otherwise, $\tilde{b}_p=\tilde{a}_{Pp}$, and the condition $\Sigma_{p=1}^{P}a_p=1$ has been used.

Formally putting $\xi_q \equiv 0$ reduces Eqs. (35) and (36) to the explicit symplectic Runge-Kutta-Nyström (SRKN) form [8] corresponding to standard nongradient decomposition schemes. Note that to avoid confusion we should distinguish between three different types of algorithms—namely, the traditional explicit nonsymplectic Runge-Kutta (referred to as RK) [1–3], the more recent implicit symplectic Runge-Kutta (referred to as IRK) [48], and the modern SRKN [8] ones. The advanced gradientlike approach proposed cannot be fitted to any previously known integration scheme. It constitutes a new class of decomposition integrators which will be referred to as the extrapolated gradientlike algorithms. They generalize both the nongradient- and gradient-SRKN-type schemes. Like the nongradient scheme, the advanced formulation is free from any gradient calculations and involves only force evaluations. At the same time it exhibits all the advantages of the gradient approach (since the decomposition coefficients c_p may accept nonzero values—i.e., $\xi_q \neq 0$). The latter can be reproduced by expanding the extrapolated acceleration over the displacement $\xi_q \mathbf{a}(\mathbf{r}^{(q)},t_q)\Delta t^2$ around the original position $\mathbf{r}^{(q)}$ by restricting ourselves to first-order spatial derivatives.

III. DERIVATION OF ALGORITHMS

A. General classification

The force-gradient algorithms corresponding to the standard decomposition scheme [Eq. (8)] have been comprehensively derived and classified in Ref. [47] up to $P=6$ stages and order $K=6$. The classification within the advanced gradientlike decomposition approach [Eq. (19)] will be similar for orders $K=2$ and $K=4$ but may differ for $K=6$. This is so because the new D_k terms ($k\geq 2$) appear in the error functions E_{2k+1} only beginning from the fifth order [Eqs. (21)–(23)]. For the same reason, all the decomposition coefficients a_p, b_p, and c_p ($p=1,2,\ldots,P$) obtained in Ref. [47] for gradient schemes of orders $K=2$ and $K=4$ with any number P of stages considered are applicable for the corresponding extrapolated gradientlike algorithms. However, the D_2 term may influence on the norm $\mathcal{O}_5=(\gamma_1^2+\gamma_2^2+\gamma_3^2+\gamma_4^2+\bar{\eta}_1^2+\gamma_4\bar{\eta}_1)^{1/2}$ of the error function E_5. Therefore, the extrapolated fourth-order algorithms can be optimized additionally by minimizing $\mathcal{O}_5$.

The above form of $\mathcal{O}_5$ follows from the uniform weight concept (see below) and the mutual independence of the fifth-order commutators arising at functions γ_{1-4} in Eq. (22). At the same time, the commutators C_2 at γ_4 and D_2 at η_1 are not independent and connected according to Eq. (30). This has been taken into account by including two terms with $\bar{\eta}_1=\eta_1/4$ into $\mathcal{O}_5$. Similarly, the norm of the seventh-order truncation uncertainties E_7 can be introduced as $\mathcal{O}_7=[\Sigma_{k=1}^{10}\zeta_k^2+\eta_2^2(1/36^2+1/24^2)-\eta_2(\zeta_1/36+\zeta_2/24)+(\eta_3/4)^2-\zeta_6\eta_3/4]^{1/2}$, where the relations [Eq. (30)] between the commutators $C_3^{\mathrm{I,II}}$ at $\gamma_{1,2}$ and D_3 at η_2 as well as between $[A,[A,D_2]]$ at η_3 and $[A,[B,[A,[B,[A,[B,A]]]]]]\equiv[A,[A,C_2]]$ at ζ_6 have been used. For $K=2$, the norm $\mathcal{O}_3=(\alpha^2+\beta^2)^{1/2}$ of the third-order error function E_3 is exactly the same as for standard gradient algorithms (when $\eta_{1-3}\equiv 0$).

In rare cases, an extra optimization can be performed exploiting specific features of extraordinary systems. In particular, for order $K=4$ we could introduce weights g_{1-5} at γ_{1-4} and η_1 when constructing the norm $\mathcal{O}_5$ and analyze a relative contribution of each commutator into the full function E_5 to determine g_{1-5}. However, this cumbersome analysis leads, as a rule, only to a slight improvement of the precision and cannot be realized, in general, for interacting systems with arbitrary interparticle potentials. As has been shown in the case of MD simulations [46,47], an efficient optimization of decomposition algorithms can be done within the uniform weight concept ($g_k\equiv 1$). It allows one to obtain uniquely decomposition coefficients and evaluates a minimum which almost coincides with that related to actual errors arising during MD simulations.

The corresponding order conditions for each given number P of stages and order K have been solved with respect to unknown decomposition coefficients in the spirit of Ref. [47], taking into account the new generating functions η_{1-3} apart from the standard ones ν, σ, α, β, γ_{1-4}, and ζ_{1-10}. Thus, the second-order algorithms were reproduced by letting $\nu=\sigma=1$. Algorithms of order 4 were derived by solving additionally the two conditions $\alpha=0$ and $\beta=0$. For order 6, they were supplemented by the five new members $\gamma_{1-4}=0$ and $\eta_1=0$ and so on for higher orders. With increasing the number P of stages at a given order K, the number of order conditions can be less than the number of decomposition coefficients. In such a case, the algorithms have been optimized by minimizing the leading error norm $\mathcal{O}_{K+1}$ with respect to the remaining $n>0$ free parameters. For $n=0$, the system of order conditions can have one or more sets of solutions. In the latter case, only the optimal set which minimizes $\mathcal{O}_{K+1}$ has been chosen. The results obtained within the advanced gradientlike approach [Eqs. (18) and (19)] are presented in Table I up to $P=5$ and order $K=6$.

The algorithms are shown in an abbreviated form, where the letters A and B designate the exponential operators $e^{Aa_p\Delta t}$ and $e^{Bb_p\Delta t}$, respectively, whereas the letter C denotes $\exp(Bb_p\Delta t+Cc_p\Delta t^3+\Sigma_{k=2}^{K/2-1}D_k d_{pk}\Delta t^{2k+1})$. Each group of algorithms corresponding to the same number $S=2P-1$ ($S=3$, 5, 7, or 9 for $P=2$, 3, 4, or 5) of exponential operators are separated by horizontal lines. Within the same group, the algorithms are allocated in the order of increasing K and the number n_{f} of force evaluations per step. For the same values of K and n_{f}, the velocity versions (where the letters C or B, but not A, appear first) are written before their position counterparts. The efficiency X_{Eff} of the algorithms has been measured using the formula $X_{\mathrm{Eff}}=1/(n_{\mathrm{f}}^K\mathcal{O}_{K+1})$ [47]. For the purpose of comparison, the number n_{g} of gradient evaluations per step related to the corresponding standard gradient scheme [Eq. (8)] is also included in Table I. Then the ratio of computational times spent during this standard scheme and the extrapolated gradientlike propagation can be estimated as $\kappa_\theta=[(n_{\mathrm{f}}-n_{\mathrm{g}})+\theta n_{\mathrm{g}}]/n_{\mathrm{f}}$, where $2\lesssim\theta\lesssim 5$ denotes the factor defining the relative cost spent on one gradient evaluation with respect to that of one force calculation [see Sec. II C and remember that in the extrapolated scheme the total number n_{f} of force evaluations consists of the number of genuine accelerations ($n_{\mathrm{f}}-n_{\mathrm{g}}$) plus the number n_{g} of modified ones, while the gradient evaluations are absent at all]. Even at $\theta=2$ (this value was used for the calculation of the efficiency of standard gradient algorithms in Ref. [47]), the extrapolated algorithms can be quicker up to $\kappa_2\sim 1.5$ times, while up to $\kappa_5\sim 3$ times at $\theta=5$. For the latter reason, the efficiency of the standard-gradient algorithms presented in Ref. [47] (see Table 2 there) should be in fact decreases by a factor of $\kappa_5/\kappa_2\sim 2$.

As can be seen from Table I, at $S\leq 9$ there are 30 possible self-adjoint combinations of exponential operators A, B, and C resulting in 29 different algorithms. Some among them are particularly outstanding since they reduce the norms of truncation errors in several orders without any additional computational costs. The particularly outstanding algorithms exhibiting the highest efficiency at each given number n_{f} of force evaluations per time step are highlighted in Table I by the bold font. The best second-order ($K=2$) algorithm is under No. 5 in Table I and corresponds to the extended nongradient propagation with $n_{\mathrm{f}}=2$. Its efficiency is approximately 3 times higher with respect to the Verlet integrators (Nos. 1 and 2) related to $n_{\mathrm{f}}=1$. This confirms the results obtained previously in Ref. [28]. In the case of order $K=4$, the pattern is quite different. Here, the best algorithms definitely

TABLE I. The complete family of extrapolated self-adjoint decomposition algorithms with up to nine exponential operators

Algorithm	Order	n_f	n	n_g	$\mathcal{O}_3$	$\mathcal{O}_5$	$\mathcal{O}_7$	Efficiency	Remarks	No
BAB	**2**	**1**	0	0	$\mathbf{9.3\times10^{-2}}$	$\mathbf{9.1\times10^{-3}}$	$\mathbf{1.3\times10^{-3}}$	**11**	[13,32]	**1**
ABA	2	1	0	0	9.3×10^{-2}	9.1×10^{-3}	1.3×10^{-3}	11	[13]	2
CAC	2	2	1	1	8.3×10^{-2}	1.3×10^{-2}	2.3×10^{-3}	3	New	3
ACA	2	2	1	1	4.2×10^{-2}	7.2×10^{-3}	8.0×10^{-4}	6	New	4
BABAB	**2**	**2**	1	0	$\mathbf{8.5\times10^{-3}}$	$\mathbf{1.0\times10^{-3}}$	$\mathbf{1.1\times10^{-4}}$	**29**	[28]	**5**
ABABA	2	2	1	0	8.5×10^{-3}	1.1×10^{-3}	1.1×10^{-4}	29	[28]	6
CABAC	4	3	1	1	0	3.6×10^{-3}	2.7×10^{-4}	3	New	7
BACAB	**4**	**3**	0	1	**0**	$\mathbf{8.1\times10^{-4}}$	$\mathbf{6.5\times10^{-5}}$	**15**	New	**8**
CACAC	4	4	1	2	0	6.8×10^{-4}	4.8×10^{-5}	6	New	9
ACACA	4	4	0	2	0	6.8×10^{-4}	5.6×10^{-5}	6	New	10
BABABAB	4	3	0	0	0	3.8×10^{-2}	1.3×10^{-2}	0.3	[10]	11
ABABABA	4	3	0	0	0	2.8×10^{-2}	6.3×10^{-3}	0.4	[10]	12
CABABAC	4	4	1	1	0	1.0×10^{-3}	4.3×10^{-5}	4	New	13
ABACABA	**4**	**4**	1	1	**0**	$\mathbf{1.4\times10^{-4}}$	$\mathbf{9.9\times10^{-6}}$	**27**	New	**14**
BACACAB	**4**	**5**	1	2	**0**	$\mathbf{5.4\times10^{-5}}$	$\mathbf{4.5\times10^{-6}}$	**30**	New	**15**
ACABACA	4	5	1	2	0	9.2×10^{-5}	8.0×10^{-6}	17	New	16
CACACAC	4	6	2	3	0	4.2×10^{-5}	6.0×10^{-6}	18	New	17
ACACACA	4	6	2	3	0	3.0×10^{-5}	4.0×10^{-6}	25	New	18
BABABABAB	4	4	1	0	0	6.5×10^{-4}	6.5×10^{-5}	6	New	19
ABABABABA	4	4	1	0	0	6.1×10^{-4}	4.6×10^{-5}	6	New	20
BABACABAB	4	5	2	1	0	6.4×10^{-5}	5.5×10^{-6}	25	New	21
CABABABAC	4	5	2	1	0	4.0×10^{-4}	2.9×10^{-5}	4	New	22
CABACABAC	4	6	3	2	0	2.4×10^{-5}	7.1×10^{-6}	32	New	23
BACABACAB	**4**	**6**	2	2	**0**	$\mathbf{1.5\times10^{-5}}$	$\mathbf{1.7\times10^{-6}}$	**52**	New	**24**
ABACACABA	4	6	2	2	0	3.4×10^{-5}	1.8×10^{-6}	23	New	25
ACABABACA	4	6	2	2	0	5.0×10^{-5}	4.8×10^{-6}	15	New	26
CACABACAC	4	7	3	3	0	1.5×10^{-5}	2.1×10^{-6}	28	New	27
ACACACACA	4	8	3	4	0	7.6×10^{-6}	2.1×10^{-6}	32	New	28
CACACACAC ≡ *BACACACAB*	6	7	−1	3	0	0	1.5×10^{-3}	0.006	New	29

belong to the extrapolated gradientlike decomposition class. They include the algorithms under Nos. 8, 14, 15, and 24, related to the cases $n_f=3$, 4, 5, and 6, respectively. The efficiency of these integrators may exceed the efficiency of the corresponding best nongradient ($n_g=0$) algorithms under No. 12 ($n_f=3$) and No. 20 ($n_f=4$) from several times up to two orders. At $n_f=7$, we come to the sixth-order algorithm (No. 29).

It is worth pointing out that the efficiency of the most outstanding extrapolated algorithms of order $K=4$ increases from $X_{\mathrm{Eff}}=15$ (No. 8) to $X_{\mathrm{Eff}}=52$ (No. 24) with increasing the number of exponential operators from $S=5$ to 9. This does not mean, however, that the algorithms with the highest efficiency should be considered as the best in general. The reason is that with rising S, the number n_f of force evaluations and thus the computational efforts increase as well. Of course, this increase is overcompensated by a much rapid decrease of the truncation errors $\mathcal{O}_{K+1}$, as this follows from the structure of the formula $X_{\mathrm{Eff}}=1/(n_f^K\mathcal{O}_{K+1})$. For instance, the increase of X_{Eff} in a certain number of times at a given n_f merely indicates that the precision (which is inverse proportional to $\mathcal{O}_{K+1}$) can be improved in the same number of times within the same computational costs. But when such a very high accuracy is not required, as in the case of MD simulations, we can restrict ourselves to the extrapolated gradientlike algorithm No. 8 with three force evaluations per step. The higher-precision algorithms Nos. 14, 15, and 24 can be used in CM applications.

B. Decomposition coefficients

The explicit values of decomposition coefficients corresponding to the particularly outstanding algorithms Nos. 1, 5, 8, 14, 15, 24, and 29 are presented below for $n_f=1$, 2, 3, 4, 5, 6, and 7, respectively, and orders $K=2$, 4, and 6. The nongradient VV algorithm No. 1 and its extended version No. 5

were introduced earlier [13,28], while algorithms Nos. 8, 14, 15, 24, and 29 belong to the extrapolated gradientlike decomposition class [Eqs. (18) and (19)]. Note that the decomposition coefficients of the extrapolated algorithms Nos. 8 and 29 coincide with those obtained within the standard gradient approach [47]. The reason is that in these cases $n \leq 0$, so that there are no free parameters to perform additional optimization of $\mathcal{O}_5$. For extrapolated algorithms Nos. 14, 15, and 24, when $n>0$, such an optimization results in new decomposition coefficients.

1. One force per step, order 2

Here $P=2$, $K=2$, and $n_f=1$, and we reproduce the well-known nongradient velocity Verlet algorithm [13] (No. 1 in Table I) with the following decomposition coefficients:

$$a_1 = 0, \quad b_1 = b_2 = \frac{1}{2}, \quad c_1 = c_2 = 0, \quad a_2 = 1. \tag{37}$$

2. Two forces per step, order 2

The extended second-order nongradient VV version (No. 5) is obtained [28] at $P=3$, $K=2$, and $n_f=2$ with

$$a_1 = 0, \quad b_1 = b_3 = 0.1931833275037836,$$

$$a_2 = a_3 = \frac{1}{2}, \quad b_2 = 1 - 2b_1, \quad c_1 = c_2 = c_3 = 0. \tag{38}$$

3. Three forces per step, order 4

Increasing the number of force evaluations per step to $n_f=3$ gives the possibility at $P=3$ to raise the order to $K=4$, and we come to the extrapolated gradientlike algorithm No. 8 with

$$a_1 = 0, \quad b_1 = b_3 = \frac{1}{6}, \quad c_1 = c_3 = 0$$

$$a_2 = a_3 = \frac{1}{2}, \quad b_2 = 1 - 2b_1, \quad c_2 = \frac{1}{72}. \tag{39}$$

Because here $n=0$, the decomposition coefficients [Eq. (39)] coincide with those obtained previously by Suzuki [33] and Chin [35] in the context of the standard gradient method. But using the same coefficients within the extrapolated scheme allows us to improve the efficiency approximately 3 times (cf. with line No. 8 in Table 2 of Ref. [47]).

4. Four forces per step, order 4

Raising the number of stages to $P=4$ yields the best algorithm No. 14 of order $K=4$ with $n_f=4$ force evaluations per time step with the following coefficients:

$$a_1 = 0.0929644676988989 = a_4,$$

$$b_1 = 0.2514919641733771 = b_3,$$

$$c_1 = 0 = c_3,$$

$$a_2 = 0.4070355323011011 = a_3,$$

$$b_2 = 0.4970160716532459,$$

$$c_2 = 0.0067114358547654,$$

$$b_4 = 0,$$

$$c_4 = 0. \tag{40}$$

They differ (because now $n>0$) with respect to those of the corresponding standard gradient scheme, and even at $\theta=2$ the gain in efficiency consists of 2.4—namely $X_{\mathrm{Eff}}=27$ versus $X_{\mathrm{Eff}}=11.3$ (line No. 14 in Table 2 of Ref. [47]).

5. Five forces per step, order 4

At $P=4$, the precision of the fourth-order ($K=4$) propagation can be improved further by rising the number of forces to $n_f=5$. Then one finds the algorithm No. 15 with

$$a_1 = 0,$$

$$b_1 = 0.0865575774230974 = b_4,$$

$$c_1 = 0 = c_4,$$

$$a_2 = 0.2799084428277709 = a_4,$$

$$b_2 = 0.4134424225769026 = b_3,$$

$$c_2 = 0.0030896906707527 = c_3,$$

$$a_3 = 0.4401831143444583. \tag{41}$$

Again this is a completely new algorithm with new coefficients (because of $n>0$), and the improvement in the efficiency over the corresponding standard gradient algorithm (No. 15 in Ref. [47]) consists of $30/9.4 \sim 3$.

6. Six forces per step, order 4

The highest precision for order $K=4$ is achieved at $P=5$ and $n_f=6$ resulting in the extrapolated algorithm No. 24 with the following new ($n>0$) decomposition constants:

$$a_1 = 0,$$

$$b_1 = 0.0692875001184967 = b_5,$$

$$c_1 = 0 = c_5,$$

$$a_2 = 0.2217840444237697 = a_5,$$

$$b_2 = 0.3145154584255536 = b_4,$$

$$c_2 = 0.0016168084229208 = c_4,$$

$$a_3 = 0.2782159555762303 = a_4,$$

$$b_3 = 0.2323940829118995,$$

$$c_3 = 0. \tag{42}$$

Here the efficiency is better with respect to the gradient case by a factor of $52/37.6 \sim 1.4$ (Ref. [47], line No. 24 of Table 2).

7. Seven forces per step, order 6

Finally, letting the number of force evaluations to $n_f=7$ allows one at $P=5$ to increase the order of precision to $K=6$. The corresponding algorithm No. 29 reads

$$a_1 = 0,$$

$$b_1 = 0.3599508087941436 = b_5,$$

$$c_1 = 0 = c_5,$$

$$a_2 = 1.0798524263824309 = a_5,$$

$$b_2 = -0.1437147273026540 = b_4,$$

$$c_2 = -0.0139652542242388 = c_4,$$

$$a_3 = -0.5798524263824309 = a_4,$$

$$b_3 = 0.5675278370170208,$$

$$c_3 = -0.0392470293823456. \quad (43)$$

Despite the fact that here the number of decomposition coefficients is less by 1 ($n=-1$) with respect to the number of order conditions due to the presence of the additional requirement $\eta_1=0$, the decomposition coefficients [Eq. (43)] appear to be the same as in the standard gradient case [47]. The reason is a fortunate cancellation of terms with opposite signs in function η_1 [Eq. (28)]. Notice that some of the coefficients in Eq. (43) are negative. This is contrary to the decomposition constants of Eqs. (37)–(42) which are all positive.

IV. APPLICATIONS

The extrapolated gradientlike algorithms will now be tested in MD and CM simulations to confirm our theoretical predictions. The most outstanding fourth-order algorithm No. 8 with three force evaluations per step will be applied to MD runs, while the more complicated schemes Nos. 14, 15, and 24 of order 4 as well as No. 29 of order 6 will be examined in CM calculations.

A. MD simulations

Here the system considered was a Lennard-Jones (LJ) fluid modeled by a collection of $N=512$ identical ($m_i \equiv m$) particles interacting through the modified potential $\varphi(r) = \Phi(r) - \Phi(r_c) - (r-r_c)\Phi'(r_c)$ for $r<r_c$ and $\varphi(r)=0$ otherwise, where $\Phi(r)=4\varepsilon[(\sigma/r)^{12}-(\sigma/r)^6]$ denotes the original LJ function. The particles were placed in a cubic box of volume $V=L^3$ in the absence of external fields ($u \equiv 0$). The periodic boundary conditions as well as the above modification of $\Phi(r)$ with $r_c=L/2 \approx 4.23\sigma$ have been used to reduce finite-size effects. The simulations were carried out in a microcanonical ensemble at a reduced density of $n^*=\frac{N}{V}\sigma^3 = 0.845$ and a reduced temperature of $T^*=k_BT/\varepsilon=1.7$ (a typical thermodynamic point of the LJ fluid). All MD tests were started from an identical well equilibrated initial configuration $\boldsymbol{\rho}(0)$ and then were continued over 100 000 or 1 000 000 time steps. The precision of the algorithms was measured in terms of the ratio $\mathcal{R}(t)=\mathcal{H}(t)/\mathcal{U}(t)$ of the relative total energy fluctuations $\mathcal{H}(t)=\langle(H-\langle H\rangle_t)^2\rangle_t^{1/2}/|\langle H\rangle_t|$ to the fluctuations $\mathcal{U}(t)=\langle(U-\langle U\rangle_t)^2\rangle_t^{1/2}/|\langle U\rangle_t|$ of the potential energy $U=\frac{1}{2}\Sigma_{i\neq j}^N\varphi(r_{ij})$ [see Eq. (1)], where $\langle S\rangle_t=\frac{1}{l}\Sigma_{k=1}^l S_k$ denotes the cumulative averaging of quantity S (with $S\equiv H$ or U) during time t along the produced trajectories, S_k is the current value of S at the kth time step, and $l=t/\Delta t$ is the total number of steps corresponding to t. Note that in microcanonical ensembles the total energy is an integral of motion, $\langle H\rangle_t=H$, and thus smaller values of $\mathcal{R}$ indicate a better precision of the integration.

In the MD simulations, the equations of motion were solved at two typical dimensionless time steps of $\Delta t^* = \Delta t(\varepsilon/m\sigma^2)^{1/2}=0.0025$ and $\Delta t^*=0.005$ using the most outstanding extrapolated gradientlike algorithm No. 8 of order 4 and its usual gradient counterpart. For the purpose of comparison, the runs corresponding to standard nongradient integrators were also performed. They include the traditional explicit Runge-Kutta schemes [1,8] of the second (RK2, two forces per step) and fourth (RK4, four forces) orders as well as their more recent [48] implicit versions IRK2 and IRK4 (requiring one and two force evaluations per iteration, respectively), the original (No. 1) [13,32] and extended (No. 5) [28] velocity Verlet algorithms (VV and EVV, one and two forces per step, respectively) of order 2, the popular second-order Störmer-Verlet (SV) algorithm [51] (one force per step), the fourth-order integrator by Cowel and Numerov (CN) [52] (one force per iteration), and the fourth-order algorithm by Forest and Ruth (FR) [10] (three forces per step, No. 12 in Table I). For the implicit integrators IRK2, IRK4, and CN, the number of iterations varied from 3 to 5 in dependence on the convergence conditions.

The ratios $\mathcal{R}(t)$ of energy fluctuations obtained in each of the cases are shown in Fig. 1 as functions of the length $l=t/\Delta t$ of the MD runs at $\Delta t^*=0.0025$ [subset (a)] and $\Delta t^*=0.005$ [subset (b)]. As can be seen, the RK2 and RK4 integrators exhibit very poor stability and conservative properties. Here the function $\mathcal{R}(t)$ increases with increasing the observation time t by several orders. This is despite the fact that for these integrators we used even smaller time steps—namely, $\Delta t^*=0.001$ and $\Delta t^*=0.0025$, respectively—contrary to those $\Delta t^*=0.0025$ and $\Delta t^*=0.005$ applied in all the rest cases. It has been mentioned in the Introduction that the instability follows from the nonsymplecticity of the RK algorithms, preventing us from using them in long-duration MD simulations. A similar instability is exhibited by conditionally symplectic IRK2 and IRK4 algorithms when the number of iterations is relatively small. For example, the IRK4 algorithm with three iterations is clearly bad (see the dashed curve marked by IRK4′), and only with rising to five iterations (the solid curve marked by IRK4) the stability is recovered. Note that with such a rise the computational efforts increase proportionally, resulting in an unacceptable large number ($5\times 2=10$) of force evaluations per time step. For this reason, the IRK algorithms also cannot be recommended

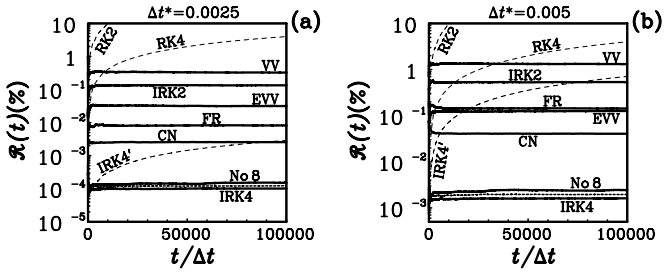

FIG. 1. The ratio of energy fluctuations as a function of the length of the MD simulations carried out at two different time steps, $\Delta t^*=0.0025$ [subset (a)] and $\Delta t^*=0.005$ [subset (b)] using various standard nongradient integrators, as well as the most outstanding extrapolated gradientlike algorithm No. 8 (solid curves) and its usual gradient counterpart (short-dashed curves). The standard set includes the classical explicit (long-dashed curves) and implicit (solid curves) Runge-Kutta schemes of the second (RK2 and IRK2) and fourth (RK4 and IRK4) orders and the second-order velocity Verlet (VV) algorithm and its extended (EVV) version, as well as the Cowell-Numerov (CN) and Forest-Ruth (FR) integrators (solid curves) (see the text for other explanations).

for efficient producing of MD trajectories in many-body systems. A somewhat better convergence of iterations by smaller computational costs is observed in the case of the CN integrator. The mean numbers with 3.0 (at $\Delta t^*=0.0025$) and 4.4 (at $\Delta t^*=0.005$) iterations (resulting in the same number of force evaluations per step) were sufficient here to achieve the required stability.

The standard integrators are, however, much less efficient than the extrapolated gradientlike decomposition algorithm No. 8. Indeed, as can be seen from Fig. 1, this algorithm with $n_f=3$ force evaluations per step reduces the energy fluctuations significantly—namely, up to a level comparable with that of the cumbersome IRK4 integrator requiring $n_f=10$ forces (see above). On the other hand, the best previously known decomposition FR algorithm (No. 12) which requires the same computational work (three force evaluations per step) is also obviously worse. The ratio of energy fluctuations for such an algorithm consists of about $\mathcal{R}\sim 0.1\%$, which is in two order larger with respect to the value $\mathcal{R}\sim 0.001\%$ corresponding to algorithm No. 8. Therefore, for $n_f=3$, the proposed algorithm is indeed the best. For a lower accuracy level of $\mathcal{R}\sim 0.1\%-1\%$ (which is acceptable in MD applications dealing with the study of simple structural and thermodynamic properties) it is quite sufficient to apply second-order algorithms No. 1 (VV) or No. 5 (EVV) which are the best in the cases $n_f=1$ and $n_f=2$, respectively. For higher accuracy, the preference should be given to the extrapolated gradientlike algorithm No. 8. Such an algorithm can be used in precise MD applications for the investigation of complicated properties (such as phase coexistence quantities, for instance) as well as for the observation of subtle effects which are very sensitive to the accuracy of the integration of the equations of motion (especially when approaching critical regions).

It can be verified readily that the level $\mathcal{R}(t)$ of the energy fluctuations obtained at the end of the simulations ($t\gg\Delta t$) corresponding to algorithm No. 8 is proportional to $\mathcal{R}(t)\sim\Delta t^4$ like in the FR case, while $\mathcal{R}(t)\sim\Delta t^2$ for the VV and EVV integrators. In particular, the fluctuations decrease from $\mathcal{R}\approx(2.4\times 10^{-3})\%$ to $\mathcal{R}\approx(1.5\times 10^{-4})\%$ when decreasing the size of the time step twice from $\Delta t^*=0.005$ to $\Delta t^*=0.0025$—i.e., in $24/1.5=2^4$ times—as this should be for the fourth-order integrator No. 8. Moreover, the ratio $\mathcal{R}/\Delta t^K$ (where $K=4$ or 2) is in turn proportional to the leading norm $\mathcal{O}_{K+1}$ of truncation errors presented in Table I and the coefficient of this proportionality is approximately the same for each of the algorithms. Therefore, our error-function theory developed in Sec. II for estimation of the efficiency of the decomposition integrators is in excellent agreement.

The results related to the standard gradient counterpart of algorithm No. 8 (which then reduces to the Suzuki-Chin integrator [35] with two forces and one force-gradient evaluations per step) are also plotted in Fig. 1 by short-dashed curves. As can be seen, such curves lie very close to those of the extrapolated gradientlike algorithm. This confirms the fact that the avoidance of time-consuming gradient evaluations has been performed without decrease of accuracy (order) of the integration.

It is interesting to remark that the extrapolated acceleration methodology proposed in Sec. II for obtaining gradientlike decomposition algorithms can be successfully used to improve the efficiency of some nondecomposition integrators as well. In order to demonstrate this, one introduces a three parameters (ϑ,μ,ξ) family of algorithms described by the following position and velocity evaluations

$$\mathbf{r}(t+\Delta t)=-\mathbf{r}(t-\Delta t)+2\mathbf{r}(t)+\Delta t^2[\vartheta\mathbf{a}_\xi(t-\Delta t)+(1-2\vartheta)\mathbf{a}_\xi(t)+\vartheta\mathbf{a}_\xi(t+\Delta t)],$$

$$\mathbf{v}(t)=\frac{\mathbf{r}(t+\Delta t)-\mathbf{r}(t-\Delta t)}{2\Delta t}-\mu[\mathbf{a}_\xi(t+\Delta t)-\mathbf{a}_\xi(t-\Delta t)]\frac{\Delta t}{12}, \tag{44}$$

where $\mathbf{a}_\xi(\tau)=\mathbf{a}(\mathbf{r}(\tau)+\xi\Delta t^2\mathbf{a}(\mathbf{r}(\tau)))$ denotes the extrapolated acceleration. For $\xi=0$, $\vartheta=1/12$, and $\mu=1$ one reproduces immediately the fourth-order implicit CN integrator. The explicit second-order SV algorithm is obtained from Eq. (44) at $\xi=0$, $\vartheta=0$, and $\mu=0$. Note that the CN and SV algorithms are not self-started, since require values $\mathbf{r}(\tau)$ of position from two previous steps $\tau=t$ and $\tau=t-\Delta t$ to calculate its current value at $\tau=t+\Delta t$, while velocity is not involved in the integration process. This is contrary to decomposition integrators which are one step by definition and involve both position and velocity in the propagation [see Eq. (35), for instance]. Note also that when the initial values $\mathbf{r}(t)$ and $\mathbf{r}(t-\Delta t)$ at the very beginning of the integration satisfy a certain relation, the VV and SV algorithms will produce exactly the same [51] phase trajectory $\boldsymbol{\rho}(t)$ for any time t. In our case we used the one-step IRK4 integrator for first two steps to start the two-step integration [Eq. (44)]. It has been realized that after a fast transient in several time steps, the SV and VV trajectories become indistinguishable (the difference between the SV and VV curves is not visible in scale of Fig. 1). A very surprising result is obtained when modifying ($\xi\neq 0$) the accelerations—namely, at $\xi=-1/12$, $\vartheta=0$, and $\mu=1/2$. This

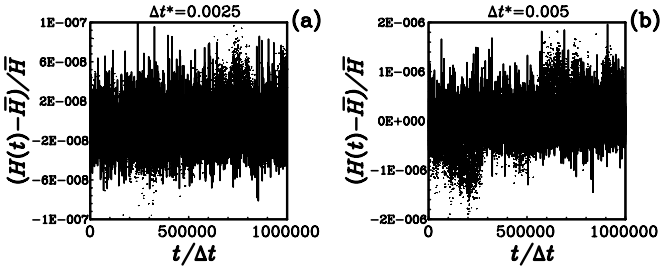

FIG. 2. The normalized deviations of the total energy versus the length of the MD simulations performed at two different time steps, $\Delta t^*=0.0025$ [subset (a)] and $\Delta t^*=0.005$ [subset (b)] using the extrapolated gradientlike algorithm No. 8 (dots) and its second-level acceleration counterpart (solid curves).

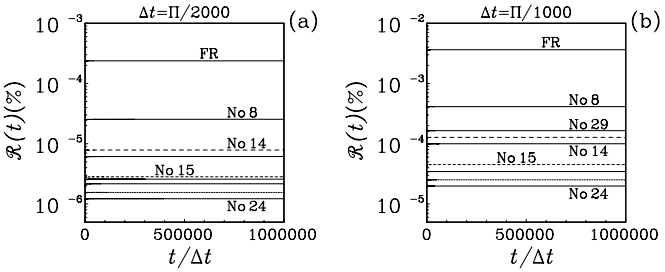

FIG. 3. The ratio of energy fluctuations obtained in the CM simulations at two time steps, $\Delta t=\Pi/2000$ [subset (a)] and $\Delta t=\Pi/1000$ [subset (b)] using the standard fourth-order integrator by Forest and Ruth (FR), the most outstanding extrapolated gradientlike algorithms Nos. 8, 14, 15, and 24 of order 4, and the sixth-order scheme No. 29 (see the corresponding solid curves). The usual gradient counterparts of algorithms Nos. 14, 15, and 24 are plotted by the long-, medium-, and short-dashed curves, respectively.

specific choice will be referred to as the advanced SV (ASV) algorithm. A characteristic feature of this algorithm is that (unlike the SV integrator) conservation of the total energy is satisfied with the fourth-order precision due to a fortunate cancellation of truncation error terms, despite the fact that position and velocity are still produced with the second-order accuracy (like in the SV case). At the same time, the ASV algorithm requires only two force evaluations per step [related to the original $\mathbf{a}(\mathbf{r}(t))$ and modified $\mathbf{a}_\xi(t)=\mathbf{a}(\mathbf{r}(t)+\xi\Delta t^2\mathbf{a}(\mathbf{r}(t)))$ accelerations], while the CN integrator involves three force recalculations per iteration. The function $\mathcal{R}(t)$ corresponding to the ASV algorithm practically coincides with the CN one (small deviations are only within the first thousand of steps) and thus is omitted in Fig. 1 to simplify the graph presentation.

The extrapolated gradientlike algorithm No. 8 exhibits also excellent stability properties in extremely long-duration simulations. In order to demonstrate this, additional MD runs have been performed with 1 000 000 time steps. In such runs, the normalized deviations $[H(t)-\bar{H}]/\bar{H}$ of the current total energy $H(t)$ from its mean value $\bar{H}$ (averaged during the total number of steps) have been measured at each time step $k=t/\Delta t=1,2,\ldots,1\,000\,000$. The second-level acceleration scheme [Eq. (33)] corresponding to the extrapolated algorithm No. 8 was tested in such MD simulations too. The results on this are shown in subsets (a) and (b) of Fig. 2 for $\Delta t^*=0.0025$ and $\Delta t^*=0.005$, respectively. As can be seen, despite some small local dispersion in the distribution of $[H(t)-\bar{H}]/\bar{H}$ related to the (first-level) extrapolated algorithm No. 8 [see dots in Fig. 2 which are masked by solid curves when $H(t)$ approaches $\bar{H}$], the total energy continues to fluctuate around the same value $\bar{H}$ even after 1 000 000 time steps. The dispersion amplitude can be reduced approximately twice by utilizing the second-level acceleration scheme (solid curves in Fig. 2). On the other hand, the magnitude of energy deviations decreases nearly in 2^4 times with decreasing the size of the time step from $\Delta t^*=0.005$ to $\Delta t^*=0.0025$.

B. CM simulations

Here we analyzed a motion of a body (planet) of mass m_1 in the (gravitation) field $u(r)=-c/r$ of the central particle (Sun) with mass $m_2\gg m_1$, where $c>0$ denotes the intensity of the interaction. Neglecting ($\varphi=0$) the influence of all other ($i=3,4,\ldots,N$) bodies (planets, for which $m_i\ll m_2$) in the (solar) system, we come to the two-dimensional Kepler problem. Then the equations of motion become particularly simple,

$$\frac{d\mathbf{r}}{dt}=\mathbf{v},\quad \frac{d\mathbf{v}}{dt}=-\frac{\mathbf{r}}{r^3}, \tag{45}$$

where $\mathbf{r}=\mathbf{r}_1-\mathbf{r}_2$ and units in which the reduced mass $m_1m_2/(m_1+m_2)$ and interaction constant c are equal to 1 have been used. The quantity $H=\mathbf{v}^2/2-1/r$ should be associated now with the total energy, while $U=-1/r$ is its potential part. We applied the same initial conditions $\mathbf{r}(0)=(10,0)$ and $\mathbf{v}(0)=(0,1/10)$ as those employed by previous authors [35,36]. This corresponds to a highly eccentric ($e=0.9$) orbit and provides a nontrivial testing ground for the trajectory integration. The standard FR integrator, the most outstanding extrapolated gradientlike algorithms Nos. 8, 14, 15, and 24, as well as their usual gradient counterparts, and the sixth-order scheme No. 29 have been utilized to integrate the equations of motion [Eq. (45)].

The ratios $\mathcal{R}(t)$ of energy fluctuations obtained in the CM simulations are presented in Fig. 3 for two time steps, $\Delta t=\Pi/2000$ [subset (a)] and $\Delta t=\Pi/1000$ [subset (b)], where $\Pi=\pi/[2|H(0)|^3]^{1/2}$ denotes the period of the elliptical orbit. As in the case of MD simulations (see Fig. 1), the extrapolated gradientlike algorithm No. 8 ($n_f=3$) leads to much more precise integration with respect to the standard nongradient FR integrator ($n_f=3$). More accurate extrapolated gradientlike algorithms Nos. 14, 15, and 24 with $n_f=4$, 5, and 6 force evaluations, respectively, allow one to decrease the energy fluctuations additionally up to a negligible small level of $\mathcal{R}\sim 10^{-5}\%-10^{-6}\%$, making the integration almost exact. It is worth mentioning that the gradient counterpart of the extrapolated algorithm No. 8 represents the Suzuki-Chin (SC) integrator [35] (two forces and one gradient evaluations per step). It has been established that the energy fluctuations $\mathcal{R}(t)$ corresponding to the SC integrator are practically the

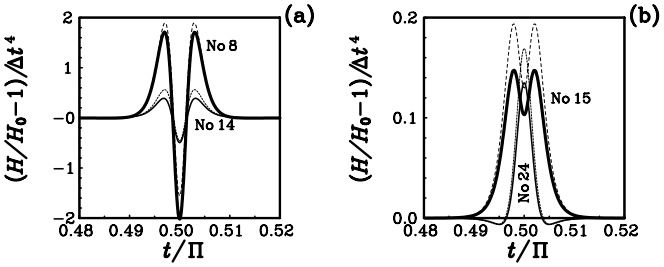

FIG. 4. The normalized total energy deviations in a Keplerian orbit. The results obtained within the extrapolated gradientlike algorithms Nos. 8, 14 [subset (a)], Nos. 15, 24 [subset (b)], and their usual gradient counterparts, are plotted by the solid and dashed curves, respectively.

same as those of the extrapolated gradientlike algorithm No. 8 (the deviations are invisible in the scale of Fig. 3). On the other hand, the usual gradient counterparts of the extrapolated algorithms Nos. 14, 15, and 24 are slightly worse (see the dashed curves in Fig. 3). The sixth-order integrator algorithm No. 29 is obviously poor. Despite the fact that it conserves energy even somewhat better than algorithm No. 8 (but worse with respect to the others at $\Delta t=\Pi/1000$), the former requires the largest number $n_f=7$ of force evaluations per step. With decreasing the time step to $\Delta t=\Pi/2000$ the precision of algorithm No. 29 becomes comparable with that of integrator No. 15, but again it remains inefficient because of too large a number of force recalculations.

Finally, the normalized total energy deviations $[H(t)/H(0)-1]/\Delta t^4$ obtained along the Keplerian orbit are plotted in Fig. 4 versus the reduced observation time t/Π during a period. Note that for fourth-order algorithms, such deviations should be independent of the size of the time step at sufficiently small values of Δt. This was observed in the vicinity of $\Delta t=\Pi/5000$, where the influence of higher-order terms becomes negligible. Note also that the energy deviations are substantial only near midperiods ($t/\Pi=0.5,1.5,2.5,\ldots$), when the body is at its closest position to the attractive center. It has been realized that all the algorithms considered conserve energy periodically. This is a consequence of the fact that in our case the periodic motion is integrated by symplectic and time-reversible algorithms. Another consequence is that the energy error returns to nearly zero after each period, as this is shown in Fig. 4 in the neighborhood of a midperiod. Moreover, then the energy fluctuations $\mathcal{R}(t)$ become bounded and independent of t when averaged over long times $t\gg\Pi$ [the independence of $\mathcal{R}(t)$ at $t\gg\Delta t$ has already been demonstrated in Fig. 3]. It can be seen in Fig. 4 that the energy deviations in the case of the extrapolated gradientlike algorithms are comparable or even smaller with respect to those of their usual gradient counterparts [the long-dashed curve in Fig. 4(a) corresponds to the SC integrator]. But the main advantage of the extrapolated algorithms is that they are free of any force-gradient evaluations.

V. CONCLUSION

In this work we have proposed an advanced gradientlike decomposition approach for the construction of highly efficient symplectic MD and CM algorithms. Like the standard gradient decomposition method, the approach presented is based on the factorization of the evolution propagator into exponential operators, but takes into account additional higher-order analytically integrable commutators under the exponentials. This has allowed one to express the exponential operators in terms of only force evaluations, maintaining at the same time the gradient structure of the factorization. In such a way, the time-consuming direct evaluations of force gradients have been avoided by force extrapolation without any loss of precision of the integration. Using the advanced approach, the extrapolated gradientlike decomposition algorithms have been explicitly derived and classified within up to a nine-exponential description and up to order 6 in the time step. The most outstanding algorithms obtained were applied to MD and CM simulations. Comparison with previously known well-established integrators has shown that the algorithms presented lead to more efficient integration, especially when moderate or high accuracy of the calculations is required.

The extrapolated gradientlike algorithms can easily be adapted for applications in other areas, such as quantum dynamics and hybrid Monte Carlo simulations. Moreover, in multiple-time-scale MD simulations, the expensive long-ranged forces can be handled by the usual Verlet algorithm, while the fastest processes caused by the (much less expensive) short-ranged interactions can be integrated using more accurate gradientlike algorithms. The proposed approach can also be extended to more complicated systems with orientational or spin degrees of freedom. These and other relevant problems will be the subject of a further investigation.

ACKNOWLEDGMENT

This work was supported in part by the Fonds zur Förderung der wissenschaftlichen Forschung under Project No. 18592-PHY.

[1] C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations* (Prentice-Hall, Englewood Cliffs, NJ, 1971).

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

[3] R. L. Burden and J. D. Faires, *Numerical Analysis*, 5th ed. (PWS, Boston, 1993).

[4] H. Kinoshita, H. Yoshida, and H. Nakai, Celest. Mech. Dyn. Astron. **50**, 59 (1991).

[5] B. Gladman, M. Duncan, and J. Candy, Celest. Mech. Dyn. Astron. **52**, 221 (1991).

[6] D. Frenkel and B. Smit, *Understanding Molecular Simulation:*

From Algorithms to Applications (Academic Press, New York, 1996).
[7] S. Reich, SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal. **36**, 1549 (1999).
[8] E. Hairer, C. Lubich, and G. Wanner, in *Springer Series in Computational Mathematics*, 1st ed. (Springer-Verlag, Berlin, 2002), Vol. 31.
[9] H. Yoshida, Phys. Lett. A **150**, 262 (1990).
[10] E. Forest and R. D. Ruth, Physica D **43**, 105 (1990).
[11] M. Suzuki, Phys. Lett. A **165**, 387 (1992).
[12] E. Forest, J. Comput. Phys. **99**, 209 (1992).
[13] M. Tuckerman, B. J. Berne, and G. J. Martyna, J. Chem. Phys. **97**, 1990 (1992).
[14] M. Qin and W. J. Zhu, Computing **47**, 309 (1992).
[15] M. Suzuki and K. Umeno, in *Computer Simulation Studies in Condensed Matter Physics VI*, edited by D. P. Landau, K. K. Mon, and H.-B. Schüttler (Springer, Berlin, 1993).
[16] P.-V. Koseleff, Lect. Notes Comput. Sci. **673**, 213 (1993).
[17] J. M. Sanz-Serna and M. P. Calvo, *Numerical Hamiltonian Problems* (Chapman and Hall, London, 1994).
[18] R.-C. Li, Ph.D. thesis, Department of Mathematics, University of California at Berkeley, 1995.
[19] R. I. McLachlan, SIAM (Soc. Ind. Appl. Math.) J. Sci. Stat. Comput. **16**, 151 (1995).
[20] P.-V. Koseleff, Fields Inst. Commun. **10**, 103 (1996).
[21] S. J. Stuart, R. Zhou, and B. J. Berne, J. Chem. Phys. **105**, 1426 (1996).
[22] W. Kahan and R.-C. Li, Math. Comput. **66**, 1089 (1997).
[23] A. Dullweber, B. Leimkuhler, and R. McLachlan, J. Chem. Phys. **107**, 5840 (1997).
[24] M. Krech, A. Bunker, and D. P. Landau, Comput. Phys. Commun. **111**, 1 (1998).
[25] A. Murua and J. M. Sanz-Serna, Philos. Trans. R. Soc. London, Ser. A **357**, 1079 (1999).
[26] S. Blanes and P. C. Moan, J. Comput. Phys. **170**, 205 (2001).
[27] I. P. Omelyan, I. M. Mryglod, and R. Folk, Phys. Rev. Lett. **86**, 898 (2001).
[28] I. P. Omelyan, I. M. Mryglod, and R. Folk, Phys. Rev. E **65**, 056706 (2002).
[29] I. P. Omelyan, I. M. Mryglod, and R. Folk, Comput. Phys. Commun. **146**, 188 (2002).
[30] S. A. Chin and C. R. Chen, J. Chem. Phys. **117**, 1409 (2002).
[31] R. I. McLachlan and G. R. W. Quispel, Acta Numerica **11**, 341 (2002).
[32] W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson, J. Chem. Phys. **76**, 637 (1982).
[33] M. Suzuki, in *Computer Simulation Studies in Condensed Matter Physics VIII*, edited by D. P. Landau, K. K. Mon, and H.-B. Schüttler (Springer-Verlag, Berlin, 1995).
[34] M. Suzuki, Phys. Lett. A **201**, 425 (1995).
[35] S. A. Chin, Phys. Lett. A **226**, 344 (1997).
[36] S. A. Chin and D. W. Kidwell, Phys. Rev. E **62**, 8746 (2000).
[37] S. R. Scuro and S. A. Chin, Phys. Rev. E **71**, 056703 (2005).
[38] A. N. Drozdov and J. J. Brey, Phys. Rev. E **57**, 1284 (1998).
[39] H. A. Forbert and S. A. Chin, Phys. Rev. E **63**, 016703 (2000).
[40] S. A. Chin and C. R. Chen, J. Chem. Phys. **114**, 7338 (2001).
[41] J. Auer, E. Krotscheck, and S. A. Chin, J. Chem. Phys. **115**, 6841 (2001).
[42] S. A. Chin, Phys. Rev. E **69**, 046118 (2004).
[43] G. Goldstein and D. Baye, Phys. Rev. E **70**, 056703 (2004).
[44] S. A. Chin and E. Krotscheck, Phys. Rev. E **72**, 036705 (2005).
[45] S. A. Chin and P. Anisimov, J. Chem. Phys. **124**, 054106 (2006).
[46] I. P. Omelyan, I. M. Mryglod, and R. Folk, Phys. Rev. E **66**, 026701 (2002).
[47] I. P. Omelyan, I. M. Mryglod, and R. Folk, Comput. Phys. Commun. **151**, 272 (2003).
[48] M. Sofroniou and W. Oevel, SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal. **34**, 2063 (1997).
[49] M. Suzuki, Proc. Jpn. Acad., Ser. B: Phys. Biol. Sci. **69**, 161 (1993).
[50] S. A. Chin, Phys. Rev. E **71**, 016703 (2005).
[51] A. K. Mazur, J. Comput. Phys. **136**, 354 (1997).
[52] R. D. Skeel, G. Zhang, and T. Schlick, SIAM (Soc. Ind. Appl. Math.) J. Sci. Stat. Comput. **18**, 203 (1997).