

Construction of high-order force-gradient algorithms for integration of motion in classical and quantum systems

I. P. Omelyan,^{1,2} I. M. Mryglod,^{1,2} and R. Folk²

¹*Institute for Condensed Matter Physics, 1 Svientsitskii Street, UA-79011 Lviv, Ukraine*

²*Institute for Theoretical Physics, Linz University, A-4040 Linz, Austria*

(Received 5 November 2001; published 2 August 2002)

A consequent approach is proposed to construct symplectic force-gradient algorithms of arbitrarily high orders in the time step for precise integration of motion in classical and quantum mechanics simulations. Within this approach the basic algorithms are first derived up to the eighth order by direct decompositions of exponential propagators and further collected using an advanced composition scheme to obtain the algorithms of higher orders. Contrary to the scheme proposed by Chin and Kidwell [Phys. Rev. E **62**, 8746 (2000)], where high-order algorithms are introduced by standard iterations of a force-gradient integrator of order four, the present method allows one to reduce the total number of expensive force and its gradient evaluations to a minimum. At the same time, the precision of the integration increases significantly, especially with increasing the order of the generated schemes. The algorithms are tested in molecular dynamics and celestial mechanics simulations. It is shown, in particular, that the efficiency of the advanced fourth-order-based algorithms is better approximately in factors 5 to 1000 for orders 4 to 12, respectively. The results corresponding to sixth- and eighth-order-based composition schemes are also presented up to the sixteenth order. For orders 14 and 16, such highly precise schemes, at considerably smaller computational costs, allow to reduce unphysical deviations in the total energy up in 100 000 times with respect to those of the standard fourth-order-based iteration approach.

DOI: 10.1103/PhysRevE.66.026701

PACS number(s): 02.60.Cb, 05.10.-a, 95.10.Ce, 95.75.Pq

I. INTRODUCTION

Understanding the dynamic phenomena in classical and quantum many-body systems is of importance in most areas of physics and chemistry. The development of efficient algorithms for solving the equations of motion in such systems should therefore impact a lot of fields of fundamental research. During the last decade a considerable activity [1–9] has been directed on the construction of symplectic time-reversible algorithms that employ decompositions of the evolution operators into analytically solvable parts. The decomposition algorithms exactly preserve all Poincaré invariants and, thus, are ideal for long-time integration in molecular dynamics [10] and astrophysical [11] simulations. The reason is that for these algorithms the errors in energy conservation appear to be bounded even for relatively large values of the size of the time step. This is in a sharp contrast to traditional Runge-Kutta and predictor-corrector schemes [12,13], where the numerical uncertainties increase linearly with increasing the integration time [9,14–17].

The main attention in previous investigations has been devoted to derive different-order decomposition algorithms involving only force evaluations during the time propagation. For instance, the widely used velocity- and position-Verlet algorithms [18,19] relate, in the general classification, to a three-stage decomposition scheme of the second order with one force evaluation per step. The fourth-order algorithm by Forest and Ruth [2] corresponds to a scheme with three such force recalculations and consists of seven single-exponential stages. Sixth-order schemes are reproduced [4,6] beginning from fifteen stages and seven evaluations of force for each body in the system per given time step. With further

increasing the order of force decomposition schemes, the number of stages and thus the number of the corresponding nonlinear equations (which are necessary to solve numerically to obtain the required time coefficients for single-exponential propagations) increases drastically. In addition, such equations become too cumbersome and all these, taking into account the capabilities of modern supercomputers, lead to the impossibility of representing the direct decomposition algorithms of order eight and higher in an explicit form [6]. In order to simplify this problem, it was proposed [1,3,5,6,20–23] to derive higher-order integrators by composing schemes of lower (actually second) orders. The resulting second-order-based composition algorithms have been explicitly obtained up to the tenth order [5,6,22].

Relatively recently [24–26], a deeper analysis of the operator factorization process has shown that the class of analytically integrable decomposition integrators can be extended including additionally a higher-order commutator into the single-exponential propagations. As a consequence, a set of so-called force-gradient algorithms of the fourth order has been introduced. A distinguishable feature of these algorithms is the possibility to generate solutions using only positive values for time coefficients during each substage of the integration. This is contrary to the original decomposition approach, where beyond second order (as has been rigorously proved by Suzuki [3]) any scheme expressed in terms of only force evaluation must produce some negative time coefficients. We mention that applying negative time propagations is impossible, in principle, in such important fields as nonequilibrium statistical mechanics, quantum statistics, stochastic dynamics, etc., because one cannot simulate diffusion or stochastic processes backward in time nor sample configurations with negative temperatures. In the case of stochastic dynamics simulations it has been demonstrated explicitly

[27,28] that using fourth-order force-gradient algorithms leads to much superior propagation over standard Verlet-based schemes of the second order in that it allows much larger time steps with no loss of precision. A similar pattern was observed in classical dynamics simulations comparing the usual fourth-order algorithm by Forest and Ruth with its force-gradient counterparts [26].

Quite recently, Chin and Kidwell [29] have considered a question of how to iterate the force-gradient algorithms to higher orders. The iteration was based on Creutz and Gocksch's approach [30] according to which an algorithm of order $K+2$ can be obtained by triplet construction of a self-adjoint (i.e., time-reversible) scheme of order K . Then starting from a fourth-order integrator, it has been shown in actual celestial mechanics simulations that for orders 6, 8, 10, and 12, the numerical errors corresponding to the force-gradient-based schemes are significantly smaller than those of the schemes basing on iterations of usual nongradient algorithms. The resulting efficiency of the integration has also increased considerably despite increased computational efforts spent on the calculations of force gradients. The same has been seen in the case of quantum mechanics simulations when solving the time-dependent Schrödinger equation [31].

It is worth emphasizing, however, that the iteration scheme proposed by Chin and Kidwell is far to be optimal for deriving high-order integrators belonging to the force-gradient class. The reason is that the number of total force and its gradient evaluations increases too rapidly with increasing K . Remembering that such evaluations constitute the most time-consuming part of the calculations, this may restrict the region of applicability of force-gradient algorithms to relative low orders only. Note that high-order computations are especially desirable in problems of astrophysical interest, because then one can observe a system over a very long period of time. They may also be useful in highly precise molecular dynamics and quantum mechanics simulations to identify or confirm very subtle effects.

In the present paper we propose a general approach to construction of symplectic force-gradient algorithms of arbitrary orders. The approach considers the splitting and composing of the evolution operators on the basic level, taking into account the explicit structure of truncation terms at each given order in the time step. This has allowed us to obtain exclusively precise and economical algorithms by using significantly smaller number of single-exponential propagations than that appearing within standard decomposition and iteration schemes. The paper is organized as follows. The equations of motion for classical and quantum systems are presented in Sec. II A. The integration of these equations by direct decompositions and their force-gradient generalization are described in Sec. II B. Explicit expressions for basic force-gradient algorithms of orders 2, 4, 6, and 8 are also given there. The higher-order integration based on advanced compositions of lower-order schemes is considered in Sec. II C. The composition constants for fourth-, sixth-, and eighth-order-based schemes are calculated and written in the same section up to the overall order 16. Sections III A and III B are devoted to applications of obtained force-gradient algorithms to molecular dynamics and celestial mechanics

simulations, respectively. A comparative analysis of the introduced algorithms with existing integrators is made there as well. The final discussion and concluding remarks are highlighted at the end in Sec. IV.

II. GENERAL THEORY OF CONSTRUCTION OF FORCE-GRADIENT ALGORITHMS

A. Basic equations of motion for classical and quantum systems

Consider first a classical N -body system described by the Hamiltonian

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

where \mathbf{r}_i is the position of particle i moving with velocity $\mathbf{v}_i = d\mathbf{r}_i/dt$ and carrying mass m_i , $\varphi(r_{ij}) \equiv \varphi(|\mathbf{r}_i - \mathbf{r}_j|)$ denotes the interparticle potential of interaction, and T and U relate to the total kinetic and potential energies, respectively. Then the equations of motion can be presented in the following compact form:

$$\frac{d\boldsymbol{\rho}}{dt} = [\boldsymbol{\rho} \circ H] \equiv L\boldsymbol{\rho}(t). \quad (2)$$

Here $\boldsymbol{\rho} = \{\mathbf{r}, \mathbf{v}\} \equiv \{\mathbf{r}_i, \mathbf{v}_i\}$ is the full set ($i=1, 2, \dots, N$) of phase variables, $[\circ]$ represents the Poisson bracket and

$$L = \sum_{i=1}^N \left(\mathbf{v}_i \frac{\partial}{\partial \mathbf{r}_i} + \frac{\mathbf{f}_i}{m_i} \frac{\partial}{\partial \mathbf{v}_i} \right) \quad (3)$$

is the Liouville operator with $\mathbf{f}_i = -\sum_{j(j \neq i)}^N \varphi'(r_{ij}) \mathbf{r}_{ij}/r_{ij}$ being the force acting on particles due to the interactions.

In the case of quantum systems, the state evolution can be described by the time-dependent Schrödinger equation,

$$i\hbar \frac{\partial \psi}{\partial t} = \mathcal{H}(\mathbf{r}) \psi \equiv [T + \mathcal{U}(\mathbf{r})] \psi, \quad (4)$$

where $T = -\frac{1}{2} \sum_{i=1}^N \hbar^2 \nabla_i^2 / m_i$ and \mathcal{U} are the kinetic and potential energy operators, respectively, and ψ is the wave function. The so-called quantum-classical dynamics models [32] can also be introduced. This leads to a coupled system of Newtonian (2) and Schrödinger (4) equations. But, in order to simplify notations, we restrict ourselves to the above purely classic and quantum considerations.

If an initial configuration $\boldsymbol{\rho}(0)$ or $\psi(0)$ is provided, the unique solution to Eq. (2) or (4) can be formally cast as

$$\mathcal{R}(t) = e^{\mathcal{L}t} \mathcal{R}(0) \equiv (e^{\mathcal{L}\Delta t})^l \mathcal{R}(0), \quad (5)$$

where Δt and $l = t/\Delta t$ are the size of the single time step and the total number of steps, respectively, \mathcal{R} denotes either $\boldsymbol{\rho}$ or ψ , whereas \mathcal{L} corresponds to L or $-i\mathcal{H}/\hbar$. As is well known, the time evolution of many-particle systems cannot be performed exactly in the general case. Thus, the problem arises on evaluating the propagator $e^{\mathcal{L}\Delta t}$ by numerical methods.

B. Integration by direct decompositions

1. Original decomposition approach

The main idea of decomposition integration consists in factorization of the full exponential operator $e^{\mathcal{L}\Delta t}$ on such subpropagators that allow to be evaluated analytically or at least be presented in quadratures. Within the original approach, 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 $\mathcal{A} = -i\mathcal{T}\hbar$ and $\mathcal{B} = \mathbf{a} \cdot \partial / \partial \mathbf{v}$ with $\mathbf{a} = \{\mathbf{a}_i\} = \{\mathbf{f}_i / m_i\}$ being the acceleration or $\mathcal{B} = -i\mathcal{U}\hbar$ for the cases of classical or quantum mechanics, respectively. Then, taking into account the smallness of Δt , the total propagator can be decomposed [1–3,5,24] using the formula

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

where the coefficients a_p and b_p are chosen in such a way so as to provide the highest possible value for $K \geq 1$ at a given integer number $P \geq 1$. As a result, integration (5) can be performed approximately with the help of Eq. (6) by neglecting truncation terms $O(\Delta t^{K+1})$. The precision will increase with increasing the order K and decreasing the size Δt of the time step.

As can be verified readily, the exponential subpropagators $e^{\mathcal{A}\tau}$ and $e^{\mathcal{B}\tau}$, appearing in the right-hand-side of Eq. (6), are analytically integrable for classical systems. Indeed, taking into account the independence of \mathbf{v} on \mathbf{r} and \mathbf{a} on \mathbf{v} yields

$$\begin{aligned} e^{\mathcal{A}\tau} \rho &\equiv e^{\tau \mathbf{v} \cdot \partial / \partial \mathbf{r}} \{\mathbf{r}, \mathbf{v}\} = \{\mathbf{r} + \mathbf{v}\tau, \mathbf{v}\}, \\ e^{\mathcal{B}\tau} \rho &\equiv e^{\tau \mathbf{a} \cdot \partial / \partial \mathbf{v}} \{\mathbf{r}, \mathbf{v}\} = \{\mathbf{r}, \mathbf{v} + \mathbf{a}\tau\}, \end{aligned} \quad (7)$$

which represent simple shift operators in position and velocity spaces, respectively, with τ being equal to $a_p\Delta t$ or $b_p\Delta t$. For quantum mechanics propagations, the kinetic part $e^{\mathcal{A}\tau} \equiv e^{-i\tau\mathcal{T}\hbar}$ will require carrying out two, one direct and one inverse, spatial Fourier transforms [31], whereas the calculation of $e^{\mathcal{B}\tau} \equiv e^{-i\tau\mathcal{U}\hbar}$ is trivial.

In view of decompositions (6), one can reproduce integrators of various orders in the time step. In particular, the well-known second-order ($K=2$) velocity-Verlet algorithm [19,18]

$$e^{(\mathcal{A}+\mathcal{B})\Delta t + O(\Delta t^3)} = e^{\mathcal{B}(\Delta t/2)} e^{\mathcal{A}\Delta t} e^{\mathcal{B}(\Delta t/2)} \quad (8)$$

is readily derived from Eq. (6) by putting $P=2$ with $a_1=0$, $b_1=b_2=1/2$, and $a_2=1$. The fourth-order ($K=4$) algorithm proposed by Forest and Ruth [2] is obtained from Eq. (6) at $P=4$ with $a_1=0$, $a_2=a_4=\theta$, $a_3=(1-2\theta)$, $b_1=b_4=\theta/2$, and $b_2=b_3=(1-\theta)/2$, where $\theta=1/(2-\sqrt[3]{2})$. Schemes of the sixth order ($K=6$) are derivable starting from $P=8$ with numerical representation of time coefficients [4,6].

The original decomposition approach has, however, a set of disadvantages. First of all, it is worth pointing out that with further increasing the order of integration (6) to $K=8$ and higher, the number $2P$ of unknowns a_p and b_p begins to

increase too rapidly. This leads to the impossibility of representing algorithms of such a type for $K>6$ in an explicit form [6], because it becomes impossible to solve the same number of the resulting cumbersome nonlinear equations (with respect to a_p and b_p) even using the capabilities of modern supercomputers. Another drawback consists in the fact that for $K>2$ it is impossible [3] at any P to derive from Eq. (6) a decomposition scheme with the help of only positive time coefficients. For example, in the case of Forest-Ruth integration, three of eight coefficients, namely, a_3 , b_2 , and b_3 , are negative. As was mentioned in the introduction, schemes with negative time coefficients have a restricted region of application and are not acceptable for simulating nonequilibrium statistical mechanics, quantum statistics, stochastic dynamics, and other important processes. Moreover, for schemes expressed in terms of force evaluation only, the main term $O(\Delta t^{K+1})$ of truncation uncertainties appears to be, as a rule, too big, resulting in a decrease in the efficiency of the computations.

2. Generalized force-gradient decomposition method

From the aforesaid, it is quite desirable to introduce a more general approach that is free of the above disadvantages. At the same time, this approach, like the original scheme, must be explicit, i.e., lead to analytical propagations. In addition, it is expected that the already known decomposition algorithms should appear from it as particular cases.

Let us first analyze the structure of third-order truncation errors $O(\Delta t^3)$ of the velocity-Verlet algorithm in detail. Expanding both the sides of Eq. (8) into Taylor's series with respect to Δt , one finds

$$O(\Delta t^3) = \left(\frac{1}{12} [\mathcal{A}, [\mathcal{A}, \mathcal{B}]] + \frac{1}{24} [\mathcal{B}, [\mathcal{A}, \mathcal{B}]] \right) \Delta t^3 + O(\Delta t^5), \quad (9)$$

where $[\cdot, \cdot]$ denotes the commutator of two operators. Taking into account the explicit expressions for operators \mathcal{A} and \mathcal{B} it can be shown that one of the two third-order operators in Eq. (9), namely $[\mathcal{B}, [\mathcal{A}, \mathcal{B}]]$, is relatively simple and, more importantly, it allows to be handled explicitly, contrary to the operator $[\mathcal{A}, [\mathcal{A}, \mathcal{B}]]$. In the case of classical systems it can be obtained readily that

$$\mathcal{C} \equiv [\mathcal{B}, [\mathcal{A}, \mathcal{B}]] = \sum_{i=1}^N \frac{\mathbf{g}_i}{m_i} \frac{\partial}{\partial \mathbf{v}_i} \equiv \mathbf{G} \frac{\partial}{\partial \mathbf{v}}, \quad (10)$$

where $\mathbf{g}_{i\alpha} = 2 \sum_{j\beta} \mathbf{f}_{j\beta} / m_j \partial \mathbf{f}_{i\alpha} / \partial \mathbf{r}_{j\beta}$. In view of the expression $\mathbf{f}_{i\alpha} = - \sum_{j(j \neq i)} \varphi'_{ij}(r_{ij}) (\mathbf{r}_{i\alpha} - \mathbf{r}_{j\alpha}) / r_{ij}$ for forces, the required force-gradient evaluations $\partial \mathbf{f}_{i\alpha} / \partial \mathbf{r}_{j\beta}$ are explicitly representable, i.e.,

$$\begin{aligned} \mathbf{g}_i &= -2 \sum_{j(j \neq i)}^N \left[(\mathbf{a}_i - \mathbf{a}_j) \frac{\varphi'_{ij}}{r_{ij}} + \frac{\mathbf{r}_{ij}}{r_{ij}^3} (r_{ij} \varphi''_{ij} - \varphi'_{ij}) [\mathbf{r}_{ij} \cdot (\mathbf{a}_i - \mathbf{a}_j)] \right] \\ &\equiv \sum_{j(j \neq i)}^N \mathbf{g}(\mathbf{r}_{ij}) = \mathbf{g}_i(\mathbf{r}). \end{aligned} \quad (11)$$

As can be seen easily from Eqs. (10) and (11), the operator \mathcal{C} commutes with $\mathcal{B} \equiv \mathbf{a} \cdot \partial / \partial \mathbf{v}$, and, in addition, the function \mathbf{G} like \mathbf{a} does not depend on velocity. Then the force-gradient part $\mathcal{C}\Delta t^3/24$ of truncation uncertainties (9) can be extracted by transferring them from the left-hand side of Eq. (8) to its right-hand side and further symmetrically collecting with operator \mathcal{B} under exponentials. This yields the following force-gradient version:

$$e^{(A+B)\Delta t + O(\Delta t^3)} = e^{B(\Delta t/2) - C(\Delta t^3/48)} e^{A\Delta t} e^{B(\Delta t/2) - C(\Delta t^3/48)} \quad (12)$$

of the velocity-Verlet integrator, where already $O(\Delta t^3) = [\mathcal{A}, [\mathcal{A}, \mathcal{B}]]\Delta t^3/12$.

In the case of higher-order ($K > 2$) integration (6), the operator \mathcal{C} will enter into truncation uncertainties $O(\Delta t^{K+1})$ by various combinations. They can be extracted similarly as for $K=2$, and we come to a force-gradient decomposition approach. The most general representation of this approach is

$$e^{(A+B)\Delta t + O(\Delta t^{K+1})} = \prod_{p=1}^P e^{A a_p \Delta t} e^{B b_p \Delta t + C c_p \Delta t^3}, \quad (13)$$

where again at a given P the coefficients a_p , b_p , as well as c_p must be chosen in such a way to cancel the truncation terms $O(\Delta t^{K+1})$ to the highest possible order K . For $c_p \equiv 0$, generalized factorization (13) reduces to usual representation (6). It is worth emphasizing that in view of the velocity independence of \mathbf{G} on \mathbf{v} , the modified operator of shifting velocities remains to be evaluated exactly for any b_p and c_p , namely,

$$e^{B b_p \Delta t + C c_p \Delta t^3} \{\mathbf{r}, \mathbf{v}\} = \{\mathbf{r}, \mathbf{v} + b_p \mathbf{a} \Delta t + c_p \mathbf{G} \Delta t^3\}. \quad (14)$$

For quantum systems, where $\mathcal{C} = i \sum_i |\nabla_i \mathcal{U}|^2 / (\hbar m_i)$, the corresponding calculations also present no difficulties (at least for particles in external fields), because this requires only knowing the gradient of the potential.

An important feature of decomposition integration (13) is that it, being applied to classical dynamics simulations, conserves the symplectic map of particle's flow in phase space. This is so because separate shifts of positions (7) and velocities (14) do not change the phase volume. The time reversibility $\mathcal{S}(-t)\mathcal{R}(t) = \mathcal{R}(0)$ of solutions [following from the property $\mathcal{S}^{-1}(t) = \mathcal{S}(-t)$ of evolution operator $\mathcal{S}(t) = e^{\mathcal{L}t}$] can be reproduced exactly as well by imposing additional constraints on the coefficients a_p , b_p , and c_p . In particular, for velocitylike decompositions such constraints read: $a_1 = 0$, $a_{p+1} = a_{p-p+1}$, $b_p = b_{p-p+1}$, and $c_p = c_{p-p+1}$. Then single-exponential subpropagators will enter symmetrically into the decompositions, providing automatically the required reversibility. The case when the operators of shifting velocity and position are replaced by each other in the resulting symmetrical decomposition is also possible. This leads to a positionlike integration that can be reproduced from Eq. (13) at $a_p = a_{p-p+1}$, $b_p = b_{p-p}$, and $c_p = c_{p-p}$ at $b_p = 0$ and $c_p = 0$.

The above symmetry will result in its turn to automatic disappearing of all even-order terms in the error function $O(\Delta t^{K+1})$. For this reason, the order K of time-reversible (self-adjoint) algorithms may accept only even numbers ($K = 2, 4, 6, \dots$). The cancellation of the remaining odd-order terms up to a given order will be provided by fulfilling a set of basic conditions for a_p , b_p , and c_p . For example, the condition $\sum_{p=1}^P a_p = \sum_{p=1}^P b_p = 1$ is required to cancel the first-order truncation uncertainties. Then the error function can be cast in the form

$$O(\Delta t^{K+1}) = O_3 \Delta t^3 + O_5 \Delta t^5 + O_7 \Delta t^7 + \dots + O_{K+1} \Delta t^{K+1}. \quad (15)$$

In order to kill higher odd-order truncation terms in Eq. (15), let us write down explicit expressions for O_3 , O_5 , and O_7 (this will be enough to derive algorithms up to the eighth order). Expanding both the sides of Eq. (13) into Taylor's series, and collecting the terms with the same powers of Δt one finds

$$O_3 = \alpha [\mathcal{A}, [\mathcal{A}, \mathcal{B}]] + \beta [\mathcal{B}, [\mathcal{A}, \mathcal{B}]], \quad (16)$$

$$O_5 = \gamma_1 [\mathcal{A}, [\mathcal{A}, [\mathcal{A}, [\mathcal{A}, \mathcal{B}]]]] + \gamma_2 [\mathcal{A}, [\mathcal{A}, [\mathcal{B}, [\mathcal{A}, \mathcal{B}]]]] + \gamma_3 [\mathcal{B}, [\mathcal{A}, [\mathcal{A}, [\mathcal{A}, \mathcal{B}]]]] + \gamma_4 [\mathcal{B}, [\mathcal{B}, [\mathcal{A}, [\mathcal{A}, \mathcal{B}]]]], \quad (17)$$

$$\begin{aligned} O_7 = & \zeta_1 [\mathcal{B}, [\mathcal{B}, [\mathcal{A}, [\mathcal{B}, [\mathcal{A}, [\mathcal{B}, \mathcal{A}]]]]]] + \zeta_2 [\mathcal{B}, [\mathcal{B}, [\mathcal{B}, [\mathcal{A}, [\mathcal{A}, [\mathcal{B}, \mathcal{A}]]]]]] + \zeta_3 [\mathcal{B}, [\mathcal{B}, [\mathcal{A}, [\mathcal{A}, [\mathcal{A}, [\mathcal{B}, \mathcal{A}]]]]]] \\ & + \zeta_4 [\mathcal{B}, [\mathcal{A}, [\mathcal{B}, [\mathcal{A}, [\mathcal{A}, [\mathcal{B}, \mathcal{A}]]]]]] + \zeta_5 [\mathcal{A}, [\mathcal{B}, [\mathcal{B}, [\mathcal{A}, [\mathcal{A}, [\mathcal{B}, \mathcal{A}]]]]]] + \zeta_6 [\mathcal{A}, [\mathcal{B}, [\mathcal{A}, [\mathcal{B}, [\mathcal{A}, [\mathcal{B}, \mathcal{A}]]]]]] \\ & + \zeta_7 [\mathcal{B}, [\mathcal{A}, [\mathcal{A}, [\mathcal{A}, [\mathcal{A}, [\mathcal{B}, \mathcal{A}]]]]]] + \zeta_8 [\mathcal{A}, [\mathcal{B}, [\mathcal{A}, [\mathcal{A}, [\mathcal{A}, [\mathcal{B}, \mathcal{A}]]]]]] + \zeta_9 [\mathcal{A}, [\mathcal{A}, [\mathcal{B}, [\mathcal{A}, [\mathcal{A}, [\mathcal{B}, \mathcal{A}]]]]]] \\ & + \zeta_{10} [\mathcal{A}, [\mathcal{A}, [\mathcal{A}, [\mathcal{A}, [\mathcal{A}, [\mathcal{B}, \mathcal{A}]]]]]]. \end{aligned} \quad (18)$$

Here we take into account the fact that the operators \mathcal{B} and \mathcal{C} commute between themselves, i.e., $[\mathcal{B}, \mathcal{C}] = 0$, so that any occurrence of constructions containing the chain $[\mathcal{B}, [\mathcal{B}, [\mathcal{A}, \mathcal{B}]]]$ has been ignored (in particular, for fifth-order

truncation term 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 α , β , γ_{1-4} , and ζ_{1-10} , arising in Eqs. (16)–(18), are functions of the coeffi-

icients a_p , b_p , and c_p , where $p=1,2,\dots,P$. The concrete form of these functions will depend on P and the version (velocity or position) under consideration.

The most simple way to obtain explicit expressions for the multipliers consists in the following. First, since we are dealing with self-adjoint schemes, the total number of single-exponential operators (stages) in Eq. (13) is actually equal to $S=2P-1$, i.e., it accepts only odd values (mention that one of the boundary set of coefficients is set to zero, $a_1=0$ or $b_p=c_p=0$). Then we can always choose a central single-exponential operator, and further consecutively applying $P-1$ times the two types of symmetric transformation,

$$e^{\mathcal{W}^{(n+1)+O^{(n+1)}}} = e^{-Aa^{(n)}\Delta t} e^{\mathcal{W}^{(n)+O^{(n)}}} e^{Aa^{(n)}\Delta t},$$

$$e^{\mathcal{W}^{(n+1)+O^{(n+1)}}} = e^{Bb^{(n)}\Delta t + Cc^{(n)}\Delta t^3} e^{\mathcal{W}^{(n)+O^{(n)}}} e^{Bb^{(n)}\Delta t + Cc^{(n)}\Delta t^3} \quad (19)$$

come to factorization (13), where

$$\mathcal{W} = (\nu A + \sigma B)\Delta t,$$

and O is defined by Eq. (15). The quantities $a^{(n)}$, $b^{(n)}$, and $c^{(n)}$ are related to a_p , b_p , and c_p , respectively (the relationship between n and p is determined below). For velocitylike decomposition with even P or positionlike at odd P , the central operator is correspondingly $e^{-Aa^{(p-2)/2+1}\Delta t}$ or $e^{Aa^{(p-1)/2+1}\Delta t}$. So that here we must put $\sigma^{(0)}=0$ as well as $\alpha^{(0)}=\beta^{(0)}=\gamma_{1-4}^{(0)}=\zeta_{1-10}^{(0)}=0$ and let either $\nu^{(0)}=a_{(p-2)/2+1}$ or $\nu^{(0)}=a_{(p-1)/2+1}$ on the very beginning ($n=0$) of the recursive procedure. The start of the procedure should be performed with the second line of Eq. (19) at $b^{(0)}=b_{(p-2)/2}$ and $c^{(0)}=c_{(p-2)/2}$ or $b^{(0)}=b_{(p-1)/2}$ and $c^{(0)}=c_{(p-1)/2}$ with further consecutive decreasing the index p by unity with increasing the number $n=1,2,\dots,P-1$ at $a^{(n)}\equiv a_p$, $b^{(n)}\equiv b_p$, and $c^{(n)}\equiv c_p$ in both the lines of transformation (19). For velocitylike decomposition with odd P or positionlike at even P , the central operator will be $e^{Bb^{(p-1)/2+1}\Delta t + Cc^{(p-1)/2+1}\Delta t^3}$ or $e^{Bb^{(p-2)/2+1}\Delta t + Cc^{(p-2)/2+1}\Delta t^3}$, corresponding to $\sigma^{(0)}=b_{(p-1)/2+1}$ and $\beta^{(0)}=c_{(p-1)/2+1}$ or $\sigma^{(0)}=b_{(p-2)/2+1}$ and $\beta^{(0)}=c_{(p-2)/2+1}$, respectively, with $\nu^{(0)}=0$ and $\alpha^{(0)}=\gamma_{1-4}^{(0)}=\zeta_{1-10}^{(0)}=0$. In this case, the procedure should be started with the first type of transformation at $a^{(0)}=a_{(p-1)/2+1}$ or $a^{(0)}=a_{(p-2)/2+1}$ with decreasing p at increasing n for $b^{(n)}\equiv b_p$, $c^{(n)}\equiv c_p$, and $a^{(n)}\equiv a_p$ in Eq. (19).

The recursive relations between the multipliers ν , σ , α , β , and γ_{1-4} corresponding to the first line of Eq. (19) are

$$\nu^{(n+1)} = \nu^{(n)} + 2a^{(n)}, \quad \sigma^{(n+1)} = \sigma^{(n)}, \quad (20)$$

$$\alpha^{(n+1)} = \alpha^{(n)} - a^{(n)}\sigma^{(n)}(a^{(n)} + \nu^{(n)})/6, \quad (21)$$

$$\beta^{(n+1)} = \beta^{(n)} - a^{(n)}\sigma^{(n)2}/6, \quad (22)$$

$$\gamma_1^{(n+1)} = \gamma_1^{(n)} + a^{(n)}(a^{(n)} + \nu^{(n)})[7(a^{(n)2} + 7a^{(n)}\nu^{(n)} + \nu^{(n)2})\sigma^{(n)} - 60\alpha^{(n)}]/360,$$

$$\gamma_2^{(n+1)} = \gamma_2^{(n)} + a^{(n)}(30\alpha^{(n)}\sigma^{(n)} - 30a^{(n)}\beta^{(n)} - 30\beta^{(n)}\nu^{(n)} + 3a^{(n)2}\sigma^{(n)2} + 2a^{(n)}\nu^{(n)}\sigma^{(n)2} + \nu^{(n)2}\sigma^{(n)2})/180, \quad (23)$$

$$\gamma_3^{(n+1)} = \gamma_3^{(n)} + a^{(n)}\sigma^{(n)}[(8a^{(n)2} + 12a^{(n)}\nu^{(n)} + \nu^{(n)2})\sigma^{(n)} - 120\alpha^{(n)}]/360,$$

$$\gamma_4^{(n+1)} = \gamma_4^{(n)} + a^{(n)}\sigma^{(n)}[(6a^{(n)} + \nu^{(n)})\sigma^{(n)2} - 60\beta^{(n)}]/180.$$

For the second type of transformation the relations read

$$\nu^{(n+1)} = \nu^{(n)}, \quad \sigma^{(n+1)} = \sigma^{(n)} + 2b^{(n)}, \quad (24)$$

$$\alpha^{(n+1)} = \alpha^{(n)} + b^{(n)}\nu^{(n)2}/6, \quad (25)$$

$$\beta^{(n+1)} = \beta^{(n)} + [12c^{(n)} + b^{(n)}\nu^{(n)}(b^{(n)} + \sigma^{(n)})]/6, \quad (26)$$

$$\gamma_1^{(n+1)} = \gamma_1^{(n)} - b^{(n)}\nu^{(n)4}/360,$$

$$\gamma_2^{(n+1)} = \gamma_2^{(n)} - \nu^{(n)}\{60\alpha^{(n)}b^{(n)} - \nu^{(n)}[30c^{(n)} - b^{(n)}\nu^{(n)}(6b^{(n)} + \sigma^{(n)})]\}/180,$$

$$\gamma_3^{(n+1)} = \gamma_3^{(n)} + b^{(n)}\nu^{(n)}[60\alpha^{(n)} + \nu^{(n)2}(4b^{(n)} - \sigma^{(n)})]/360, \quad (27)$$

$$\gamma_4^{(n+1)} = \gamma_4^{(n)} - [30\alpha^{(n)}b^{(n)}(b^{(n)} + \sigma^{(n)}) - \nu^{(n)}(30\beta^{(n)}b^{(n)} + 60b^{(n)}c^{(n)} - 3b^{(n)3}\nu^{(n)} + 30c^{(n)}\sigma^{(n)} - 2b^{(n)2}\nu^{(n)}\sigma^{(n)} - b^{(n)}\nu^{(n)}\sigma^{(n)2})]/180.$$

The relations for ζ_{1-10} are presented in the Appendix. In such a way, at the end of the recursive process (i.e., after $P-1$ steps) the multipliers can readily be obtained. The form of the first two multipliers are particularly simple and look as $\nu = \sum_{p=1}^P a_p$ and $\sigma = \sum_{p=1}^P b_p$. So that, as was already mentioned above, putting $\nu=1$ and $\sigma=1$ will cancel the first-order truncation uncertainties (because the resulting exponential propagator must behave like $e^{(A+B)\Delta t}$). Next multipliers should be set to zero and we come to the necessity of solving a system of nonlinear equations (so-called order conditions) with respect to a_p , b_p , and c_p . We shall now consider actual self-adjoint algorithms of orders $K=2, 4, 6$, and 8 .

3. Force-gradient algorithms of order two

Putting $P=2$ in Eq. (13) with $a_1=0$, $b_1=b_2=1/2$, $a_2=1$, and $c_1=c_2\equiv\xi$ leads to the following velocity-force-gradient algorithm of the second ($K=2$) order:

$$e^{(A+B)\Delta t + O(\Delta t^3)} = e^{B(\Delta t/2) + \xi C\Delta t^3} e^{A\Delta t} e^{B(\Delta t/2) + \xi C\Delta t^3}, \quad (28)$$

with $\alpha=1/12$ and $\beta=1/24+2\xi$. Note that here and below, for reducing the number of unknowns, we will always take into account in advance the symmetry of coefficients a_p , b_p , and c_p as well as the fulfilling the first-order conditions $\nu = \sum_{p=1}^P a_p = 1 = \sum_{p=1}^P b_p = \sigma$ when writing decomposition

formulas. Then solving the equation $\beta=0$ yields $\xi=-1/48$ and we come to the already found integrator (12). It is worth remarking that negative values of quantities c_p at force gradients have nothing to do with the above problem of positiveness of time coefficients arising at velocities and forces, i.e., for a_p and b_p . The reason is that the incremental velocity $b_p \mathbf{a} \Delta t + c_p \mathbf{G} \Delta t^3$ in Eq. (14) can be rewritten as $(b_p \mathbf{a} + c_p \mathbf{G} \Delta t^2) \Delta t \equiv b_p \tilde{\mathbf{a}} \Delta t$, and thus treated as the velocity changing in a modified step-size-dependent acceleration field $\tilde{\mathbf{a}} = \mathbf{a} + (c_p/b_p) \mathbf{G} \Delta t^2$.

The position counterpart of Eq. (28) is obtained from Eq. (13) at $a_1=0$, $a_1=a_2=1/2$, $b_1=1$, $b_2=0$, $c_1 \equiv \xi$, and $c_2=0$, which yields

$$e^{(A+B)\Delta t + O(\Delta t^3)} = e^{A(\Delta t/2)} e^{B\Delta t + \xi C \Delta t^3} e^{A(\Delta t/2)}, \quad (29)$$

for which $\alpha = -1/24$ and $\beta = -1/12 + \xi$. Letting $\xi = 1/12$ will minimize the third-order truncation errors to the value $\alpha[\mathcal{A}, [\mathcal{A}, \mathcal{B}]] \Delta t^3$, which is even twice smaller in magnitude than that of the velocity version. Note, however, that for both versions (28) and (29), which require one force plus one force-gradient evaluations per time step, the order of integration is not increased with respect to the usual (when $\xi=0$) Verlet integrators requiring only one force recalculation. In view of this, the applying force gradients in a particular case of $P=2$ can be justified only for strongly interacting systems when the kinetic part \mathcal{A} of the Liouville operator \mathcal{L} is much smaller than the potential part \mathcal{B} , i.e., when $\mathcal{L} = \varepsilon \mathcal{A} + \mathcal{B}$ with $\varepsilon \ll 1$. Then the remaining part $\alpha[\mathcal{A}, [\mathcal{A}, \mathcal{B}]] \Delta t^3$ of local uncertainties will behave like $\propto \varepsilon^2$ and can be neglected.

4. Force-gradient algorithms of order four

Further increasing P on unity allows us to kill exactly both the multipliers α and β , which is needed for obtaining fourth-order ($K=4$) integrators. So that choosing $P=3$ leads to the velocitylike propagation

$$e^{(A+B)\Delta t + O(\Delta t^5)} = e^{B\lambda \Delta t + \xi C \Delta t^3} e^{A(\Delta t/2)} e^{B(1-2\lambda)\Delta t + \chi C \Delta t^3} \times e^{A(\Delta t/2)} e^{B\lambda \Delta t + \xi C \Delta t^3} \quad (30)$$

following from Eq. (13) at $a_1=0$, $a_2=a_3=1/2$, $b_1=b_3=\lambda$, $b_2=1-2\lambda$, $c_1=c_3=\xi$, and $c_2=\chi$. Here relations (21), (22), (25), and (26) come to the two order conditions

$$\alpha = -\frac{1-6\lambda}{24} = 0, \quad \beta = -\frac{1}{12} + \frac{\lambda}{2} - \frac{\lambda^2}{2} + 2\xi + \chi = 0,$$

with three unknowns λ , ξ , and χ . The first unknown is immediately obtained satisfying the first condition,

$$\lambda = \frac{1}{6}. \quad (31)$$

The second equality is then reduced to $2\xi + \chi = 1/72$, resulting in a whole family of velocity-force-gradient algorithms of the fourth order. In general, such algorithms will require two force and two force-gradient recalculations per time step.

Remembering that we are interested in the derivation of most efficient integrators, three cases deserve to be considered. The two of them are aimed to reduce the number of force-gradient recalculations from two to one. This is possible by choosing either

$$\xi=0, \quad \chi = \frac{1}{72} \quad (32)$$

or

$$\chi=0, \quad \xi = \frac{1}{144}. \quad (33)$$

In the third case we will try to minimize the norm

$$\gamma = \sqrt{\gamma_1^2 + \gamma_2^2 + \gamma_3^2 + \gamma_4^2} \quad (34)$$

of fifth-order truncation errors $O(\Delta t^5)$ at $\xi \neq 0$ and $\chi = 1/72 - 2\xi \neq 0$, treating ξ as a free parameter. In view of recursive relations (23) and (27), explicit expressions for the components of $O(\Delta t^5)$ are

$$\begin{aligned} \gamma_1 &= \frac{7-30\lambda}{5760}, \quad \gamma_2 = \frac{1}{480} - \frac{\chi}{24} - \frac{\lambda^2}{24} + \frac{\xi}{6}, \\ \gamma_3 &= \frac{1}{360} - \frac{\lambda}{48} + \frac{\lambda^2}{24}, \\ \gamma_4 &= \frac{1}{120} - \frac{\lambda}{16} + \frac{7\lambda^2}{48} - \frac{\lambda^3}{8} + \frac{\xi}{6} - \frac{\chi}{2} \left(\frac{1}{3} - \lambda \right). \end{aligned}$$

Then taking into account Eq. (31) one finds the function

$$\gamma = \frac{1}{135\sqrt{2048}} \sqrt{19 + 12\,240\xi + 6\,480\,000\xi^2}$$

with the minimum $\gamma_{\min} = \sqrt{661}/43\,200 \approx 0.000\,595$ at

$$\xi = -\frac{17}{18\,000}, \quad \chi = \frac{71}{4\,500}. \quad (35)$$

At the same time, the values of γ corresponding to first two algorithms (32) and (33) constitute $\sqrt{19/2048}/135 \approx 0.000\,713$ and $7\sqrt{17}/8640 \approx 0.003\,34$, respectively.

Position version of Eq. (30) reads

$$e^{(A+B)\Delta t + O(\Delta t^5)} = e^{-A\lambda \Delta t} e^{B(\Delta t/2) + \xi C \Delta t^3} e^{-A(1-2\lambda)\Delta t} \times e^{B(\Delta t/2) + \xi C \Delta t^3} e^{-A\lambda \Delta t} \quad (36)$$

and is obtained from Eq. (13) at $P=3$ with $a_1=a_3=\lambda$, $a_2=(1-2\lambda)$, $b_1=b_2=1/2$, $c_1=c_2=\xi$, and $b_3=c_3=0$. Here, the number of unknowns coincides with the number of the order conditions,

$$\alpha = \frac{1}{12} - \frac{\lambda}{2} + \frac{\lambda^2}{2} = 0, \quad \beta = \frac{1}{24} - \frac{\lambda}{4} + 2\xi = 0,$$

the solving of which yields two solutions,

$$\lambda = \frac{1}{2} \left(1 \mp \frac{1}{\sqrt{3}} \right), \quad \xi = \frac{1}{48} (2 \mp \sqrt{3}). \quad (37)$$

Then the norm of truncation uncertainties $O(\Delta t^5)$ appearing in Eq. (36) is $\gamma = (1873 \mp 40\sqrt{2187})^{1/2}/2160$, so that the preference should be given to sign “-” in Eq. (37), because this leads to a smaller value, $\gamma_- \approx 0.000715$, of γ (whereas $\gamma_+ \approx 0.0283$). Position algorithm (36) needs, like velocity version (35), the same number of two force and force-gradient evaluations per time step.

Integrators (32) and (37) have been previously derived by Suzuki [24] based on McLachlan’s method of small perturbation [33] and referred by Chin [26] to schemes *A* and *B*, respectively. Algorithms (33) and (35) will be labeled by us as *A'* and *A''*. While scheme *A'* seems to have no advantages over the *A* integrator, algorithm *A''* corresponds to the best accuracy of the integration, because it minimizes γ . It requires, however, one extra force-gradient evaluation and, thus, can be recommended for situations when this evaluation does not present significant difficulties.

With the aim of considerably decreasing the truncation errors with a little additional computational efforts, Chin [26] has proposed to consider extended force-gradient algorithms of the fourth order. This has been achieved by increasing the number of force recalculations on unity with respect to the necessary minimum, i.e., choosing $n_f = 3$. At the same time, the number of force-gradient evaluations was fixed to its minimal value $n_g = 1$. Within our general approach, it is possible to introduce two fourth-order schemes satisfying the above requirements. The schemes are

$$e^{(A+B)\Delta t + O(\Delta t^5)} = e^{A\theta\Delta t} e^{B\lambda\Delta t} e^{A(1-2\theta)(\Delta t/2)} e^{B(1-2\lambda)\Delta t} e^{\chi C\Delta t^3} \times e^{A(1-2\theta)(\Delta t/2)} e^{B\lambda\Delta t} e^{A\theta\Delta t} \quad (38)$$

and

$$e^{(A+B)\Delta t + O(\Delta t^5)} = e^{B\lambda\Delta t + \xi C\Delta t^3} e^{A\theta\Delta t} e^{B(1-2\lambda)(\Delta t/2)} e^{A(1-2\theta)\Delta t} \times e^{B(1-2\lambda)(\Delta t/2)} e^{A\theta\Delta t} e^{B\lambda\Delta t + \xi C\Delta t^3}, \quad (39)$$

following from Eq. (13) at $P = 4$ and corresponding to positionlike and velocitylike integration, respectively. Note that further we will not present the relationship between the coefficients a_p , b_p , c_p of Eq. (13) and reduced variables [such as, for example, θ , λ , χ in Eq. (38)] in view of its evidence.

The order conditions for scheme (38) are

$$\alpha = -\frac{1}{24} + \lambda \left(\frac{1}{4} - \theta + \theta^2 \right) = 0, \\ \beta = -\frac{1}{12} + \frac{\lambda}{2} - \frac{\lambda^2}{2} - \lambda\theta(1-\lambda) + \chi = 0,$$

and solving them one obtains

$$\theta = \frac{1}{2} \pm \frac{1}{\sqrt{24\lambda}}, \quad \chi = \frac{1 \pm \sqrt{6\lambda}(1-\lambda)}{12}. \quad (40)$$

Relations (40) constitute a family of extended force-gradient position algorithms (38) of the fourth order with λ being a free parameter. Chin [26] has introduced an algorithm like Eq. (38) in somewhat another way, namely, as a symmetric product of two third-order schemes. This results only in one set of time coefficients which can be reproduced (at sign “-”) from Eq. (40) as a particular case corresponding to

$$\lambda = \frac{3}{8}, \quad \theta = \frac{1}{6}, \quad \chi = \frac{1}{192} \quad (41)$$

and has been referred to as scheme *C*.

Solution (41) may not, however, be necessarily optimal in view of the fact that it does not minimize the norm γ [see Eq. (34)] of truncation uncertainties $O(\Delta t^5)$. Indeed, the components of γ for scheme (38) are

$$\gamma_1 = -\frac{1}{1920} + \frac{1}{6912\lambda}, \quad \gamma_2 = \frac{6 \pm 5\sqrt{6\lambda}}{2880}, \\ \gamma_3 = -\frac{1}{360} \left(\frac{3}{2} \pm \frac{5}{\sqrt{96\lambda}} \pm 5\sqrt{\frac{\lambda}{24}} \right), \\ \gamma_4 = -\frac{1}{1440} (3 \pm 5\sqrt{24\lambda} + 45\lambda - 30\lambda^2),$$

where Eq. (40) has been used to express the function $\gamma(\lambda)$ in terms of one parameter λ exclusively. The global minimum of this function is $\gamma_{\min} \approx 0.000141$ and is achieved (at sign minus) at

$$\lambda = 0.2470939580390842, \quad \text{and thus}$$

$$\theta = 0.08935804763220157,$$

$$\chi = 0.006938106540706989 \quad (42)$$

(all results found numerically will be presented within sixteen significant digits for schemes up to the eighth order and within thirty two digits for order ten and higher). On the other hand, the value of γ corresponding to scheme *C* [Eq. (41)] is equal only to $\sqrt{87817}/414720 \approx 0.000715$, i.e., it is approximately five times larger than that of the optimized algorithm (42). The last algorithm we will designate as scheme *C'*.

A similar pattern is observed in the case of extended velocity-force-gradient integration (39). Previously, Chin and Chen [31] have indicated that for quantum mechanics simulations the integration of such a type is more preferable than positionlike scheme (38), because it requires a fewer number of spatial Fourier transforms. Again using the symmetric product of two third-order integrators to increase the order from three to four, they have obtained the following set:

$$\lambda = \frac{1}{8}, \quad \theta = \frac{1}{3}, \quad \xi = \frac{1}{384} \quad (43)$$

of time coefficients and referred to it as scheme D . We have realized that this set is not the only one possible and found a whole family of solutions [which includes Eq. (43)], namely,

$$\lambda = \frac{1}{12} \left(6 + \frac{1}{\theta(\theta-1)} \right), \quad \xi = -\frac{1}{288} \left(6 - \frac{1}{\theta(\theta-1)^2} \right),$$

where θ should be considered as a free parameter. The optimal solution, which minimizes the norm γ of fifth-order errors to the value $\gamma_{\min} \approx 0.000855$, is

$$\begin{aligned} \lambda &= 4.432\,204\,907\,934\,768 \times 10^{-2}, \\ \theta &= 2.409\,202\,729\,169\,543 \times 10^{-1}, \\ \xi &= 4.179\,297\,897\,540\,420 \times 10^{-3} \end{aligned} \quad (44)$$

and will be labeled as scheme D' . At the same time, the norm of errors corresponding to scheme D [Eq. (43)] is equal to $\gamma = \sqrt{237\,457/414\,720} \approx 0.00117$, i.e., it exceeds the minimum, which may decrease the precision of the calculations.

As can be ensured readily, the time coefficients arising at basic operators \mathcal{A} and \mathcal{B} under exponentials are positive for all the fourth-order force-gradient algorithms described in this subsection. Therefore, contrary to usual force Forest-Ruth-like schemes, such algorithms can simulate dynamical processes in all areas of physics and chemistry without any principal restrictions.

5. Force-gradient algorithms of order six

Beginning from $P=5$, the force-gradient factorization being written in velocity representation allows to eliminate the components of truncation uncertainties up to the sixth order ($K=6$) inclusively. In view of Eq. (13), such a representation reads

$$\begin{aligned} &e^{(\mathcal{A}+\mathcal{B})\Delta t + O(\Delta t^7)} \\ &= e^{\mathcal{B}\vartheta\Delta t + \mu\mathcal{C}\Delta t^3} e^{\mathcal{A}\theta\Delta t} e^{\mathcal{B}\lambda\Delta t + \xi\mathcal{C}\Delta t^3} e^{\mathcal{A}(1-2\theta)(\Delta t/2)} \\ &\quad \times e^{\mathcal{B}(1-2(\lambda+\vartheta))\Delta t + \chi\mathcal{C}\Delta t^3} e^{\mathcal{A}(1-2\theta)(\Delta t/2)} e^{\mathcal{B}\lambda\Delta t + \xi\mathcal{C}\Delta t^3} \\ &\quad \times e^{\mathcal{A}\theta\Delta t} e^{\mathcal{B}\vartheta\Delta t + \mu\mathcal{C}\Delta t^3}. \end{aligned} \quad (45)$$

The number of unknowns in propagation (45) is the same as the number of order conditions which now take the form

$$\alpha = \lambda \left(\frac{1}{4} - \theta + \theta^2 \right) - \frac{1}{24} (1 - 6\vartheta) = 0,$$

$$\begin{aligned} \beta &= \chi - \frac{1}{12} [1 - 24\mu - 6\lambda^2(2\theta-1) - 6\vartheta + 6\vartheta^2 \\ &\quad - 6\lambda(2\theta-1)(2\vartheta-1) - 24\xi] = 0, \end{aligned}$$

$$\gamma_1 = \frac{1}{5760} [7 - 30\lambda(2\theta-1)^2(1+4\theta-4\theta^2) - 30\vartheta] = 0,$$

$$\begin{aligned} \gamma_2 &= \frac{1}{480} [1 - 20\chi + 80\mu - 20\lambda^2(1-8\theta+18\theta^2-12\theta^3) \\ &\quad - 20\vartheta^2 + 20\lambda(2\theta-1)(\theta+2\vartheta-6\theta\vartheta) + 80\xi \\ &\quad - 480\theta\xi + 480\theta^2\xi] = 0, \end{aligned} \quad (46)$$

$$\begin{aligned} \gamma_3 &= \frac{1}{720} \{2 - 30\lambda^2(2\theta-1)^3 - 15\vartheta + 30\vartheta^2 - 15\lambda(2\theta-1)^2 \\ &\quad \times [1 - 4\vartheta - \theta(4\vartheta-2)]\} = 0, \end{aligned}$$

$$\begin{aligned} \gamma_4 &= \frac{1}{240} \{2 + 40\mu - 30\lambda^3(2\theta-1)^2 - 40\chi[1 + \lambda(6\theta-3) \\ &\quad - 3\vartheta] - 15\vartheta + 35\vartheta^2 - 30\vartheta^3 - 5\lambda^2(2\theta-1)[7 - 18\vartheta \\ &\quad + 6\theta(2\vartheta-1)] + 5\lambda(2\theta-1)[3 - 14\vartheta + 18\vartheta^2 + 2\theta(1 \\ &\quad - 6\vartheta + 6\vartheta^2)] + 40\xi - 240\theta\xi + 480\theta\vartheta\xi\} = 0. \end{aligned}$$

The unique real solution to system (46) is

$$\theta = \frac{1}{2} + \frac{\sqrt[3]{[675+75\sqrt{6}]}}{30} + \frac{5}{2\sqrt[3]{[675+75\sqrt{6}]}}}, \quad \vartheta = \frac{\theta}{3},$$

$$\lambda = -\frac{5\theta}{3}(\theta-1), \quad \xi = -\frac{5\theta^2}{144} + \frac{\theta}{36} - \frac{1}{288},$$

$$\chi = \frac{1}{144} - \frac{\theta}{36} \left(\frac{\theta}{2} + 1 \right), \quad \mu = 0. \quad (47)$$

Solution (47) constitutes a velocity-force-gradient algorithm of the sixth order with four force and three (since $\mu=0$) force-gradient evaluations per time step, i.e., with $n_f=4$ and $m_g=3$. Its advantage over usual sixth-order integrators consists in the fact that it is composed of a considerably smaller number, namely, $S=2P-1=9$, instead of 15, of single exponential operators. The norm

$$\zeta = \sqrt{\sum_{k=1}^{10} \zeta_k^2} \quad (48)$$

of seventh-order truncation errors $O(\Delta t^7)$ [see Eq. (45)], corresponding to solution (47), is equal to $\zeta \approx 0.00150$. Note also that the position version of decomposition (45) does not exist at $P=5$, because then the number of unknowns is less than the number of order equations, resulting in the absence of solutions.

As has been shown in the preceding subsection for the case of fourth-order integration, algorithms with minimal numbers n_f of force evaluation may not lead to optimal solutions. The reason is that slightly increasing n_f may significantly decrease the local errors and thus overcompensate the increased computational efforts. So that increasing n_f as well

as P on unity (note that $n_f = P - 1$) and not changing n_g , i.e., choosing $P = 6$ with $n_f = 5$ and $n_g = 3$, it is possible to derive from decomposition (13) up four (two velocitylike and two positionlike) extended sixth-order schemes. They are

$$\begin{aligned} &CACABABACAC, \quad CABACACABAC, \\ &ABACACACABA, \quad ACABACABACA, \end{aligned} \quad (49)$$

where A and B denote exponential operators $e^{Aa_p \Delta t}$ and $e^{Bb_p \Delta t}$, respectively, whereas letter C corresponds to $e^{Bb_p \Delta t + Cc_p \Delta t^3}$. Each of these extended schemes has itself correspondingly six, eight, four, and two sets of real solutions for time coefficients. We have realized that the smallest values of the norm ζ [see Eq. (48)] of local errors $O(\Delta t^7)$ within the sets are 0.000 026 4, 0.000 014 7, 0.000 146, and 0.000 006 07, respectively. So that the last scheme should be considered as the best. The more explicit form for it is

$$\begin{aligned} &e^{(A+B)\Delta t + O(\Delta t^7)} \\ &= e^{A\rho \Delta t} e^{B\vartheta \Delta t + \mu C \Delta t^3} e^{A\theta \Delta t} e^{B\lambda \Delta t} e^{A[1-2(\theta+\rho)](\Delta t/2)} \\ &\quad \times e^{B[1-2(\lambda+\vartheta)]\Delta t + \chi C \Delta t^3} e^{A[1-2(\theta+\rho)](\Delta t/2)} e^{B\lambda \Delta t} \\ &\quad \times e^{A\theta \Delta t} e^{B\vartheta \Delta t + \mu C \Delta t^3} e^{A\rho \Delta t}, \end{aligned} \quad (50)$$

with the optimal solution

$$\begin{aligned} \rho &= 0.109\,705\,972\,394\,868\,2, \\ \theta &= 0.414\,063\,226\,731\,083\,1, \\ \vartheta &= 0.269\,331\,584\,893\,530\,1, \\ \lambda &= 1.131\,980\,348\,651\,556, \\ \chi &= -0.013\,246\,386\,434\,160\,52, \\ \mu &= 0.000\,864\,216\,133\,970\,616\,6, \end{aligned} \quad (51)$$

corresponding to $\zeta = 0.000\,006\,07$. In such a way, the error function has been reduced more than in 200 times with respect to scheme (47) for which $\zeta \approx 0.001\,50$.

6. Force-gradient algorithms of order eight

In the case when $K = 8$ we must satisfy up to 18 order conditions, namely, $\nu = 1$, $\sigma = 1$, $\alpha = 0$, $\beta = 0$, $\gamma_{1-4} = 0$, and $\zeta_{1-10} = 0$. Taking into account the symmetry of time coefficients a_p , b_p , and c_p , this can be achieved at least at $P = 12$, i.e., using $S = 2P - 1 = 23$ single exponential operators. For $P = 12$ the velocitylike and positionlike force-gradient decomposition (13) transforms into the schemes

$$CACACACACACACACACACAC \quad (52)$$

and

$$ACACACACACACACACACACA, \quad (53)$$

respectively. The number of unknowns for both the schemes is also equal to 18 and we can try to solve the system of order conditions with respect to these unknowns.

It is worth remarking that such a system appears to be very cumbersome for schemes under consideration. For instance, the resulting nonlinear equations of this system being written explicitly in MATHEMATICA create a file of size 0.5 Mb. In view of this, our attempts to solve the equations symbolically have not met with much success. We mention that all the results presented above for algorithms of orders 2, 4, and 6 have been solved analytically or in quadratures. Saying in quadratures we mean that the problem was reduced to finding real zeros for a one-dimensional polynomial of a given order, so that we could identify exactly the number of solutions and their locations. Here the situation is somewhat different because we must solve the system using purely numerical approaches, such as the Newton method. As a result, one cannot guarantee that we will find all possible solutions. However, solving the system on a computer during a significantly long period of time, one can say with a great probability that we have found almost all physically interesting solutions and chosen among them nearly optimal sets.

The numerical calculations have been performed in FORTRAN using the well-recognized Newton solver with numerical determination of partial derivatives. The values for nonlinear functions (that constitute the system of equations) were obtained using recursive relations (20)–(27), (A1), and (A2), but not explicit expressions for them to save the processor time and increase the precision of the computations. The initial guesses for solutions were generated at random within the interval $[-2.5, 2.5]$ in each of the eighteen directions. If Newton's iterations began to diverge at a particular guess or during the calculations, a next random point was involved to repeat the process. In such a way, after several days of continuously attacking the systems of equations on an Origin 3800 workstation, we found two and five solutions for schemes (52) and (53), respectively. The optimal among them are the following:

$$\begin{aligned} a_1 &= 0, \\ b_1 &= b_{12} = 1.839\,699\,354\,244\,402 \times 10^{-1}, \\ c_1 &= c_{12} = 0, \\ a_2 &= a_{12} = 6.922\,517\,172\,738\,832 \times 10^{-1}, \\ b_2 &= b_{11} = 7.084\,389\,757\,230\,299 \times 10^{-1}, \\ c_2 &= c_{11} = 3.976\,209\,968\,238\,716 \times 10^{-2}, \\ a_3 &= a_{11} = -3.183\,450\,347\,119\,991 \times 10^{-1}, \\ b_3 &= b_{10} = 1.981\,440\,445\,033\,534 \times 10^{-1}, \\ c_3 &= c_{10} = 2.245\,403\,440\,322\,733 \times 10^{-2}, \\ a_4 &= a_{10} = 6.766\,724\,088\,765\,565 \times 10^{-1}, \\ b_4 &= b_9 = -6.409\,380\,745\,116\,974 \times 10^{-2}, \end{aligned}$$

$$\begin{aligned}
c_4 &= c_9 = 9.405\,266\,232\,181\,224 \times 10^{-4}, \\
a_5 &= a_9 = -7.207\,972\,470\,858\,706 \times 10^{-1}, \\
b_5 &= b_8 = -6.887\,429\,532\,761\,409 \times 10^{-1}, \\
c_5 &= c_8 = -7.336\,500\,519\,635\,302 \times 10^{-2}, \\
a_6 &= a_8 = 3.580\,316\,862\,350\,045 \times 10^{-1}, \\
b_6 &= b_7 = 1.622\,838\,050\,764\,871 \times 10^{-1}, \\
c_6 &= c_7 = 2.225\,664\,796\,363\,730 \times 10^{-2}, \\
a_7 &= -3.756\,270\,611\,751\,488 \times 10^{-1},
\end{aligned}$$

for velocitylike integration (52), and

$$\begin{aligned}
b_{12} &= c_{12} = 0, \\
a_1 &= a_{12} = 4.100\,967\,473\,880\,111\,192\,878\,469\,300\,508\,0 \\
&\quad \times 10^{-1}, \\
b_1 &= b_{11} = 4.824\,930\,981\,741\,495\,291\,269\,584\,266\,478\,5 \\
&\quad \times 10^{-3}, \\
c_1 &= c_{11} = 1.474\,393\,690\,779\,752\,836\,471\,724\,476\,073\,6 \\
&\quad \times 10^{-4}, \\
a_2 &= a_{11} = -3.412\,334\,575\,605\,278\,048\,910\,169\,737\,849\,9 \\
&\quad \times 10^{-1}, \\
b_2 &= b_{10} = 1.749\,239\,486\,109\,037\,560\,341\,900\,137\,420\,7 \\
&\quad \times 10^{-1}, \\
c_2 &= c_{10} = 2.328\,845\,053\,193\,254\,535\,719\,496\,760\,015\,5 \\
&\quad \times 10^{-4}, \\
a_3 &= a_{10} = 2.564\,471\,402\,106\,815\,049\,236\,176\,163\,174\,3 \\
&\quad \times 10^{-1}, \\
b_3 &= b_9 = 2.930\,436\,637\,095\,706\,616\,436\,454\,620\,428\,8 \\
&\quad \times 10^{-1}, \\
c_3 &= c_9 = 6.164\,865\,963\,553\,596\,249\,770\,561\,988\,475\,2 \\
&\quad \times 10^{-3}, \\
a_4 &= a_9 = 2.776\,527\,397\,581\,243\,839\,410\,047\,624\,264\,1 \\
&\quad \times 10^{-1}, \\
b_4 &= b_8 = 4.744\,894\,016\,845\,977\,028\,423\,813\,648\,251\,1 \\
&\quad \times 10^{-2},
\end{aligned}$$

$$\begin{aligned}
c_4 &= c_8 = -1.230\,751\,686\,083\,124\,071\,673\,201\,696\,003\,4 \\
&\quad \times 10^{-2}, \\
a_5 &= a_8 = -5.692\,626\,686\,975\,377\,390\,293\,965\,732\,115\,9 \\
&\quad \times 10^{-1}, \\
b_5 &= b_7 = -1.529\,986\,341\,174\,397\,449\,921\,965\,232\,047\,7 \\
&\quad \times 10^{-3}, \\
c_5 &= c_7 = -7.329\,664\,855\,912\,638\,538\,701\,716\,164\,379\,8 \\
&\quad \times 10^{-5}, \\
a_6 &= a_7 = 4.662\,994\,989\,012\,485\,357\,679\,442\,382\,019\,4 \\
&\quad \times 10^{-1}, \\
b_6 &= -3.742\,299\,425\,900\,257\,160\,684\,246\,260\,379\,1 \\
&\quad \times 10^{-2}, \\
c_6 &= 1.529\,586\,099\,452\,374\,473\,199\,329\,384\,700\,1 \\
&\quad \times 10^{-2},
\end{aligned}$$

for its positionlike counterpart (53). The number of force evaluations per times step for schemes (52) and (53) is $n_f = P - 1 = 11$, whereas the number of force-gradient recalculations consists $n_g = 10$ [since $c_1 = 0$ and thus the two boundary letters C in formula (52) should be actually replaced by B] and $n_g = 11$, respectively.

In view of a complicated structure of the ninth-order truncation uncertainties $O(\Delta t^9)$, the optimal solutions just presented have been chosen in a somewhat different way than above, namely, by providing a minimum for the function $\delta = \max_{p=1}^P (|a_p|, |b_p|)$. This simplified criterion was used, in particular, by Kahan and Li [6,22], when optimizing usual force algorithms. As a result, we have obtained $\delta_{\min} = |a_5| = |a_9| \approx 0.721$ for scheme (52) and $\delta_{\min} = |a_5| = |a_8| \approx 0.569$ for scheme (53). Since δ_{\min} is smaller in the last case, the positionlike integration should be considered as more preferable. Its time coefficients have been presented even with thirty two significant digits to be used in applications for very accurate integration. In order to ensure that all the digits shown are correct in both the cases, we have carried out a few additional Newton's iterations in MAPLE with up 200 digits during the internal computations, and taking as initial guesses the solutions already obtained in FORTRAN.

The positionlike decomposition (53) has another advantage over the velocity version (52) in that all the intermediate ($q \leq P$) states in velocity and position space stay during the integration within a given interval $[0, \Delta t]$, i.e., $0 \leq \sum_{p=1}^q a_p \leq 1$ and $0 \leq \sum_{p=1}^q b_p \leq 1$. This property may be important when solving ordinary differential equations (for specific physical systems or in pure mathematics applications) with singularities beyond the interval of the integration (note that, in particular, any system of differential equations of the form $d^2 \mathbf{x} / dt^2 = \mathbf{f}(\mathbf{x})$ is reduced to the equations of motion under consideration in this paper). Note also that in order to construct eighth-order schemes within usual decomposition approach (6) (i.e., without involving any force gradients), it could be necessary to apply up to $2 \times 18 - 1 = 35$ (instead of

TABLE I. The basic decomposition force-gradient algorithms.

Algorithm	Order	n_f	n_g	Err3	Err5	Err7	Equations	Remarks	Label
<i>CAC</i>	2	1	1	8.33×10^{-2}	1.34×10^{-2}	2.24×10^{-3}	(28)	This work	$G2'$
<i>ACA</i>	2	1	1	4.17×10^{-2}	6.48×10^{-3}	7.25×10^{-4}	(29)	This work ^a	$G2$
<i>BACAB</i>	4	2	1	0	7.13×10^{-4}	6.30×10^{-5}	(30,32)	Refs. [24,26]	A
<i>CABAC</i>	4	2	1	0	3.34×10^{-3}	2.72×10^{-4}	(30,33)	This work	A'
<i>CACAC</i>	4	2	2	0	5.95×10^{-4}	4.83×10^{-5}	(30,35)	This work	A''
<i>ACACA</i>	4	2	2	0	7.15×10^{-4}	5.59×10^{-5}	(36,37)	Refs. [24,26]	B
<i>ABACABA</i>	4	3	1	0	1.41×10^{-4} ^b	1.04×10^{-5} ^b	(38),(41)/(42)	Ref. [26]/This work ^a	C/C'
<i>CABABAC</i>	4	3	1	0	8.55×10^{-4} ^c	2.24×10^{-5} ^c	(39),(43)/(44)	Ref. [31]/This work	D/D'
<i>BACACACAB</i>	6	4	3	0	0	1.50×10^{-3}	(45),(47)	This work	$G6'$
<i>CACABABACAC</i>	6	5	3	0	0	2.64×10^{-5}	(49)	This work	$G6''$
<i>CABACACABAC</i>	6	5	3	0	0	1.47×10^{-5}	(49)	This work	$G6'''$
<i>ABACACACABA</i>	6	5	3	0	0	1.46×10^{-4}	(49)	This work	$G6''''$
<i>ACABACABACA</i>	6	5	3	0	0	6.07×10^{-6}	(50,51)	This work ^a	$G6$
<i>BACACACACACA</i>									
<i>CACACACACAB</i>	8	11	10	0	0	0	(52)	This work	$G8'$
<i>ACACACACACAC</i>									
<i>ACACACACACA</i>	8	11	11	0	0	0	(53)	This work ^a	$G8$

^aThe best algorithm within a given order.

^bThe value corresponding to scheme C' .

^cThe value corresponding to scheme D' .

23) single exponential propagators. Such schemes have never been derived by decomposition (6) because of the serious technical difficulties. They can be explicitly introduced only by compositions of lower-order integrators (see the following section). Instead, using generalized scheme (13) has allowed us to derive eighth-order algorithms by direct decompositions (the force-gradient algorithms have been presented in Sec. II B 5 for order six).

All the decomposition algorithms obtained by us in Secs. II B 3, II B 4, II B 5, and II B 6 are collected in Table I. Here, the designations Err3, Err5, and Err7 relate to the norms $\sqrt{\alpha^2 + \beta^2}$, γ , and ζ of correspondingly third-, fifth-, and seventh-order truncation errors [see Eqs. (6), (15)–(18), (34), and (48)], whereas n_f and n_g denote the numbers of force and force-gradient evaluations per time step. The optimal algorithms for orders 2, 4, 6, and 8 are labeled by $G2$, C' , $G6$, and $G8$, respectively. Among other schemes presented for each given order, such algorithms reduce the truncation uncertainties to a minimum. Taking into account that this reduction is achieved at the same or nearly the same computational efforts, the optimal algorithms should be considered as the best not only with respect to their precision but in view of the overall efficiency as well (see also comments on this in Sec. III).

Finally, it is worth remarking that the problem of constructing algorithms with only positive coefficients a_p and b_p for orders six and higher still remains. We mention that for order four, this problem has been resolved (see Sec. II B 4) by transferring the force-gradient component of truncation uncertainties into the exponential propagators. For orders $K \geq 6$, additional higher-order gradients should appear under these exponentials to provide the required positiveness. Our analysis has shown, however, that such high-order exponen-

tials (besides their very cumbersome forms) cannot be evaluated in quadratures and need performing implicit calculations by iteration. In view of this we can come to a conclusion that beyond fourth order, analytically integrable decomposition algorithms with purely positive coefficients do not exist. Mathematically rigorous proof of this statement will be considered in our further investigation and presented elsewhere.

C. Integration by advanced compositions

With increasing the order of integration to ten and higher, the construction of algorithms by direct decompositions (13) becomes inefficient because of a large number of order conditions and time coefficients. However, having the already derived force-gradient integrators of lower orders K , we can try to compose them as

$$S_Q(\Delta t) = S_K(d_1 \Delta t) \cdots S_K(d_P \Delta t) \cdots S_K(d_1 \Delta t) \quad (54)$$

for obtaining an algorithm of order $Q > K$. Then the composition constants d_p , where $p = 1, 2, \dots, P$, should be chosen in such a way so as to provide the maximal possible value of Q at a given number $P \geq 2$. Note that lower-order propagations $S_K(d_p \Delta t)$ enter symmetrically in composition (54) and their total number $2P - 1$ accepts odd values. So that if a basic integrator S_K is self-adjoint, the resulting algorithm S_Q will be self-adjoint as well. The idea of using a formula like (54) is not new and has been applied by different authors in previous investigations [1,3,5,6,20–23]. But these investigations were focused, in fact, on the compositions of usual second-order ($K = 2$) schemes (to our knowledge, no actual calculations of composition constants for fourth- and higher-order-based integrators have been reported). Although using

the second-order-based approach allowed to introduce algorithms to the tenth order [5,6,22], further increasing Q has led to unresolved numerical difficulties when finding the coefficients of the compositions.

Usually, these difficulties are obviated with the help of Creutz and Gocksch's method [30]. We mention that according to this method, an algorithm of order $K+2$ can be derived by triplet concatenation,

$$S_{K+2}(\Delta t) = S_K(D_K \Delta t) S_K[(1-2D_K)\Delta t] S_K(D_K \Delta t), \quad (55)$$

of a self-adjoint integrator of order K , where $D_K = 1/(2 - 2^{1/(K+1)})$. In particular, Chin and Kidwell [29] starting from force-gradient algorithm (41) of order four and repeating procedure (55) up to order 12, have indicated a visible increase in the efficiency of the computation with respect to second-order-based schemes. In this approach, however, the number of force and force-gradient evaluations (the most time-consuming part of the calculations) increases too rapidly with increasing K , namely, as $3^{(K-4)/2}$ relatively to the fourth-order integrator.

The present study is aimed to overcome the above problems by an explicit consideration of four-, sixth-, and eighth-order-based (force-gradient) algorithms within general composition approach (54). This results in a reduction in the total number of basic propagations to a minimum and significantly speeds up the integration. The composition algorithms are derived up to the sixteenth order inclusively.

1. Fourth-order based algorithms

In the case when $K=4$, the basic self-adjoint propagation is

$$S_4(\tau) = e^{\mathcal{X}_1 \tau + \mathcal{X}_5 \tau^5 + \mathcal{X}_7 \tau^7 + \mathcal{X}_9 \tau^9 + \mathcal{X}_{11} \tau^{11} + \dots}, \quad (56)$$

where $\mathcal{X}_1 \equiv \mathcal{A} + \mathcal{B}$. Explicit forms of higher-order truncation operators \mathcal{X}_5 , \mathcal{X}_7 , \mathcal{X}_9 , \mathcal{X}_{11} , and so on [which was previously found for \mathcal{X}_5 and \mathcal{X}_7 , see Eqs. (17) and (18)] are not important within the composition approach. Then formula (54) reduces to series ($n=0, 1, \dots, P-2$) of the transformation

$$S_Q^{(n+1)}(\Delta t) = S_4(d^{(n)} \Delta t) S_Q^{(n)}(\Delta t) S_4(d^{(n)} \Delta t), \quad (57)$$

with $S_Q^{(0)} = S_4(d_P \Delta t)$ and $d^{(n)} = d_{P-n-1}$. In view of Eqs. (56) and (57), the structure of resulting propagation can be cast at each n as

$$S_Q(\Delta t) = e^{\mathcal{Y}_1 \Delta t + \mathcal{Y}_3 \Delta t^3 + \mathcal{Y}_5 \Delta t^5 + \mathcal{Y}_7 \Delta t^7 + \mathcal{Y}_9 \Delta t^9 + \mathcal{Y}_{11} \Delta t^{11} + O(\Delta t^{13})}, \quad (58)$$

with

$$\begin{aligned} \mathcal{Y}_1 &= q_1 \mathcal{X}_1, & \mathcal{Y}_5 &= q_2 \mathcal{X}_5, & \mathcal{Y}_7 &= q_3 \mathcal{X}_7 + q_4 [\mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_5], \\ \mathcal{Y}_9 &= q_5 \mathcal{X}_9 + q_6 [\mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_7] + q_7 [\mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_5], \end{aligned} \quad (59)$$

$$\begin{aligned} \mathcal{Y}_{11} &= q_8 \mathcal{X}_{11} + q_9 [\mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_9] + q_{10} [\mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_7] \\ &+ q_{11} [\mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_5] + q_{12} [\mathcal{X}_5, \mathcal{X}_1, \mathcal{X}_5]. \end{aligned}$$

Comparing Eqs. (56) and (58) yields values of q multipliers at $n=0$, namely, $q_1^{(0)} = d_P$, $q_2^{(0)} = d_P^{K+1}$, $q_3^{(0)} = d_P^{K+3}$, $q_5^{(0)} = d_P^{K+5}$, and $q_8^{(0)} = d_P^{K+7}$, whereas, $q_4^{(0)} = q_6^{(0)} = q_7^{(0)} = q_9^{(0)} = q_{10}^{(0)} = q_{11}^{(0)} = q_{12}^{(0)} = 0$. Expanding both the sides of Eq. (57) into Taylor's series with respect to Δt , one finds that values for these multipliers at $n>0$ can be obtained using the following recursive relations:

$$\begin{aligned} q_1^{(n+1)} &= q_1^{(n)} + 2d^{(n)}, & q_2^{(n+1)} &= q_2^{(n)} + 2d^{(n)K+1}, \\ q_3^{(n+1)} &= q_3^{(n)} + 2d^{(n)K+3}, & q_4^{(n+1)} &= q_4^{(n)} + d^{(n)}(q_1^{(n)} + d^{(n)}) \\ &\times (q_1^{(n)} d^{(n)K} - q_2^{(n)})/6, \\ q_5^{(n+1)} &= q_5^{(n)} + 2d^{(n)K+5}, & q_6^{(n+1)} &= q_6^{(n)} + d^{(n)}(q_1^{(n)} + d^{(n)}) \\ &\times (q_1^{(n)} d^{(n)K+2} - q_3^{(n)})/6, \\ q_7^{(n+1)} &= q_7^{(n)} + d^{(n)}(q_1^{(n)} + d^{(n)})(q_1^{(n)2} q_2^{(n)} - 60q_4^{(n)} \\ &+ 7q_1^{(n)} q_2^{(n)} d^{(n)} + 7q_2^{(n)} d^{(n)2} - q_1^{(n)3} d^{(n)K} \\ &- 7q_1^{(n)2} d^{(n)K+1} - 7q_1^{(n)} d^{(n)K+2})/360, \\ q_8^{(n+1)} &= q_8^{(n)} + 2d^{(n)K+7}, & q_9^{(n+1)} &= q_9^{(n)} + d^{(n)}(q_1^{(n)} + d^{(n)}) \\ &\times (q_1^{(n)} d^{(n)K+4} - q_5^{(n)})/6, \quad (60) \\ q_{10}^{(n+1)} &= q_{10}^{(n)} + d^{(n)}(q_1^{(n)} + d^{(n)})(q_1^{(n)2} q_3^{(n)} - 60q_6^{(n)} \\ &+ 7q_1^{(n)} q_3^{(n)} d^{(n)} + 7q_3^{(n)} d^{(n)2} - q_1^{(n)3} d^{(n)K+2} \\ &- 7q_1^{(n)2} d^{(n)K+3} - 7q_1^{(n)} d^{(n)K+4})/360, \\ q_{11}^{(n+1)} &= q_{11}^{(n)} + d^{(n)}(q_1^{(n)} + d^{(n)})(42q_1^{(n)2} q_4^{(n)} - q_1^{(n)4} q_2^{(n)} \\ &- 2520q_7^{(n)} - 11q_1^{(n)3} q_2^{(n)} d^{(n)} + 294q_1^{(n)} q_4^{(n)} d^{(n)} \\ &- 42q_1^{(n)2} q_2^{(n)} d^{(n)2} + 294q_4^{(n)} d^{(n)2} - 62q_1^{(n)} q_2^{(n)} d^{(n)3} \\ &+ q_1^{(n)5} d^{(n)K} - 31q_2^{(n)} d^{(n)4} + 11q_1^{(n)4} d^{(n)K+1} \\ &+ 42q_1^{(n)3} d^{(n)K+2} + 62q_1^{(n)2} d^{(n)K+3} \\ &+ 31q_1^{(n)} d^{(n)K+4})/15120, \\ q_{12}^{(n+1)} &= q_{12}^{(n)} + d^{(n)}(q_1^{(n)} d^{(n)4} - q_2^{(n)})(q_2^{(n)} + d^{(n)5})/6. \end{aligned}$$

Applying the above relations $P-1$ times will give the final values of q multipliers and thus lead to the desired order conditions. For instance, the first condition is very simple and reads $q_1 = d_P + 2\sum_{p=1}^{P-1} d_p = 1$. This provides $\mathcal{Y}_1 = \mathcal{X}_1$ and guarantees [see Eqs. (56), (58), and (59)] that the order of the composition scheme will be at least not lower than that of the basic scheme, i.e., $Q \geq 4$ in our case. All other multipliers $q_2, q_3, q_4, \dots, q_N$ should be consecutively set to zero, forming higher-order conditions. The total number \mathcal{N} of the conditions depends on a required order $Q > 4$ of the compo-

sition scheme. In particular, at $Q=6$ we must kill the term \mathcal{Y}_5 at fifth-order truncation uncertainties [see Eq. (58)]. Taking into account Eq. (59), this results in two order conditions, namely, $q_1=1$ and $q_2=0$, which can be satisfied at $P=2$. Then one obtains a system of equations, $q_1=2d_1+d_2=0$, and $q_1=2d_1^5+d_2^5=0$, with respect to two unknowns, d_1 and d_2 . The system can be solved analytically, and the solution is $d_1=1/(2-2^{1/5})\equiv D_4$ with $d_2=1-2d_1$ that coincides (at $K=4$) with the result of triplet construction (55). This coinciding is not surprising because, as can be seen easily, both approaches (54) and (55) are identical in a partial case when $P=2$ and $Q-K=2$.

With further increasing Q , composition approach (54) will lead to a more efficient integration. Indeed, choosing $Q=8$ requires that the term \mathcal{Y}_7 in Eq. (58) should be killed additionally. This is achieved by putting $q_3=q_4=0$ in Eq. (59), and, therefore, by solving at $P=4$ a system of four nonlinear equations, $q_1=1$, $q_2=0$, $q_3=0$, and $q_4=0$ with respect to the same number of unknowns d_1 , d_2 , d_3 , and d_4 . So that the minimal number of fourth-order integrators in the eighth-order composition should be $2P-1=7$, whereas this number is equal to $3^{(Q-K)/2}=9$ when triplet concatenation (55) is used. Expressions for the non-linear equations can readily be reproduced by applying the corresponding set of recursive relations (60). We will not present such expressions explicitly, because as has been realized, the order equations do not allow to be solved analytically at $Q-K\geq 4$ for any $K\geq 4$. But, these equations can be solved in a quite efficient way numerically using the Newton's method. Details of the numerical calculations are similar to those described in Sec. II B 6. Here (when $P=4$, $K=4$, and $Q=8$) we have found five solutions, and it seems that no other real solutions exist. The optimal set is

$$\begin{aligned} d_1 &= 0.846\,121\,147\,469\,675\,7, \\ d_2 &= 0.158\,012\,845\,800\,856\,7, \\ d_3 &= -1.090\,206\,660\,543\,938, \\ d_4 &= 1.172\,145\,334\,546\,811. \end{aligned} \quad (61)$$

Solution (61) simultaneously leads to the smallest values for the maximal composition coefficient $\max_{p=1}^4 |d_p| \approx 1.172$ and the norm $(q_5^2 + q_6^2 + q_7^2)^{1/2} \approx 0.270$ of the main ninth-order term $\mathcal{X}(\Delta t^9)$ of truncations uncertainties.

When deriving tenth-order composition algorithms (at $K=4$), i.e., when $Q=10$, three additional order conditions arise, $q_5=0$, $q_6=0$, and $q_7=0$, needed to eliminate the term $\mathcal{X}(\Delta t^9)$ [see Eqs. (58) and (59)]. Then we come in overall to seven nonlinear equations that can be satisfied by appropriately choosing composition constants d_p ($p=1, 2, \dots, P$) at $P=7$. In this case, we have identified more than 150 real solutions and probably there are some others (we stopped the search after several days of computation). Among the solutions found, the optimal set looks

$$\begin{aligned} d_1 &= 0.805\,239\,957\,695\,780\,823\,266\,281\,698\,027\,82, \\ d_2 &= -0.491\,931\,059\,146\,231\,010\,223\,881\,388\,641\,43, \\ d_3 &= 0.354\,492\,586\,543\,984\,605\,355\,292\,699\,884\,83, \\ d_4 &= -0.695\,739\,222\,711\,402\,238\,030\,364\,634\,619\,97, \\ d_5 &= 0.399\,595\,380\,303\,292\,563\,593\,499\,770\,878\,19, \\ d_6 &= 0.549\,795\,686\,014\,384\,527\,941\,280\,315\,637\,60, \end{aligned} \quad (62)$$

and $d_7=1-2(d_1+d_2+d_3+d_4+d_5+d_6)$. This set minimizes at once the norm $(q_8^2 + q_9^2 + q_{10}^2 + q_{11}^2 + q_{12}^2)^{1/2}$ of the main eleventh-order term $\mathcal{X}(\Delta t^{11})$ of truncation errors and the quantity $\max_{p=1}^7 |d_p|$ to the values 0.00412 and 0.843 ($\equiv |d_7|$), respectively. Here, the number of basic propagations (stages) is $2P-1=13$, i.e., more than two times smaller than this number $3^{(Q-K)/2}=27$ within triplet concatenation (55).

In order to introduce twelfth-order algorithms, $Q=12$, on the basis of fourth-order compositions it is necessary to deal with $P=12$ unknowns d_p to fulfill the same number of the order conditions $q_1=1$, and $q_{2-12}=0$. Here we have found more than 200 real solutions and perhaps there are some more else. The best among them, which minimizes $\max_{p=1}^{12} |d_p|$ to the value 0.611 ($\equiv |d_{12}|$), is

$$\begin{aligned} d_1 &= 0.173\,850\,160\,930\,978\,554\,360\,617\,128\,583\,03, \\ d_2 &= 0.533\,774\,798\,907\,122\,079\,492\,826\,539\,908\,42, \\ d_3 &= 0.121\,301\,386\,146\,683\,076\,738\,022\,919\,664\,95, \\ d_4 &= 0.296\,507\,470\,338\,071\,952\,734\,400\,325\,056\,29, \\ d_5 &= -0.599\,659\,998\,573\,354\,540\,184\,823\,120\,082\,33, \\ d_6 &= 0.090\,435\,812\,862\,044\,371\,458\,711\,304\,290\,94, \\ d_7 &= -0.439\,791\,462\,576\,358\,068\,867\,787\,481\,389\,62, \\ d_8 &= -0.302\,515\,529\,223\,464\,950\,570\,102\,407\,791\,04, \\ d_9 &= 0.598\,958\,729\,892\,479\,821\,145\,459\,069\,537\,12, \\ d_{10} &= 0.312\,364\,165\,382\,755\,761\,518\,162\,807\,766\,96, \\ d_{11} &= -0.590\,812\,307\,696\,478\,331\,840\,904\,434\,453\,03, \end{aligned} \quad (63)$$

with $d_{12}=1-2\sum_{p=1}^{11} d_p$. Thus, the minimum number of fourth-order stages needed to compose the twelfth-order algorithm is $2P-1=23$, instead of up to $3^{(Q-K)/2}=81$ as in the case of usual triplet construction (55).

2. Sixth- and eighth-order based algorithms

When $K=6$ or 8 , the basic propagation reads

$$S_6(\tau) = e^{\mathcal{X}_1\tau + \mathcal{X}_7\tau^7 + \mathcal{X}_9\tau^9 + \mathcal{X}_{11}\tau^{11} + \mathcal{X}_{13}\tau^{13} + \dots} \quad (64)$$

or

$$S_8(\tau) = e^{\mathcal{X}_1\tau + \mathcal{X}_9\tau^9 + \mathcal{X}_{11}\tau^{11} + \mathcal{X}_{13}\tau^{13} + \mathcal{X}_{15}\tau^{15} + \dots}, \quad (65)$$

respectively. Here, the compositions reduce to the recursive transformation

$$S_Q^{(n+1)}(\Delta t) = S_{6,8}(d^{(n)}\Delta t)S_Q^{(n)}(\Delta t)S_{6,8}(d^{(n)}\Delta t), \quad (66)$$

with $S_Q^{(0)}$ being equal to $S_6(d_P\Delta t)$ or $S_8(d_P\Delta t)$ and $n = 0, 1, \dots, P-2$. The left-hand side of expression (66) can again be presented at each n as a single exponential,

$$S_Q(\Delta t) = e^{\mathcal{Y}_1\tau + \mathcal{Y}_{K+1}\tau^{K+1} + \mathcal{Y}_{K+3}\tau^{K+3} + \mathcal{Y}_{K+5}\tau^{K+5} + \mathcal{Y}_{K+7}\tau^{K+7} + \dots},$$

where now

$$\begin{aligned} \mathcal{Y}_1 &= q_1\mathcal{X}_1, & \mathcal{Y}_{K+1} &= q_2\mathcal{X}_{K+1}, \\ \mathcal{Y}_{K+3} &= q_3\mathcal{X}_{K+3} + q_4[\mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_{K+1}], \\ \mathcal{Y}_{K+5} &= q_5\mathcal{X}_{K+5} + q_6[\mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_{K+3}] \\ &\quad + q_7[\mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_{K+1}], \\ \mathcal{Y}_{K+7} &= q_8\mathcal{X}_{K+7} + q_9[\mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_{K+5}] \\ &\quad + q_{10}[\mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_{K+3}] \\ &\quad + q_{11}[\mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_1, \mathcal{X}_{K+1}]. \end{aligned} \quad (67)$$

Recursive relations for multipliers q_{1-11} , corresponding to transformation (66), remain the same in form as in the case $K=4$, so that we should merely put either $K=6$ or $K=8$ in Eq. (60) to obtain the required set of order conditions.

In view of the equivalence of Eqs. (54) and (55) at $Q = K+2$, the first step on increasing the order of composition scheme to $Q=8$ when $K=6$ or $Q=10$ when $K=8$ is trivial and yields $P=2$, $d_1 = 1/(2-2^{1/(K+1)}) \equiv D_K$, and $d_2 = 1 - 2d_1$. The next steps on increasing Q to the higher values $K+4$, $K+6$, and $K+8$ at $K=6$ or 8 are similar to the steps described above for $K=4$. Namely, they lead to the necessity of solving numerically the system of P nonlinear equations, $q_1=1$, $q_2=0, \dots, q_P=0$, with $P=4, 7$, and 11 , respectively. The only difference from the case $K=4$ is that at $K=6$ or 8 and $Q=K+8$, the number of equations reduces from 12 to $P=11$, because of a somewhat simplified structure of the last truncation operator shown in Eq. (67) with respect to that appearing in Eq. (59). So that below we will present final results only with brief comments for each of the above cases. The best set among the solutions found was identified as the one that minimizes the quantity $\delta = \max_{p=1}^P |d_p|$ (this almost always led to the minimization of the norm for the main term of truncation errors as well).

For $K=6$ and $Q=10$ there are five solutions with the best set

$$\begin{aligned} d_1 &= 0.884\ 801\ 393\ 044\ 428\ 625\ 907\ 738\ 636\ 257\ 20, \\ d_2 &= 0.119\ 224\ 044\ 302\ 066\ 480\ 525\ 932\ 640\ 292\ 66, \\ d_3 &= -1.067\ 727\ 751\ 680\ 577\ 067\ 851\ 837\ 000\ 492\ 5, \end{aligned} \quad (68)$$

with $d_4 = 1 - 2(d_1 + d_2 + d_3)$ and $\delta_{\min} \equiv |d_4| = 1.127$ (within three significant digits). At $K=8$ and $Q=12$ we have found again five solutions and the optimal one is

$$\begin{aligned} d_1 &= 0.908\ 036\ 966\ 672\ 384\ 262\ 845\ 726\ 110\ 229\ 28, \\ d_2 &= 0.095\ 777\ 180\ 465\ 215\ 511\ 634\ 906\ 238\ 400\ 062, \\ d_3 &= -1.054\ 541\ 279\ 811\ 362\ 759\ 973\ 451\ 973\ 877\ 8, \end{aligned} \quad (69)$$

with $d_4 = 1 - 2(d_1 + d_2 + d_3)$ and $\delta_{\min} \equiv |d_4| = 1.101$.

For $K=6$ and $Q=12$ there were more than 150 solutions with the optimal set

$$\begin{aligned} d_1 &= 0.647\ 253\ 392\ 063\ 052\ 406\ 053\ 852\ 483\ 920\ 83, \\ d_2 &= 0.446\ 319\ 415\ 269\ 595\ 769\ 601\ 026\ 012\ 579\ 86, \\ d_3 &= -0.664\ 471\ 336\ 410\ 462\ 210\ 085\ 294\ 529\ 377\ 21, \\ d_4 &= -0.582\ 606\ 195\ 718\ 442\ 488\ 165\ 488\ 090\ 465\ 10, \\ d_5 &= 0.640\ 816\ 195\ 890\ 131\ 172\ 056\ 343\ 117\ 071\ 57, \\ d_6 &= 0.318\ 055\ 965\ 988\ 833\ 404\ 309\ 185\ 870\ 317\ 01, \end{aligned} \quad (70)$$

and $d_7 = 1 - 2(d_1 + d_2 + d_3 + d_4 + d_5 + d_6)$ with $\delta_{\min} \equiv |d_3| = 0.664$.

When $K=8$ and $Q=14$ we have computed more than 150 solutions also and identified among them the following optimal one:

$$\begin{aligned} d_1 &= 0.611\ 582\ 017\ 168\ 994\ 873\ 771\ 233\ 170\ 474\ 17, \\ d_2 &= 0.467\ 630\ 505\ 986\ 821\ 504\ 050\ 786\ 008\ 426\ 81, \\ d_3 &= -0.632\ 450\ 304\ 032\ 720\ 773\ 598\ 897\ 201\ 824\ 31, \\ d_4 &= -0.582\ 233\ 790\ 207\ 205\ 282\ 750\ 723\ 564\ 426\ 67, \\ d_5 &= 0.621\ 098\ 524\ 510\ 755\ 480\ 596\ 516\ 864\ 109\ 28, \\ d_6 &= 0.296\ 865\ 552\ 384\ 098\ 265\ 184\ 074\ 830\ 527\ 33, \end{aligned} \quad (71)$$

with $d_7 = 1 - 2(d_1 + d_2 + d_3 + d_4 + d_5 + d_6)$ and $\delta_{\min} \equiv |d_3| = 0.632$.

At $K=6$ and $Q=14$ the best set, among more than 200 solutions realized, is

$$\begin{aligned} d_1 &= 0.325\ 571\ 630\ 660\ 850\ 807\ 129\ 702\ 179\ 776\ 81, \\ d_2 &= -0.473\ 897\ 717\ 868\ 342\ 226\ 376\ 536\ 537\ 958\ 35, \end{aligned}$$

$$\begin{aligned}
d_3 &= 0.543\ 766\ 497\ 635\ 963\ 646\ 702\ 545\ 335\ 244\ 99, \\
d_4 &= -0.640\ 554\ 111\ 412\ 984\ 913\ 342\ 408\ 259\ 734\ 18, \\
d_5 &= 0.281\ 390\ 250\ 470\ 303\ 225\ 880\ 529\ 717\ 575\ 42, \\
d_6 &= 0.563\ 457\ 786\ 184\ 056\ 756\ 502\ 290\ 114\ 090\ 13, \\
d_7 &= 0.642\ 050\ 045\ 975\ 269\ 441\ 816\ 780\ 514\ 774\ 48, \\
d_8 &= -0.169\ 728\ 257\ 723\ 913\ 107\ 218\ 751\ 288\ 814\ 51, \\
d_9 &= -0.579\ 730\ 316\ 690\ 546\ 833\ 925\ 498\ 715\ 149\ 85, \\
d_{10} &= 0.273\ 985\ 802\ 830\ 633\ 798\ 706\ 233\ 909\ 797\ 62, \\
\end{aligned} \tag{72}$$

with $d_{11} = 1 - 2\sum_{p=1}^{10} d_p$ and $\delta_{\min} \equiv |d_7| = 0.642$.

Finally for $K=8$ and $Q=16$ the optimal solution, among again more than 200 sets calculated, is

$$\begin{aligned}
d_1 &= 0.296\ 422\ 548\ 914\ 130\ 709\ 533\ 124\ 502\ 130\ 71, \\
d_2 &= 0.552\ 685\ 631\ 853\ 014\ 883\ 248\ 829\ 940\ 187\ 46, \\
d_3 &= -0.581\ 343\ 395\ 355\ 333\ 933\ 156\ 055\ 443\ 099\ 40, \\
d_4 &= 0.234\ 036\ 652\ 654\ 204\ 812\ 435\ 632\ 023\ 332\ 67, \\
d_5 &= -0.517\ 889\ 589\ 898\ 170\ 553\ 039\ 786\ 588\ 274\ 53, \\
d_6 &= -0.439\ 839\ 754\ 779\ 929\ 205\ 228\ 119\ 705\ 278\ 74, \\
d_7 &= -0.201\ 370\ 781\ 509\ 421\ 699\ 574\ 681\ 119\ 934\ 44, \\
d_8 &= 0.344\ 128\ 720\ 025\ 288\ 946\ 229\ 759\ 271\ 974\ 16, \\
d_9 &= 0.030\ 725\ 917\ 609\ 965\ 587\ 988\ 954\ 283\ 097\ 65, \\
d_{10} &= 0.486\ 529\ 539\ 607\ 270\ 412\ 812\ 805\ 350\ 314\ 55, \\
\end{aligned} \tag{73}$$

with $d_{11} = 1 - 2\sum_{p=1}^{10} d_p$ and $\delta_{\min} \equiv |d_{11}| = 0.592$.

As can be seen, the quantity δ_{\min} decreases with increasing Q at any K (4, 6, and 8) considered. Besides the improvement of the precision of the integration, this leads to an extension of the region of stability of the composition algorithms. Indeed, the constants d_p appear in the compositions [see Eq. (54)] in the form of the term $d_p \Delta t$ [and its combinations of different orders when evaluating truncation uncertainties $O(\Delta t^{Q+1})$]. Then, taking into account that $\delta = \max_{p=1}^P |d_p|$, the maximal value for the size Δt of the time step, at which these uncertainties do not exceed an acceptable level of precision, can be estimated as $\Delta t_{\max} \sim 1/\delta_{\min}$. This also explains a well correlation of δ_{\min} with the minimum for the norm of truncation errors.

Sixth- and eighth-order based compositions may have advantages over algorithms based on fourth-order schemes especially when constructing very precise integrators with high values of Q . For instance, in order to derive an integrator of order $Q=16$ on the basis of triplet concatenation (55) of a

scheme of order $K=4$, it is necessary to apply $3^{(Q-K)/2} = 729$ fourth-order stages. Taking into account that each such stage requires $n_f=3$ force and $n_g=1$ force-gradient evaluations [see Eqs. (38) or (39)], one obtains in total the numbers $n_f=2187$ and $n_g=729$ corresponding to a whole time step. On the other hand, in view of result (73), an integrator of order $Q=16$ can be composed at $K=8$ and $P=11$ using $2P-1=21$ eighth-order stages for each of which $n_f=n_g=11$ [see Eq. (53)]. So that the overall number of force and force-gradient recalculations will be equal only to 231 that is much smaller than the above values 2187 and 729 obtained in the case $K=4$.

III. APPLICATIONS OF FORCE-GRADIENT ALGORITHMS

A. Molecular dynamics simulations

In molecular dynamics (MD) simulations we dealt with a system of $N=256$ identical ($m \equiv m_i$) particles interacting through a Lennard-Jones potential, $\Phi(r) = 4u[(\sigma/r)^{12} - (\sigma/r)^6]$. The particles were placed in a cubical box of volume $V=L^3$ and periodic boundary conditions have been used to exclude the finite-size effects. For the same reason, the initial potential was modified as $\varphi(r) = \Phi(r) - \Phi(r_c) - (r-r_c)\Phi'(r_c)$ at $r \leq r_c$ with $\varphi(r) = 0$ for $r > r_c$, where $r_c = L/2$ is the cutoff radius. Then the potential φ and its first-order derivative $\varphi' = d\varphi/dr$ will be continuous functions anywhere in r including the truncation point $r=r_c$. This avoids an energy drift caused by the passage of particles via the surface of truncation sphere as well as singularities of $\varphi'(r)$ and $\varphi''(r)$ at $r=r_c$. The simulations were carried out in a microcanonical ensemble at a reduced density of $n^* = N\sigma^3/V = 0.845$ and a reduced temperature of $T^* = k_B T/u = 1.7$. All runs of the length $l = 10\,000$ time steps each were started from an identical well equilibrated initial configuration $\rho(0)$. The precision of the integration 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.

The equations of motion were integrated using force-gradient algorithms (30), (36), and (38) of the fourth order within schemes A, A', B, C, and C' [see Eqs. (32), (33), (37), (41), and (42), respectively]. For the purpose of comparison the integration with the help of usual fourth-order algorithm by Forest and Ruth (FR) [2] [which represents, in fact, triplet concatenation (55) of second-order Verlet scheme (8)] has been performed as well. The corresponding results for the total energy fluctuations as functions of the length $l = t/\Delta t$ of the simulations are presented in Fig. 1(a) for a typical reduced time step of $\Delta t^* = \Delta t(u/m\sigma^2)^{1/2} = 0.005$. As can be seen, schemes A, B, and C exhibit a similar equivalence in energy conservation. This is in agreement with our theoretical predictions presented in Sec. II B 4, where the precision of algorithms has been estimated in terms of the norm γ [Eq. (34)] of fifth-order truncation errors $O(\Delta t^5)$. In particular, for schemes A, B, and C, it has been obtained as $\gamma \approx 0.000\ 713$, $0.000\ 715$, and $0.000\ 715$, respectively. Further, as expected, scheme A' ($\gamma \approx 0.003\ 34$) is worse in pre-

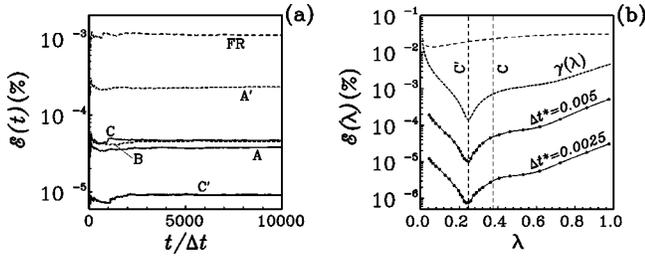


FIG. 1. (a) The total energy fluctuations \mathcal{E} as functions of the length of the simulations performed using force-gradient algorithms A, A', B, C, and C' in comparison with the result of the usual FR scheme. (b) The fluctuations \mathcal{E} obtained within the extended C'-like scheme depending on free parameter λ at two fixed time steps. The values of λ related to schemes C and C' are marked by the vertical lines. The function $\gamma(\lambda)$ is plotted by the dashed curve (see the text for additional explanations).

recision and leads to values of \mathcal{E} which are approximately $0.003\,34/0.000\,71 \approx 5$ times larger. Note that in microcanonical ensembles the total energy is an integral of motion, $E(t) = E(0)$, so that within approximate MD simulations, smaller values of \mathcal{E} correspond to a better precision of the integration. It is worth remarking also that another integral of motion, namely, total momentum $\mathbf{P} = \sum_{i=1}^N m_i \mathbf{v}_i$, is conserved exactly within force-gradient approach (13). The reasons are that all velocities are updated at once [see Eq. (14)] during each stage of decompositions and the fact that $\sum_{i=1}^N \mathbf{f}_i = 0$ as well as $\sum_{i=1}^N \mathbf{g}_i = 0$ [as can be verified readily using the structure of Eq. (11)].

The best accuracy in energy conservation can be achieved within optimized scheme C' [see Fig. 1(a)] for which $\gamma \approx 0.000\,141$. It minimizes \mathcal{E} to a level of $\sim 10^{-5}$ that is a factor $0.000\,71/0.000\,141 \approx 5$ lower than those related to schemes A, B, and C. At the same time, the usual FR algorithm leads to the worst result $\mathcal{E} \sim 10^{-3}$. We see, therefore, that applying force-gradient approach allows to reduce unphysical energy fluctuations up to two orders in magnitude. Let us show now that this overcompensates the increased computational efforts caused by additional calculations of the force gradients. The processor time used for carrying out these calculations [see Eq. (11)] was nearly three times larger than that needed for evaluations of forces itself. Further, we should take into account that algorithm C' requires $n_f = 3$ force and $n_g = 1$ force-gradient recalculations per time step, whereas $n_f = 3$ and $n_g = 0$ for FR scheme. As a result, one obtains that the size Δt of one step within FR propagation must be $(3 + 3 \times 1)/3 = 2$ times shorter than in the case of algorithm C' for spending the same overall processor time within both the cases during the integration over a fixed time interval. Finally, in view of the fact that the global error and thus the function \mathcal{E} are proportional to the fourth power of Δt , i.e., $\mathcal{E} \sim \Delta t^4$, one finds that, at the above conditions, the level of conservation of the FR scheme reduces from 10^{-3} to $\mathcal{E} \sim 10^{-3}/2^4$. So that relative efficiency of scheme C' with respect to the FR integrator is actually equal to $(10^{-3}/2^4)/10^{-5} = 100/16 \approx 6$.

In order to ensure that scheme C' [Eq. (42)] is indeed the best among the whole family (40) of C'-like integrators (38),

we carried out additional simulations in which the parameter λ , being constant within each simulation, varied from one run to another. The total energy fluctuations obtained in such simulations at the end of the runs for two (fixed within each run) undimensional time steps, namely, $\Delta t^* = 0.0025$ and 0.005 , are shown in Fig. 1(b) as functions of λ . As can be observed, the dependencies $\mathcal{E}(\lambda, \Delta t)$ have the global minimum located at the same point $\lambda \approx 0.247$ independent of the size Δt of the time step. This point coincides completely with the minimum given by Eq. (42) for the function $\gamma(\lambda)$ [see Eq. (34) and relations following just after Eq. (41)] which is also included in the subset and plotted by a dashed curve [where an upper lying part of the curve corresponds to the plus sign in Eq. (40), whereas a lower lying part as well as the simulation data are related to the minus sign], so that our criterion on measuring the precision of the integration in terms of the norm of local truncated uncertainties is in excellent accord. Moreover, the energy fluctuations appear to be proportional to that norm γ as $\mathcal{E}(\lambda, \Delta t) \sim \gamma \Delta t^4$ and the coefficient of the resulting proportionality almost does not depend on λ and Δt .

In previous studies, algorithms of such a kind have been tested for classical [26,29] and quantum [31] mechanics systems composed of a few bodies only (or even one body moving in an external field). The present investigations have demonstrated that force-gradient algorithms can be used with equal success in statistical mechanics simulations dealing with a great number of particles, i.e., when $N \gg 1$. In the last case, the calculations of force gradients also present no difficulties. Indeed, during the integration we should first evaluate usual forces \mathbf{f}_i for each particle i , where $i = 1, 2, \dots, N$. This involves number of operations proportional to the second power of N . Then in view of the structure of Eq. (11) and taking into account the fact that particle's accelerations $\mathbf{a}_i = \mathbf{f}_i/m_i$ are already known quantities, the calculations of gradients \mathbf{g}_i will require number of operations proportional to the same power of N , i.e., $\propto N^2$ (but not to $\propto N^3$, as it may look at first sight). Further reducing the computational efforts is possible, taking into account that function $\mathbf{g}(\mathbf{r}_{ij})$ [see Eq. (11)] decreases with increasing the interparticle separation r_{ij} more rapidly than initial potential $\varphi(r_{ij})$. In such a case, a secondary cutoff radius $R_c < r_c$ can be introduced when evaluating \mathbf{g}_i [which is equivalent to putting $\mathbf{g}(\mathbf{r}_{ij}) = 0$ at $r_{ij} > R_c$] in order to speed up the calculations.

B. Celestial mechanics simulations

One of the simplest ways to test force-gradient algorithms of higher orders is to apply them to solution of the two-dimensional Kepler problem. In particular, this way has been chosen by Chin and Kidwell [26,29] when testing fourth-order algorithms A, B, and C and higher-order iterated counterparts of the last scheme. As has been established, this force-gradient scheme is particularly outstanding and appears to be much more superior than usual nongradient integrators, such as fourth order by Forest and Ruth [2] as well as by Runge and Kutta [16,17], sixth order by Yoshida [1], etc. In this subsection it will be demonstrated that further significant improvement in the effectiveness of the integra-

tion can be reached by replacing standard iteration procedure (55) by advanced composition approach (54). Moreover, using our sixth- and eighth-order force-gradient algorithms as the basis for the composition has allowed us to perform the computations with extremely high precision which exceeds by several orders the accuracy observed within standard fourth-order based schemes.

We will consider the motion of a particle (planet) of mass m_1 moving in the (gravitational) field $\varphi(r) = -c/r$ of the central body (sun) with mass $m_2 \gg m_1$, where $c > 0$ is the constant responsible for intensity of the interaction. For simplifying the calculations, one neglects the influence of all other ($i = 3, 4, \dots, N$) particles (planets, for which $m_i \ll m_2$) in the (solar) system. Then the motion can be described by the following system of two equations:

$$\frac{d\mathbf{r}}{dt} = \mathbf{v}, \quad \frac{d\mathbf{v}}{dt} = -\frac{\mathbf{r}}{r^3}, \quad (74)$$

where $\mathbf{r} = \mathbf{r}_1 - \mathbf{r}_2$, and for clarity of presentation we have used units in which the reduced mass $m_1 m_2 / (m_1 + m_2)$ and the interaction constant c are equal to unity. Since the quantity $E = v^2/2 - 1/r$ (which is associated with the total energy) presents an integral of motion for Eq. (74), it should be conserved during the integration. However, this will be so if these equations are solved exactly. In numerical simulations, the local truncation uncertainties $O(\Delta t^{Q+1})$ accumulate step by step during the integration process, leading at $t \gg \Delta t$ to the global errors $O(\Delta t^Q)$, where Q denotes the order of a self-adjoint algorithm. So that the quantity E can be presented as a function of time as

$$E(t) = E_0 + E_Q(t) \Delta t^Q + O(\Delta t^{Q+2}), \quad (75)$$

where $E_0 \equiv E(0)$ and E_Q is the main step-size independent error coefficient.

In our simulations we solved two-dimensional Kepler problem (74) with the same initial conditions $\mathbf{r}(0) = (10, 0)$ and $\mathbf{v}(0) = (0, 1/10)$ as those used by previous authors [26, 29] to make comparative analysis more convenient. The resulting highly eccentric ($e = 0.9$) orbit provides a nontrivial testing ground for trajectory integration. The numerical effectiveness of each algorithm was gauged in terms of main error coefficient $E_Q = \lim_{\Delta t \rightarrow 0} [E(t) - E_0] / \Delta t^Q$ [see Eq. (75)]. It can actually be extracted from the fraction $[E(t) - E_0] / \Delta t^Q$ by choosing smaller and smaller time steps Δt to be entitled to completely ignore next higher-order corrections $O(\Delta t^{Q+2})$. This typically occurs in the neighborhood of $\Delta t \sim P/5000$, where $P = \pi / (2|E_0|^{3/2})^{1/2}$ is the period of the elliptical orbit. Since we are dealing with algorithms of high orders Q and small step sizes Δt , all the calculations have been carried out in FORTRAN using quadruple (instead of double, as for MD simulations) precision arithmetics for ensuring the correctness of the results.

The normalized energy deviations E_Q/E_0 obtained in the simulations applying fourth-, sixth-, eighth-, tenth-, twelfth-, and fourteenth-order algorithms are plotted in Figs. 2(a), 2(b), 2(c), 2(d), 2(e), and 2(f), respectively, as functions of time t during one period P of the orbit. These deviations are

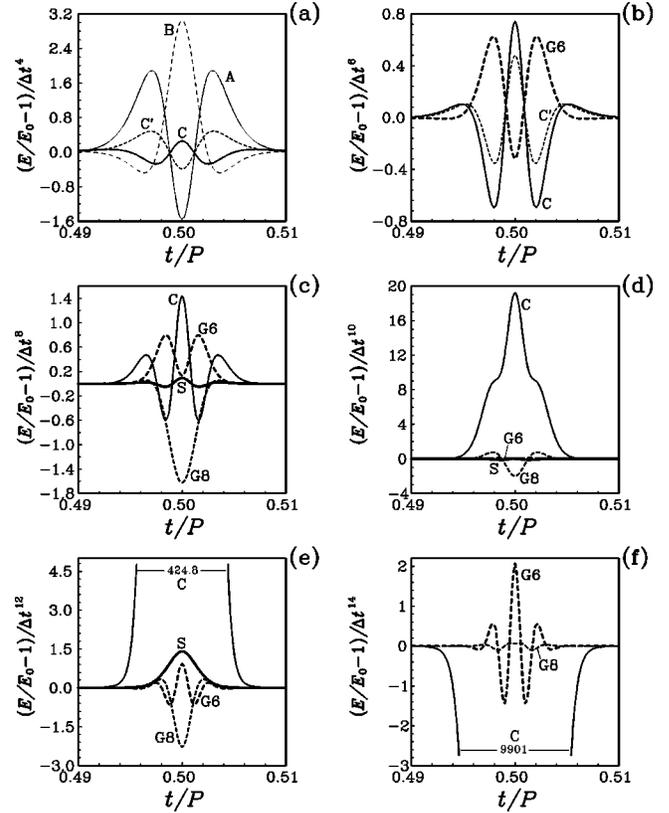


FIG. 2. The normalized energy deviation of a particle in a Keplerian orbit. The results obtained within fourth-, sixth-, eighth-, tenth-, twelfth-, and fourteenth-order algorithms are shown in (a), (b), (c), (d), (e), and (f), respectively. The basic algorithms used are: fourth-order schemes A, B, C, and C', as well as sixth- and eighth-order integrators (correspondingly marked as G6 and G8). The curves related to higher-order algorithms concatenated on the basis of scheme C by standard iterations are labeled by the same letter C in each of the sets. The fourth-, sixth-, and eighth-order based algorithms constructed within advanced composition approach are marked as S, G6, and G8, respectively (see the text).

substantial only near mid period when the particle is at its closest position to the attractive center. Note also that within symplectic integration, the nonconservation of energy for periodic orbits is periodic and its averaged (over times $t \gg P$) value is bounded and independent of t (the independence of averaged energy fluctuations at $t \gg \Delta t$ has already been demonstrated in MD simulations, see Fig. 1). That is why we presented the results in Fig. 2 within a narrow region of time near $t \sim P/2$, where the maximal deviations of E_Q will give a main contribution to the overall fluctuations.

In the case of fourth-order integration we used most typical algorithms A, B, C, and C' [see Eqs. (32), (37), (41), and (42), respectively]. As can be seen from Fig. 2(a), the pattern here is somewhat different from that in MD simulations [please compare with Fig. 1(a)]. The algorithm C is clearly better than schemes A and B, which confirms the conclusion of Ref. [26]. On the other hand, integrator C' does not exhibit an improved precision in energy conservation with respect to scheme C. Nearly the same was seen when iterating these algorithms to higher orders with the help of triplet

construction (55). In particular, the sixth-order C' counterpart appeared to be even slightly better than the corresponding counterpart of scheme C (see Fig. 2(b)). At the same time, higher-order integrators based on schemes A and B were definitely worse. So that the obvious candidates for fourth-order based iterations (55) and compositions (54) are schemes C and C' .

In order to understand why scheme C' does not lead to the expected improvement over scheme C in this particular situation, it should be taken into account that we deal with a small system, actually with one body moving in an effective external field. Moreover, such a body moves periodically and, thus, covers only small part of phase space during its displacement. This is contrary to many-body statistical systems, where the phase point may visit considerably wider regions of phase space. In the latter case, during the averaging along the phase trajectories, different components γ_{1-4} of fifth-order local uncertainties [see Eq. (17)] will enter with approximately the same weights when forming the total error vector $O(\Delta t^5)$. This has been tentatively assumed when writing the norm γ of that vector in the form of Eq. (34) and further minimizing γ to obtain algorithm C' . In the case of a few-body system, especially with periodic motion, the above weights may differ considerably. This complicates an analysis of the truncation terms and makes it impossible to find an exact global minimum for them within any analytical approach. Note, however, that even here, the assumption on uniform contribution of truncation-error components works relatively well. Indeed, in view of dependencies shown in Figs. 2(a) and 2(b), we can say that both the schemes C and C' are comparable in precision. The same was observed for their higher-order counterparts. For this reason (and to reserve more free space for other dependencies), in Figs. 2(c)–2(f) we will draw only curves corresponding to scheme C .

When considering the sixth-order integration, we realized that direct velocitylike scheme defined by Eqs. (45) and (47) is much worse (the maximum deviation of E_Q were more than two orders larger) than its extended positionlike counterpart given by Eqs. (50) and (51). This is in agreement with a prediction of Sec. II B 5. The result corresponding to the positionlike algorithm is plotted in Fig. 1(b) by the bold dashed curve marked as $G6$. As can be seen, all three curves shown in Fig. 1(b), namely, C , C' , and $G6$ are close enough to each other. But algorithm $G6$ uses only $n_f=5$ force evaluations per time step, instead of $n_f=9$ needed for iterated C - and C' -like schemes (for all these three cases the number of force-gradient evaluations is the same and equal to $n_g=3$). Therefore, for order six, direct decomposition approach (13) leads to more efficient integration than concatenations of fourth-order schemes.

Beginning from order eight, the above concatenations based on standard iterations (55) and advanced compositions (54) will result in completely different integrators. The simulation data for these iterated and composed C -based integrators are shown in Fig. 2(c) by thin (marked simply as C) and bold (marked as S) solid curves, respectively. The curves related to tenth- and twelfth-order iteration and composition integrators (based on the same fourth-order scheme C) are plotted correspondingly in Figs. 2(d) and 2(e), and marked

by the same letters C and S . We mention that C -marked curves have already been presented in the work by Chin and Kidwell [26,29] up to order 12. They are redisplayed by us in order to illustrate the evident superiority of our composition approach over the standard iteration method. Indeed, for the iteration integrators (C -marked curves) of orders $Q=8, 10$, and 12, the magnitudes of the normalized energy coefficient E_Q/E_0 after one period are 1.44, 19.24, and 424.8, respectively. On the other hand, the magnitudes related to the composition integrators (S -marked curves) constitute correspondingly 0.0953, 0.0577, and 1.41, i.e., they are approximately 15, 330, and 300 times smaller. In addition, the composition integrators are faster with respect to their iteration versions in factors 9/7, 27/13, and 81/23 for $Q=8, 10$, and 12, respectively (see Sec. III A 1), and thus the resulting efficiencies will increase yet.

What about sixth- and eight-order-based composition schemes at $Q \geq 8$? First of all, let us consider the case of eight-order integration. Here, the direct scheme chosen was position-like integrator (53) [it leads to better energy conservation with respect to its velocitylike counterpart (52)]. The result corresponding to this integrator is plotted in Fig. 2(c) by the dashed curve marked as $G8$. As can be seen, the fourth-order-based composition scheme (S curve) is better at $Q=8$ with respect to both direct $G8$ and iterated $G6$ -like versions. With increasing the order to 10 and 12, they all become nearly equivalent in the accuracy of energy conservation. But fourth-order-based approach requires somewhat fewer number of operations. For instance, for order 12, one obtains that the numbers of force and force-gradient evaluations per time step are equal for it to $n_f=23 \times 3=69$ and $n_g=23$, respectively, whereas these numbers for sixth- and eighth-order-based compositions $G6$ and $G8$ are $n_f=13 \times 5=65$, $n_g=13 \times 3=39$, and $n_f=n_g=7 \times 11=77$ (where $G6$ integrator requires less operations than $G8$ scheme). However, beginning from order 14, the situation reverses. The fourth-order-based composition S approach is no longer accessible (because of the absence of explicit expressions for its time coefficients here). On the other hand, applying the standard fourth-order-based iteration C method is very inefficient. In particular, at $Q=14$ the maximal energy deviation within this method is $|E_{14}/E_0|_{\max}=9901$ with $n_f=21 \times 5=729$ and $n_g=21 \times 3=243$. At the same time, the higher-order-based composition schemes lead to much accurate results, namely, $|E_{14}/E_0|_{\max}=2.065$ with $n_f=21 \times 5=105$ and $n_g=21 \times 3=63$ for $G6$ - as well as $|E_{14}/E_0|_{\max}=0.101$ with $n_f=n_g=13 \times 11=143$ for $G8$ -based schemes (where the better precision for the last scheme compensates to some extent its increased values for quantities n_f and n_g). We see, therefore, that the relative efficiencies of $G6$ - and $G8$ -based schemes with respect to C approach constitute about 10^4 – 10^5 . Finally, in the case $Q=16$ (not shown in Fig. 2) we have obtained the values $|E_{16}/E_0|_{\max}=2.43 \times 10^5$ and $|E_{16}/E_0|_{\max}=48.16$ corresponding to C - and $G8$ -based schemes, respectively. Taking into account the numbers of n_f and n_g for these schemes presented at the end of Sec. II C 2, one can conclude that the efficiency increases here also approximately in 10^4 to 10^5 times.

IV. CONCLUDING REMARKS

In this work we have formulated a general theory of construction of force-gradient algorithms for solving the equations of motion in classical and quantum systems. This has allowed us to extend considerably the class of analytically integrable symplectic schemes. The algorithms derived include self-adjoint direct decomposition integrators of orders two, four, six, and eight as well as their composition counterparts up to the sixteenth order in the time step. As has been proven theoretically and confirmed in actual numerical simulations, these algorithms lead to significant improvement in the efficiency of the integration with respect to existing force-gradient and nongradient schemes. It has been demonstrated that force-gradient algorithms can be used with equal success for describing the motion in few-body classical and quantum mechanics systems as well as for performing statistical molecular dynamics observations over many-particle collections. In all the cases the calculation of force gradients presents no difficulties and requires computational efforts comparable with those needed to evaluate usual forces itself. The proposed algorithms may be especially useful for the prediction and study of very subtle dynamical effects in different areas of physics and chemistry including the problems of astrophysical interest, whenever the precise integration of motion during very long times is desirable.

The algorithms introduced exactly reproduce such important features of classical dynamics as time reversibility and symplecticity. This explains their excellent energy conservation and stability properties. In this context it is worth mentioning another class of (nongradient) integrators recently developed [34] on the basis of a modified Runge-Kutta approach. Like the force-gradient algorithms, the Runge-Kutta-like integrators also allow to produce time reversible and symplectic trajectories in phase space with, in principle, arbitrary order in precision. However, such integrators are implicit and require cumbersome systems of globally coupled (via positions and forces of all particles) nonlinear equations to be solved by expensive iterations at each step of the integration process. Since, in practice, such equations cannot be solved exactly, the time reversibility and symplecticity can be violated. This may lead, in particular, to instabilities in long-term energy conservation, i.e., to the same problem inherent in the traditional (nonsymplectic) Runge-Kutta method (see the Introduction). All these disadvantages are absent in the present approach, where the phase trajectories are propagated explicitly in time by applying consecutive simple shifts of particles in velocity and position space with exact preservation of the phase volume and reversibility of the generated solutions.

The approach presented can also be adapted to the integration of motion in more complicated systems, such as systems with orientational or spin degrees of freedom, etc., where splitting of the Liouville operator into more than two parts may be necessary to obtain analytically solvable sub-propagators. These and other related problems will be considered in a separate investigation.

ACKNOWLEDGMENT

Part of this work was supported by the Fonds zur Förderung der wissenschaftlichen Forschung under Project No. 15247.

APPENDIX

The recursive relations for the higher-order multipliers ζ_{1-10} [see Eqs. (13), (15), and (18)] corresponding to the first type of self-adjoint transformations given by the first line of Eq. (19) are

$$\begin{aligned}\zeta_1^{(n+1)} = & \zeta_1^{(n)} + a^{(n)}(630\beta^{(n)2} + 1260\gamma_4^{(n)}\sigma^{(n)} - 63\beta^{(n)}(6a^{(n)} \\ & + \nu^{(n)})\sigma^{(n)2} + \sigma^{(n)3}\{21\alpha^{(n)} + (27a^{(n)2} + 9a^{(n)}\nu^{(n)} \\ & + \nu^{(n)2})\sigma^{(n)}\})/3780,\end{aligned}$$

$$\begin{aligned}\zeta_2^{(n+1)} = & \zeta_2^{(n)} + a^{(n)}(336\beta^{(n)}(6a^{(n)} + \nu^{(n)})\sigma^{(n)2} - 5040\beta^{(n)2} \\ & - 5040\gamma_4^{(n)}\sigma^{(n)} - \sigma^{(n)3}\{336\alpha^{(n)} + (120a^{(n)2} \\ & + 12a^{(n)}\nu^{(n)} - \nu^{(n)2})\sigma^{(n)}\})/45\,360,\end{aligned}$$

$$\begin{aligned}\zeta_3^{(n+1)} = & \zeta_3^{(n)} - a^{(n)}\{5040\alpha^{(n)}\beta^{(n)} + \sigma^{(n)}[5040\gamma_2^{(n)} \\ & - 84\beta^{(n)}\nu^{(n)2} + 72a^{(n)3}\sigma^{(n)2} + \nu^{(n)3}\sigma^{(n)2} \\ & + 24a^{(n)}\nu^{(n)}(\nu^{(n)}\sigma^{(n)2} - 42\beta^{(n)}) \\ & + a^{(n)2}(88\nu^{(n)}\sigma^{(n)2} - 672\beta^{(n)})\}/15\,120,\end{aligned}$$

$$\begin{aligned}\zeta_4^{(n+1)} = & \zeta_4^{(n)} + a^{(n)}\{168\alpha^{(n)}[60\beta^{(n)} - (6a^{(n)} + \nu^{(n)})\sigma^{(n)2}] \\ & + \sigma^{(n)}[10\,080\gamma_2^{(n)} + 5040\gamma_3^{(n)} - 168\beta^{(n)}\nu^{(n)2} \\ & + 192a^{(n)3}\sigma^{(n)2} + 5\nu^{(n)3}\sigma^{(n)2} \\ & + 6a^{(n)}\nu^{(n)}(13\nu^{(n)}\sigma^{(n)2} - 336\beta^{(n)}) \\ & + a^{(n)2}(272\nu^{(n)}\sigma^{(n)2} - 1344\beta^{(n)})\}/15\,120,\end{aligned}$$

$$\begin{aligned}\zeta_5^{(n+1)} = & \zeta_5^{(n)} - a^{(n)}\{2520\gamma_4^{(n)}\nu^{(n)} + 7560\gamma_3^{(n)}\sigma^{(n)} \\ & - 294\beta^{(n)}\nu^{(n)2}\sigma^{(n)} + 180a^{(n)3}\sigma^{(n)3} - \nu^{(n)3}\sigma^{(n)3} \\ & + 84\alpha^{(n)}[120\beta^{(n)} + (3\nu^{(n)} - 22a^{(n)})\sigma^{(n)2}] \\ & + a^{(n)2}(234\nu^{(n)}\sigma^{(n)3} - 1512\beta^{(n)}\sigma^{(n)}) \\ & + 6a^{(n)}(420\gamma_4^{(n)} - 308\beta^{(n)}\nu^{(n)}\sigma^{(n)}) \\ & + 3\nu^{(n)2}\sigma^{(n)3}\}/45\,360,\end{aligned}\tag{A1}$$

$$\begin{aligned}\zeta_6^{(n+1)} = & \zeta_6^{(n)} + a^{(n)}(18a^{(n)3}\sigma^{(n)3} - 84\alpha^{(n)}[15\beta^{(n)} - (a^{(n)} \\ & + \nu^{(n)})\sigma^{(n)2}] + a^{(n)2}(15\nu^{(n)}\sigma^{(n)3} - 252\beta^{(n)}\sigma^{(n)}) \\ & + 6a^{(n)}(210\gamma_4^{(n)} - 28\beta^{(n)}\nu^{(n)}\sigma^{(n)} + \nu^{(n)2}\sigma^{(n)3}) \\ & + 2(630\gamma_4^{(n)}\nu^{(n)} - 630\gamma_2^{(n)}\sigma^{(n)} - 42\beta^{(n)}\nu^{(n)2}\sigma^{(n)} \\ & + \nu^{(n)3}\sigma^{(n)3})/7560,\end{aligned}$$

$$\begin{aligned} \zeta_7^{(n+1)} &= \zeta_7^{(n)} + a^{(n)}(2520\alpha^{(n)2} - 84\alpha^{(n)}[8a^{(n)2} + 12a^{(n)}\nu^{(n)} \\ &\quad + \nu^{(n)2}]\sigma^{(n)} + \sigma^{(n)}\{5040\gamma_1^{(n)} + [48a^{(n)4} \\ &\quad + 120a^{(n)3}\nu^{(n)} + 92a^{(n)2}\nu^{(n)2} + 18a^{(n)}\nu^{(n)3} \\ &\quad + \nu^{(n)4}]\sigma^{(n)}\})/15\,120, \\ \zeta_8^{(n+1)} &= \zeta_8^{(n)} - a^{(n)}\{5040\alpha^{(n)2} + 2520\gamma_2^{(n)}\nu^{(n)} - 42\beta^{(n)}\nu^{(n)3} \\ &\quad + 2520\gamma_1^{(n)}\sigma^{(n)} - 420\alpha^{(n)}a^{(n)}(a^{(n)} + 2\nu^{(n)})\sigma^{(n)} \\ &\quad + 69a^{(n)4}\sigma^{(n)2} + \nu^{(n)4}\sigma^{(n)2} + 2a^{(n)2}\nu^{(n)}(53\nu^{(n)}\sigma^{(n)2} \\ &\quad - 294\beta^{(n)}) + a^{(n)3}(148\nu^{(n)}\sigma^{(n)2} - 294\beta^{(n)}) \\ &\quad + 6a^{(n)}(420\gamma_2^{(n)} - 56\beta^{(n)}\nu^{(n)2} \\ &\quad + 3\nu^{(n)3}\sigma^{(n)2})\}/15\,120, \\ \zeta_9^{(n+1)} &= \zeta_9^{(n)} + a^{(n)}(2520\alpha^{(n)2} - 42\alpha^{(n)}[8a^{(n)2} + 12a^{(n)}\nu^{(n)} \\ &\quad + \nu^{(n)2}]\sigma^{(n)} + 114a^{(n)4}\sigma^{(n)2} - 4a^{(n)3}(147\beta^{(n)} \\ &\quad - 59\nu^{(n)}\sigma^{(n)2}) + a^{(n)2}\nu^{(n)}(173\nu^{(n)}\sigma^{(n)2} - 1176\beta^{(n)}) \\ &\quad + 24a^{(n)}(210\gamma_2^{(n)} + 105\gamma_3^{(n)} - 28\beta^{(n)}\nu^{(n)2} \\ &\quad + 2\nu^{(n)3}\sigma^{(n)2}) + \nu^{(n)}(5040\gamma_2^{(n)} + 2520\gamma_3^{(n)} \\ &\quad - 84\beta^{(n)}\nu^{(n)2} + 5\nu^{(n)3}\sigma^{(n)2})/15\,120, \\ \zeta_{10}^{(n+1)} &= \zeta_{10}^{(n)} + a^{(n)}(a^{(n)} + \nu^{(n)})\{2520\gamma_1^{(n)} - 42\alpha^{(n)}[7a^{(n)2} \\ &\quad + 7a^{(n)}\nu^{(n)} + \nu^{(n)2}] + [31a^{(n)4} + 62a^{(n)3}\nu^{(n)} \\ &\quad + 42a^{(n)2}\nu^{(n)2} + 11a^{(n)}\nu^{(n)3} + \nu^{(n)4}]\sigma^{(n)}\}/15\,120. \end{aligned}$$

For the transformation of the second type [see the second line of Eq. (19)] we have obtained

$$\begin{aligned} \zeta_1^{(n+1)} &= \zeta_1^{(n)} - (18b^{(n)4}\nu^{(n)3} + 15b^{(n)3}\nu^{(n)3}\sigma^{(n)} \\ &\quad + 42c^{(n)}\nu^{(n)}(30\beta^{(n)} + 30c^{(n)} - \nu^{(n)}\sigma^{(n)2}) \\ &\quad - 84\alpha^{(n)}(15\beta^{(n)}b^{(n)} + 30b^{(n)}c^{(n)} - 3b^{(n)3}\nu^{(n)} \\ &\quad + 15c^{(n)}\sigma^{(n)} - 2b^{(n)2}\nu^{(n)}\sigma^{(n)} - b^{(n)}\nu^{(n)}\sigma^{(n)2}) \\ &\quad - 6b^{(n)2}[210\gamma_2^{(n)} + \nu^{(n)2}(14\beta^{(n)} + 63c^{(n)} \\ &\quad - \nu^{(n)}\sigma^{(n)2})] + b^{(n)}\{1260\gamma_4^{(n)}\nu^{(n)} - 2\sigma^{(n)}[630\gamma_2^{(n)} \\ &\quad + \nu^{(n)2}(42\beta^{(n)} + 84c^{(n)} - \nu^{(n)}\sigma^{(n)2})]\}/7560, \\ \zeta_2^{(n+1)} &= \zeta_2^{(n)} + (12b^{(n)4}\nu^{(n)3} - 39b^{(n)3}\nu^{(n)3}\sigma^{(n)} \\ &\quad + 42c^{(n)}\nu^{(n)}(120\beta^{(n)} + 120c^{(n)} - \nu^{(n)}\sigma^{(n)2}) \\ &\quad - 252\alpha^{(n)}(20\beta^{(n)}b^{(n)} + 40b^{(n)}c^{(n)} - 3b^{(n)3}\nu^{(n)} \\ &\quad + 20c^{(n)}\sigma^{(n)} - 2b^{(n)2}\nu^{(n)}\sigma^{(n)} - b^{(n)}\nu^{(n)}\sigma^{(n)2}) \\ &\quad + 24b^{(n)2}[315\gamma_3^{(n)} - \nu^{(n)2}(21\beta^{(n)} + 42c^{(n)} \end{aligned}$$

$$\begin{aligned} &\quad + \nu^{(n)}\sigma^{(n)2})] + b^{(n)}\{2520\gamma_4^{(n)}\nu^{(n)} + \sigma^{(n)}[7560\gamma_3^{(n)} \\ &\quad - \nu^{(n)2}(294\beta^{(n)} + 168c^{(n)} + \nu^{(n)}\sigma^{(n)2})]\}/45\,360, \\ \zeta_3^{(n+1)} &= \zeta_3^{(n)} - [2520\alpha^{(n)2}b^{(n)} + 57b^{(n)3}\nu^{(n)4} \\ &\quad - 840\alpha^{(n)}\nu^{(n)}(3c^{(n)} - b^{(n)2}\nu^{(n)}) + 42c^{(n)}\nu^{(n)3}\sigma^{(n)} \\ &\quad - 12b^{(n)2}(210\gamma_1^{(n)} - \nu^{(n)4}\sigma^{(n)}) - b^{(n)}(2520\gamma_2^{(n)}\nu^{(n)} \\ &\quad - 42\beta^{(n)}\nu^{(n)3} + 336c^{(n)}\nu^{(n)3} + 2520\gamma_1^{(n)}\sigma^{(n)} \\ &\quad + \nu^{(n)4}\sigma^{(n)2})]/15\,120, \\ \zeta_4^{(n+1)} &= \zeta_4^{(n)} + (5040\alpha^{(n)2}b^{(n)} - 42\alpha^{(n)}\nu^{(n)}[120c^{(n)} - b^{(n)} \\ &\quad \times \nu^{(n)}(36b^{(n)} + \sigma^{(n)})] + \nu^{(n)}\{96b^{(n)3}\nu^{(n)3} \\ &\quad + 84c^{(n)}\nu^{(n)2}\sigma^{(n)} + 18b^{(n)2}\nu^{(n)3}\sigma^{(n)} - b^{(n)}[5040\gamma_2^{(n)} \\ &\quad + 2520\gamma_3^{(n)} - \nu^{(n)2}(84\beta^{(n)} - 672c^{(n)} \\ &\quad - 5\nu^{(n)}\sigma^{(n)2})]\}/15\,120, \tag{A2} \\ \zeta_5^{(n+1)} &= \zeta_5^{(n)} - [2520\alpha^{(n)2}b^{(n)} - 36b^{(n)3}\nu^{(n)4} \\ &\quad + 42c^{(n)}\nu^{(n)3}\sigma^{(n)} + 30b^{(n)2}\nu^{(n)4}\sigma^{(n)} \\ &\quad + 168\alpha^{(n)}\nu^{(n)}[15c^{(n)} - b^{(n)}\nu^{(n)}(6b^{(n)} + \sigma^{(n)})] \\ &\quad - b^{(n)}[15\,120\gamma_3^{(n)}\nu^{(n)} - \nu^{(n)3}(252\beta^{(n)} + 504c^{(n)} \\ &\quad + \nu^{(n)}\sigma^{(n)2})]\}/45\,360, \\ \zeta_6^{(n+1)} &= \zeta_6^{(n)} - \{630\alpha^{(n)2}b^{(n)} + 27b^{(n)3}\nu^{(n)4} - 21c^{(n)}\nu^{(n)3}\sigma^{(n)} \\ &\quad + 9b^{(n)2}\nu^{(n)4}\sigma^{(n)} - 63\alpha^{(n)}\nu^{(n)}[20c^{(n)} - b^{(n)}\nu^{(n)} \\ &\quad \times (6b^{(n)} + \sigma^{(n)})] - b^{(n)}[1260\gamma_2^{(n)}\nu^{(n)} + \nu^{(n)3}(21\beta^{(n)} \\ &\quad + 252c^{(n)} - \nu^{(n)}\sigma^{(n)2})]\}/3780, \\ \zeta_7^{(n+1)} &= \zeta_7^{(n)} - b^{(n)}\nu^{(n)}[2520\gamma_1^{(n)} - 42\alpha^{(n)}\nu^{(n)2} - \nu^{(n)4}(6b^{(n)} \\ &\quad - \sigma^{(n)})]/15\,120, \\ \zeta_8^{(n+1)} &= \zeta_8^{(n)} + (5040b^{(n)}\gamma_1^{(n)}\nu^{(n)} - 42c^{(n)}\nu^{(n)4} - 6b^{(n)2}\nu^{(n)5} \\ &\quad + b^{(n)}\nu^{(n)5}\sigma^{(n)})/15\,120, \\ \zeta_9^{(n+1)} &= \zeta_9^{(n)} - \nu^{(n)3}\{84\alpha^{(n)}b^{(n)} - \nu^{(n)}[84c^{(n)} - b^{(n)}\nu^{(n)} \\ &\quad \times (12b^{(n)} + 5\sigma^{(n)})]\}/15\,120, \\ \zeta_{10}^{(n+1)} &= \zeta_{10}^{(n)} - b^{(n)}\nu^{(n)6}/15\,120. \end{aligned}$$

All these relations, as well as other symbolic expressions presented in the work, have been carried out using MATHEMATICA 4.0 and MAPLE 6 packages installed on the Silicon Graphics Origin 3800 workstation at Linz University. The numerical calculations have also been performed there.

- [1] H. Yoshida, *Phys. Lett. A* **150**, 262 (1990).
- [2] E. Forest and R. D. Ruth, *Physica D* **43**, 105 (1990).
- [3] M. Suzuki, *Phys. Lett. A* **165**, 387 (1992).
- [4] E. Forest, *J. Comput. Phys.* **99**, 209 (1992).
- [5] 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).
- [6] R.-C. Li, Ph.D. thesis, Department of Mathematics, University of California at Berkeley, CA, 1995.
- [7] A. Dullweber, B. Leimkuhler, and R. McLachlan, *J. Chem. Phys.* **107**, 5840 (1997).
- [8] M. Krech, A. Bunker, and D. P. Landau, *Comput. Phys. Commun.* **111**, 1 (1998).
- [9] I. P. Omelyan, I. M. Mryglod, and R. Folk, *Phys. Rev. Lett.* **86**, 898 (2001).
- [10] D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications* (Academic Press, New York, 1996).
- [11] J. Wisdom and M. Holman, *Astrophys. J.* **102**, 1528 (1991).
- [12] C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations* (Prentice-Hall, Engelwood Cliffs, NJ, 1971).
- [13] R. L. Burden and J. D. Faires, *Numerical Analysis*, 5th ed. (PWS Publishing, Boston, 1993).
- [14] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Clarendon, Oxford, 1987).
- [15] I. P. Omelyan, *Phys. Rev. E* **58**, 1169 (1998).
- [16] H. Kinoshita, H. Yoshida, and H. Nakai, *Celest. Mech.* **50**, 59 (1991).
- [17] B. Gladman, M. Duncan, and J. Candy, *Celest. Mech.* **52**, 221 (1991).
- [18] W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson, *J. Chem. Phys.* **76**, 637 (1982).
- [19] M. Tuckerman, B. J. Berne, and G. J. Martyna, *J. Chem. Phys.* **97**, 1990 (1992).
- [20] M. Qin and W. J. Zhu, *Computing* **47**, 309 (1992).
- [21] R. I. McLachlan, *SIAM J. Sci. Comput. (USA)* **16**, 151 (1995).
- [22] W. Kahan and R.-C. Li, *Math. Comput.* **66**, 1089 (1997).
- [23] A. Murua and J. M. Sanz-Serna, *Philos. Trans. R. Soc. London, Ser. A* **357**, 1079 (1999).
- [24] 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).
- [25] M. Suzuki, *Phys. Lett. A* **201**, 425 (1995).
- [26] S. A. Chin, *Phys. Lett. A* **226**, 344 (1997).
- [27] A. N. Drozdov and J. J. Brey, *Phys. Rev. E* **57**, 1284 (1998).
- [28] H. A. Forbert and S. A. Chin, *Phys. Rev. E* **63**, 016703 (2000).
- [29] S. A. Chin and D. W. Kidwell, *Phys. Rev. E* **62**, 8746 (2000).
- [30] M. Creutz and A. Gocksch, *Phys. Rev. Lett.* **63**, 9 (1989).
- [31] S. A. Chin and C. R. Chen, *J. Chem. Phys.* **114**, 7338 (2001).
- [32] F. A. Bornemann, P. Nettesheim, and Ch. Schütte, *J. Chem. Phys.* **105**, 1074 (1996).
- [33] R. McLachlan, *BIT Numerical Mathematics* **35**, 258 (1995).
- [34] M. Sofroniou and W. Oevel, *SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal.* **34**, 2063 (1997).