Gradient symplectic algorithms for solving the radial Schrödinger equation

Siu A. Chin[a] and Petr Anisimov
Department of Physics, Texas A&M University, College Station, Texas 77843

(Received 28 September 2005; accepted 18 November 2005; published online 2 February 2006)

The radial Schrödinger equation for a spherically symmetric potential can be regarded as a one-dimensional classical harmonic oscillator with a time-dependent spring constant. For solving classical dynamics problems, symplectic integrators are well known for their excellent conservation properties. The class of *gradient* symplectic algorithms is particularly suited for solving harmonic-oscillator dynamics. By use of Suzuki's rule [Proc. Jpn. Acad., Ser. B: Phys. Biol. Sci. **69**, 161 (1993)] for decomposing time-ordered operators, these algorithms can be easily applied to the Schrödinger equation. We demonstrate the power of this class of gradient algorithms by solving the spectrum of highly singular radial potentials using Killingbeck's method [J. Phys. A **18**, 245 (1985)] of backward Newton-Ralphson iterations. © *2006 American Institute of Physics*.
[DOI: 10.1063/1.2150831]

I. INTRODUCTION

Because of its physical importance, an immense literature exists for solving the radial Schrödinger equation,

$$\frac{d^2u(r)}{dr^2} = f(r,E)u(r), \tag{1.1}$$

where

$$f(r,E) = 2V(r) - 2E + \frac{\ell(\ell+1)}{r^2}. \tag{1.2}$$

This is usually solved by finite difference methods, such as the well-known fourth-order Numerov[1] algorithm, or further improved schemes.[2] Recent devlopments[3–5] have resulted in many exponentially fitted algorithms which seek to integrate (1.1) exactly when f is a constant. As we will see, because f can vary rapidly with $V(r)$, specially in the case of singular potentials, these algorithms do not, in general, perform better than nonfitted algorithms.

If we relabel the variables $r \to t$ and $u \to q$, (1.1) formally resembles a one-dimensional (1D) harmonic oscillator with a time-dependent spring constant $k(t,E) = -f(t,E)$,

$$\frac{d^2q(t)}{dt^2} = -k(t,E)q(t). \tag{1.3}$$

The difference here is that $k(t,E)$ can change sign with time and E must be determined simultaneously with $q(t)$ to satisfy the correct "large time" (boundary) condition. How this can be done efficiently will be discuss in Sec. IV. But assuming that E is given, then the dynamics of (1.3) corresponds to a Hamiltonian with an explicit time-dependent potential,

$$H = \frac{1}{2}p^2 + \frac{1}{2}k(t,E)q^2, \tag{1.4}$$

and any algorithm that can solve the classical time-dependent force problem can be used to solve the radial Schrödinger equation. For example, one can use Runge-Kutta-type algorithms.[6] However, for solving classical dynamics, symplectic integrators[7–10] are algorithms of choice because they conserve all Poincaré invariants and are deeply rooted in the Poisson formulation of classical mechanics. For oscillatory problems, symplectic algorithms are known to conserve energy and reduce phase error much better than Runge-Kutta-type algorithms.[11–15] The difficulty here is that in order to derive an algorithm for solving time-dependent dynamics, one must solve the problem of time-ordered exponential. Liu *et al.*[16] have recognized the time-dependent Hamiltonian structure of the Schrödinger equation, but were able to solve the time-dependent exponential, and devised a symplectic algorithm, only to second order. Kalogiratou *et al.*[17] have proposed a third-order symplectic algorithm by expanding out the exponential to third order. Such a brute force approach cannot be extended to higher orders. A more systematic way of dealing with the time-ordered exponential is via the Magnus expansion,[18–20] but the Magnus expansion requires explicit time integration in addition to evaluating higher-order commutators. A more elegant solution is Suzuki's[21] reinterpretation of the time-ordered exponential as reviewed in Ref. 22. By adapting Suzuki's rule, any factorized symplectic algorithms can be used to solve problems with an explicit time-dependent potential,[22] including the disguised radial Schrödinger equation (1.3).

In order to devise efficient algorithms for solving the radial Schrödinger equation (1.1), one must take advantage of its harmonic-oscillator character (1.3). Most algorithms, even factorized symplectic ones, are general purpose algorithms and are not specially adapted for solving the time-dependent harmonic oscillator. However, the recent class of *gradient* symplectic algorithms,[12–15,22–24] while general, seem tailor-made for solving harmonic-type dynamics. This is because these algorithms require computing the force gradient in addition to the force. While the force gradient is not difficult to compute, it is especially trivial in the case of the harmonic oscillator. This class of gradient (or more specifi-

[a]Electronic mail: chin@physics.tamu.edu

cally, *forward*) integrators has been shown to be efficient in solving both classical[12–15,23,24] and quantum[19,20,22,25,26] dynamical problems. In this work, we will show that they are also ideally suited for solving the radial Schrödinger equation.

In the next section, we briefly summarize the Lie-Poisson operator formulation of symplectic integrators and Suzuki's rule for factorizing time-ordered exponentials. In Sec. III, we describe forward, gradient-based symplectic algorithms. In Sec. IV, we review Killingbeck's method[27] of eigenvalue-function determination. In Sec. V, we compare results on the Coulomb and other singular radial potentials. In Sec. VI, we discuss the applicability of sixth-order algorithms and draw some conclusions in Sec. VII.

II. TIME-DEPENDENT SYMPLECTIC ALGORITHMS

The Poisson bracket for evolving any dynamical variable $W(q,p)$ can be regarded as an operator equation,

$$\frac{d}{dt}W(q,p) = \{W,H\} \equiv \left(\frac{\partial H}{\partial p}\frac{\partial}{\partial q} - \frac{\partial H}{\partial q}\frac{\partial}{\partial p}\right)W, \tag{2.1}$$

with formal solution

$$W(t+\varepsilon) = e^{\varepsilon(T+V)}W(t). \tag{2.2}$$

For the standard Hamiltonian,

$$H(p,q) = \frac{1}{2}p^2 + V(q), \tag{2.3}$$

the operators T and V are first-order differential operators,

$$T = \frac{\partial H}{\partial p}\frac{\partial}{\partial q} = p\frac{\partial}{\partial q}, \quad V = -\frac{\partial U}{\partial q}\frac{\partial}{\partial p} = F(q)\frac{\partial}{\partial p}. \tag{2.4}$$

The Lie transforms[7] $e^{\varepsilon T}$ and $e^{\varepsilon V}$, are then *displacement* operators which displace q and p forward in time via

$$q \to q + \varepsilon p \text{ and } p \to p + \varepsilon F. \tag{2.5}$$

Each factorization of $e^{\varepsilon(T+V)}$ into products of $e^{\varepsilon T}$ and $e^{\varepsilon V}$ (and exponentials of commutators of T and V) gives rise to a *symplectic* algorithm, which is a sequence of successive displacements (2.5) for evolving the system forward in time. This is the fundamental Lie-Poisson theory of symplectic integrators which has been studied extensively in the literature.[7–10]

For a time-dependent Hamiltonian,

$$H(t) = \frac{1}{2}p^2 + V(q,t), \tag{2.6}$$

the solution is given by the time-ordered exponential,

$$W(t+\varepsilon) = T\exp\left(\int_t^{t+\varepsilon}[T+V(s)]ds\right)W(t), \tag{2.7}$$

where $V(t)$ is now the explicitly time-dependent operator,

$$V(t) = -\frac{\partial U(q,t)}{\partial q}\frac{\partial}{\partial p} = F(q,t)\frac{\partial}{\partial p}. \tag{2.8}$$

Suzuki proved[21] that

$$T\exp\left(\int_t^{t+\varepsilon}[T+V(s)]ds\right) = e^{\varepsilon(T+V(t)+D)}, \tag{2.9}$$

where D is the *forward time derivative* operator

$$D = \frac{\overleftarrow{\partial}}{\partial t} \tag{2.10}$$

such that for any two time-dependent functions $F(t)$ and $G(t)$,

$$F(t)e^{\varepsilon D}G(t) = F(t+\varepsilon)G(t). \tag{2.11}$$

Thus symplectic algorithms for solving explicitly time-dependent problems of the form (2.6) can be obtained by factorizing the three-operator exponential of (2.9). Since D commutes with T, one can first group

$$\widetilde{T} = T + D \tag{2.12}$$

and factorize $\widetilde{T}$ and $V(t)$ as in the time-independent case. The difference between time-dependent and time-independent algorithms resides solely in the use of $\widetilde{T}$ in place of T. This makes it extremely easy to analyze and devise time-dependent algorithms. Once factorized in terms of $e^{\varepsilon\widetilde{T}} = e^{\varepsilon T}e^{\varepsilon D}$, the operator $e^{\varepsilon D}$ then shifts the time at which all the time-dependent potential to its left must be *evaluated*. This results in Suzuki's rule for solving time-dependent Hamiltonian (2.6): the time-dependent potential must be evaluated at an incremental time from the start of the algorithm equal to the sum of time steps of all the T operators to its right. We will illustrate how this is applied in the next section. For more detailed discussions and examples, see Refs. 14 and 22.

III. FORWARD FOURTH-ORDER ALGORITHMS

In order to solve the radial Schrödinger equation (1.1) efficiently, one must take advantage of its harmonic-oscillator character (1.3). This can be done easily for factorized algorithms because their error terms have a well-defined analytical structure. Consider the second-order factorization,

$$e^{\frac{1}{2}\varepsilon T}e^{\varepsilon V}e^{\frac{1}{2}\varepsilon T} = \exp\varepsilon\left[(T+V) + \frac{1}{24}\varepsilon^2([T,[V,T]] - 2[V,[T,V]]) + O(\varepsilon^4)\right]. \tag{3.1}$$

This is just a general operator equality from applying the Baker-Campbell-Hausdorff (BCH) formula. In the present context, this equality tells us that the second-order factorization on the left-hand side (LHS) deviates from the exact evolution operator $\exp^{\varepsilon(T+V)}$ by error terms which are the double commutators on the right-hand side (RHS). However, for the ordinary harmonic-oscillator Hamiltonian (1.4) with a *constant* spring constant $k=\omega^2$, one can easily verify that

$$[V,[T,V]] = -2\omega^2 V, \tag{3.2}$$

$$[T,[V,T]] = -2\omega^2 T. \tag{3.3}$$

Thus the error terms can be reexpressed in terms of the original operators T and V and be moved back to the LHS to yield,

$$e^{\varepsilon\left(\frac{1}{2}+\frac{1}{24}\omega^2\varepsilon^2\right)T}e^{\varepsilon\left(1-\frac{1}{6}\omega^2\varepsilon^2\right)V}e^{\varepsilon\left(\frac{1}{2}+\frac{1}{24}\omega^2\varepsilon^2\right)T}=e^{\varepsilon(T+V+O(\varepsilon^4))}. \tag{3.4}$$

This means that the LHS is now a fourth-order algorithm for solving the harmonic oscillator. Because of the fundamental identities (3.2) and (3.3), all higher-order commutators for the harmonic oscillator can be subsummed back to T and V yielding the exact factorization[26]

$$e^{\varepsilon C_E T}e^{\varepsilon C_M V}e^{\varepsilon C_E T}=e^{\varepsilon(T+V)}, \tag{3.5}$$

where the "edge" coefficeint C_E and the "middle" coefficient C_M are given by

$$C_E=\frac{1-\cos(\omega\varepsilon)}{\omega\varepsilon\sin(\omega\varepsilon)} \text{ and } C_M=\frac{\sin(\omega\varepsilon)}{\omega\varepsilon}. \tag{3.6}$$

The above discussion only depends on the abstract commutator relations (3.2) and (3.3) and is independent of the specific form of the operator T and V. Thus by interchanging $T\leftrightarrow V$, we can also factorize exactly,

$$e^{\varepsilon(T+V)}=e^{\varepsilon C_E V}e^{\varepsilon C_M T}e^{\varepsilon C_E V}. \tag{3.7}$$

To solve the time-dependent harmonic oscillator, one has to replace $T\rightarrow\tilde{T}$ everywhere. It is easy to verify that for any two time-dependent functions $W(t)$ and $V(t)$,

$$[V(t),[D,W(t)]]=0. \tag{3.8}$$

Hence, the commutator (3.2)

$$[V,[\tilde{T},V]]=[V,[T,V]]=2f(t,E)V \tag{3.9}$$

remains proportional to V. However,

$$[\tilde{T},[V,\tilde{T}]]=2f(t,E)T+2\left[T,\frac{\partial}{\partial t}V\right]-\frac{\partial^2}{\partial t^2}V \tag{3.10}$$

bears no simple relationship to $\tilde{T}$. This means that one can still retain the commutator $[V,[\tilde{T},V]]=[V,[T,V]]$ and move it back to the LHS, but in order to have a fourth-order algorithm, one must eliminate the commutator $[\tilde{T},[V,\tilde{T}]]$ by more elaborated factorization schemes. This coincides precisely with the way forward symplectic algorithms are derived.[12,14,28] For example, the simplest fourth-order forward factorization scheme[12,28] 4B for evolving the system forward for time ε is

$$\begin{aligned}\mathcal{T}_B^{(4)}(\varepsilon)&=e^{a\varepsilon\tilde{T}}e^{\varepsilon\frac{1}{2}V^*}e^{b\varepsilon\tilde{T}}e^{\varepsilon\frac{1}{2}V^*}e^{a\varepsilon\tilde{T}},\\&=e^{a\varepsilon T}e^{\varepsilon\frac{1}{2}V^*(t+a'\varepsilon)}e^{b\varepsilon T}e^{\varepsilon\frac{1}{2}V^*(t+a\varepsilon)}e^{a\varepsilon T},\end{aligned} \tag{3.11}$$

where $a=\frac{1}{2}\left(1-\frac{1}{\sqrt{3}}\right)$, $b=\frac{1}{\sqrt{3}}$, $a'=a+b=\frac{1}{2}\left(1+\frac{1}{\sqrt{3}}\right)$, and the effective potential operator is given by

$$\begin{aligned}V^*(t)&=V(t)+\frac{1}{24}(2-\sqrt{3})\varepsilon^2[V(t),[\tilde{T},V(t)]]\\&=\left[1+\frac{1}{12}(2-\sqrt{3})\varepsilon^2 f(t,E)\right]V(t).\end{aligned} \tag{3.12}$$

This results in the use of an effective time-dependent force

$$F^*(t,E)q=\left[1+\frac{1}{12}(2-\sqrt{3})\varepsilon^2 f(t,E)\right]f(t,E)q, \tag{3.13}$$

which is no more difficult to evaluate than the original. Factorization scheme (3.11) translates into the following fourth-order algorithm for solving the time-dependent harmonic oscillator:

$$\begin{aligned}q_1&=q_0+a\varepsilon p_0\\p_1&=p_0+\frac{1}{2}\varepsilon F^*(t+a\varepsilon)q_1\\q_2&=q_1+b\varepsilon p_1\\p_2&=p_1+\frac{1}{2}\varepsilon F^*(t+a'\varepsilon)q_2\\q_3&=q_2+a\varepsilon p_2\end{aligned} \tag{3.14}$$

The last numbered p and q are the updated values. In the present context, since q is the wave function and p is only an ancillary variable, we will be interested only in algorithms that begin and end with q. These *position*-type algorithms make full use of force evaluations at intermediate time to update the final position. As will be discussed in the next section, this point is important for Killingbeck's method of iterating the last position $q(0,E)$ to zero.

In general, the commutator

$$[V,[T,V]]=2F_j\frac{\partial F_i}{\partial q_j}\frac{\partial}{\partial p_i}=\nabla_i(|\mathbf{F}|^2)\frac{\partial}{\partial p_i} \tag{3.15}$$

produces a force which is the gradient of the square of the original force. For the 1D harmonic oscillator, this is simply $\partial(fq)^2/\partial q=2f^2q$. By incorporating the force gradient, algorithm 4B (3.14) is fourth order with only *two* evaluations of the effective force (3.13).

For three force evaluations, one can use algorithm 4C:[12]

$$\begin{aligned}\mathcal{T}_C^{(4)}(\varepsilon)\equiv{}&e^{\frac{1}{6}\varepsilon T}e^{\frac{3}{8}\varepsilon V+\frac{\alpha}{192}\varepsilon^3 U}e^{\frac{1}{3}\varepsilon T}e^{\frac{1}{4}\varepsilon V+\frac{(1-2\alpha)}{192}\varepsilon^3 U}\\&\times e^{\frac{1}{3}\varepsilon T}e^{\frac{3}{8}\varepsilon V+\frac{\alpha}{192}\varepsilon^3 U}e^{\frac{1}{6}\varepsilon T},\end{aligned} \tag{3.16}$$

where $U\equiv[V,[T,V]]$. One is free to distribute the commutator term symmetrically via α without affecting its fourth-order convergence. The three obvious choices are $\alpha=0$, $3/8$, and $1/2$. The first and the last case concentrate the gradient term at the center and at the two sides, respectively. The second case distributes the gradient term in the same proportion as the original force so that the same effective force

$$F^*(t,E)q=\left[1+\frac{1}{96}\varepsilon^2 f(t,E)\right]f(t,E)q, \tag{3.17}$$

is evaluated at three different times. This is a direct generalization of algorithm 4B. We shall refer to these three variants as 4C, 4C′, and 4C″. For any specific application, one can fine-tune α to minimize, or even eliminate, the algorithm's fourth-order step-size error. We shall refer to this optimized case as $4C_{opt}$. Other forward, or just gradient-based algorithms, can be found in Refs. 12, 14, and 22–24.

IV. KILLINGBECK'S BACKWARD ITERATION

Killingbeck's method[27] for solving the eigenvalue-function pair requires no wave-function matching and can be highly automated. It consists of two key steps: (1) backward integration to ensure numerical stability and (2) quadratic energy convergence via Newton-Ralphson iterations. One begins with an initial guess of the eigenvalue $E^{(0)}$ and chooses a large time value T (large R in the original problem) to set $q(T)=0$ and $p(T)=p_\infty$, where p_∞ is an arbitrary but small number. One then iterates the algorithm, such as (3.14), backward in time to $t=0$. [In practice, it may be simpler to run the algorithm forward in time and change the potential from $V(t)$ to $V(T-t)$.] If E is a correct eigenvalue, then it must satisfy the eigencondition

$$q(0,E)=0. \tag{4.1}$$

Thus the eigenvalue E is a root of the above equation and can be solved by Newton-Ralphson iterations:

$$E^{(n+1)}=E^{(n)}-\frac{q(0,E^{(n)})}{q'(0,E^{(n)})}. \tag{4.2}$$

Killingbeck suggested that the derivative $q'(0,E)=\partial q(0,E)/\partial E$, which obeys

$$\frac{d^2q'(t)}{dt^2}=f(t,E)q'(t)-2q(t), \tag{4.3}$$

can be solved simultaneously with $q(t)$, i.e., differentiating any algorithm, such as (3.14), line by line with respect to E. The resulting algorithm can be iterated at the same time to determine both $q(0,E)$ and $q'(0,E)$ simultaneously so that (4.2) can be updated directly. By rerunning the algorithm with the updated energy, the procedure can be repeated until convergence. The convergence is quadratic in the number of iterations. The converged eigenvalue (and eigenfunction) will deviate from the exact value in powers of ε^n depending on the order of the algorithm used. In solving the radial Schrödinger equation, $t=0$ (i.e., $r=0$) is the absolute boundary and $f(t,E)$ is not defined for $t<0$. Thus in applying Killingbeck's method, one must not use any algorithm which evaluate the force at an intermediate time greater than $t+\varepsilon$.

V. RESULTS FOR SINGULAR POTENTIALS

One important application of solving the radial Schrödinger equation is in atomic (e.g, density functional) calculations, where the dominant interaction is the Coulomb potential

$$V(r)=-\frac{1}{r}. \tag{5.1}$$

As a prototype test case, we show the convergence of various algorithms in solving for the ground state of (5.1) in Fig. 1. We use $T=26$ ($R=26$); beyond $T=25$, there is no change in the eigenvalue on the order of 10^{-12}. For an initial guess of $E^{(0)}=-0.6$, the Killingbeck iteration converges to 12 decimal places in about six iterations. For $E^{(0)}=-1$, -2, -3, -4, and -5, the number of iterations required for convergence are, respectively, 11, 17, 20, 23, and 26. For an initial guess that is off by an order of magnitude, the iteration still converges. In most cases, once a good guess is found, only a few iterations are necessary.

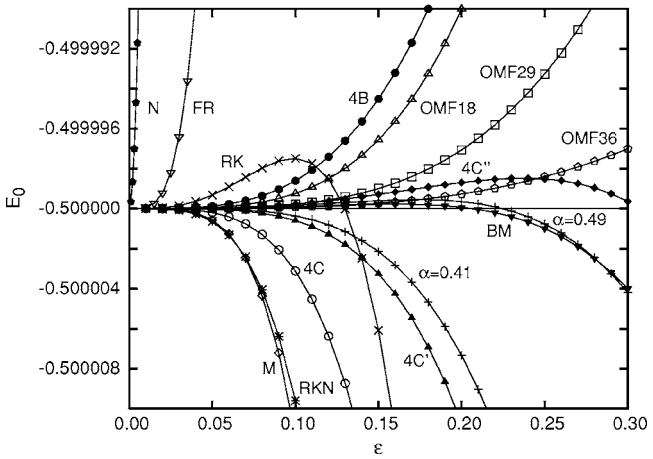

FIG. 1. The convergence of various fourth-order algorithms in solving for the ground-state energy of the Coulomb potential. The solid lines only connect data points to guide the eye. See text for identification of each algorithm.

It is well known that when the Numerov (N) algorithm[1] is used in Killingbeck's method, the Coubomb ground state only converges quadratically.[29] While the reason for this is understood[30] and a simple remedy is available,[31] most of the self-starting fourth-order algorithms used here suffered no such order reduction. Most can be well fitted by the power law $E(\varepsilon)-E_0=c\varepsilon^4$. RKN and RK are the three and four force-evaluation Runge-Kutta-Nystrom and Runge-Kutta algorithms,[6] respectively. FR is the Forest-Ruth[32] symplectic algorithm which uses three force evaluations. This is the first fourth-order symplectic algorithm found and is well known for its relative large error. M is McLachlan's improved fourth-order algorithm[33] which uses four force evaluations. BM is Blanes and Moan's latest[34] refined fourth-order algorithm which uses *six* force evaluations.

FR, M, and BM are examples of conventional symplectic algorithms which have negative intermediate time steps. As shown in Fig. 1, algorithm 4B, which uses only *two* evaluations of the effective force, outperforms all the aforementioned algorithms except BM regardless of the number of force evaluation. OMF18, OMF29, and OMF36 are Omelyan, Mryglod, and Folk's listed[24] fourth-order algorithms 18, 29, and 36. These are gradient algorithms, similar to 4B and 4C′, which use three, four, and five effective force evaluations, respectively. As α is varied from 0 to 0.5, the error of the general 4C algorithm changes from negative to positive. At $\alpha=0.49$, the error curve resembles that of BM. At the optimal value of $\alpha=0.41$, the fourth-order error should have been nearly eliminated with the algorithm showing sixth-order convergence. The fact that it does not will be discussed in the next section. We fitted all the results in Fig. 1 via a power law of the form $E(\varepsilon)-E_0=c\varepsilon^n$ to verify their order of convergence. All can be well fitted with $n=4$ except 4B and $4C_{\text{opt}}$ at $\alpha=0.41$. For 4B, $n\approx 3.5$ and for $4C_{\text{opt}}$, $n\approx 4.5$. Why algorithm 4B should suffer such an order reduction is not understood. It is possible that for 4B, its power-

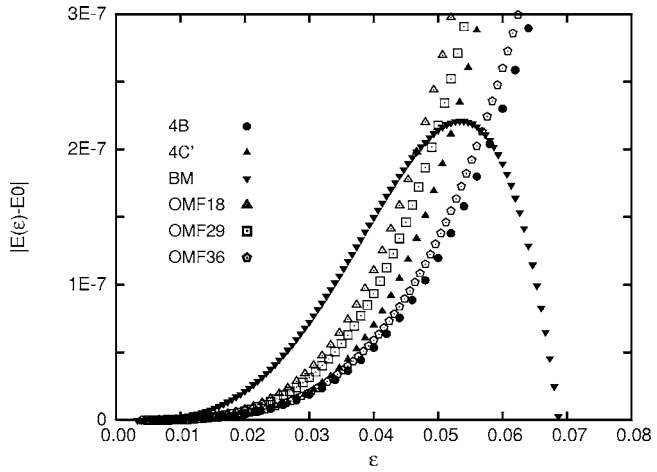

FIG. 2. Equal computational effort comparison of selected algorithms in solving for the Coulomb ground-state energy. See text for discussion.

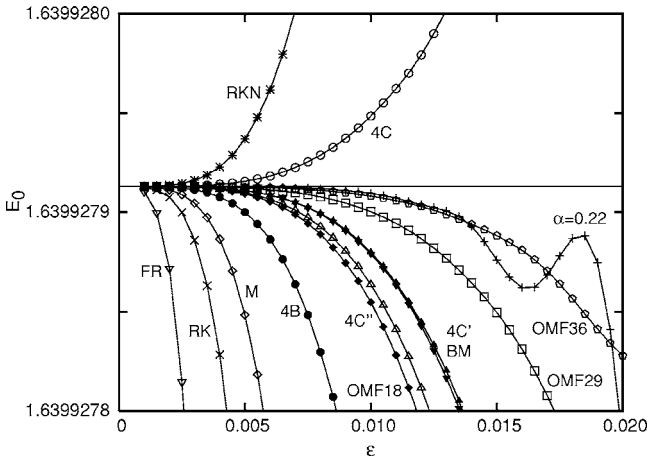

FIG. 3. The convergence of various fourth-order algorithms in solving for the ground-state energy of the spiked harmonic oscillation (5.2) with $M=6$ and $\lambda=0.001$. Same plotting symbols are used to designate the same algorithm compared in Fig. 1.

law behavior only sets in at smaller ε. The case of $4C_{opt}$ will be discussed in the next section.

Since algorithm OMF29 uses four effective force evaluations, one can run algorithm 4B twice at half the time step size. Thus one should compare OMF29 with $4B(\varepsilon/2)$, or OMF29(2ε) with $4B(\varepsilon)$. Thus relative to the computational effort of $4B(\varepsilon)$, one should compare $4C'(1.5\varepsilon)$, OMF18(1.5ε), OMF29(2ε), OMF36(2.5ε), and BM(3ε). This comparison is shown in Fig. 2. In this equal effort comparison, algorithm 4B's fourth-order error is as small as, if not smaller than, all the other gradient algorithm's error. This illustrates the case that efficiency is not necessarily enhanced by increasing the number of force evaluations. Also, all gradient algorithms have errors smaller than that of BM despite fewer force evaluations. We conclude that in solving the Coulomb ground state, the efficiency of algorithms 4B and $4C'$ is unsurpassed by any other algorithms except by the tunable 4C algorithm.

In the first column of Table I, we list the energy obtained by all the algorithms at $\varepsilon=0.01$ weighted by their number of force evaluations. Algorithms 4B and $4C'$ indeed turn in the best result and are outperformed only by $4C_{opt}$ at $\alpha=0.41$. For more accurate results, one can just reduce ε.

As a more stringent test of our algorithms and Killingbeck's method, we next consider the spiked harmonic oscillator (SHO) with potential

$$V(r)=\frac{1}{2}\left(r^2+\frac{\lambda}{r^M}\right). \tag{5.2}$$

For extensive references and discussion on SHO, see Refs. 35–38. Figure 3 shows the convergence of the ground-state energy for the well-studied case[36–38] of $M=6$ with $\lambda=0.001$. For $T=10$ and a reasonable initial guess of $E^{(0)}=1.5$, only five or less iterations are needed for the energy to converge to 12 decimal places. For very incorrect guesses such as $E^{(0)}=1.0$, -1.0, and -3.0, the number of iterations required are only 7, 10, and 13, respectively. The convergence is very robust. (The required number of iterations for the same incorrect initial guess is nearly identical for all the different M cases considered below.) For such a singular potential, the convergent step size has to be much smaller, but surprisingly, only a magnitude smaller. Despite the high degree of singularity, nearly all algorithms remained fourth order and none is downgraded to a lower order. At a glance, all gradient-

TABLE I. Equal computational effort comparison of all fourth-order algorithms.

	Coulomb $\varepsilon=0.01$	SHO (M=6) $\varepsilon=0.001$	SHO (M=4) $\varepsilon=0.001$	SHO (M=5/2) $\varepsilon=0.0002$
$4B(\varepsilon)$	−0.499 999 999 68	1.639 927 912 94	1.534 381 583 86	1.502 005 640
$FR(1.5\varepsilon)$	−0.499 999 735 51	1.639 927 899 76	1.534 381 082 57	1.502 005 464
$RKN(1.5\varepsilon)$	−0.500 000 005 43	1.639 927 913 16	1.534 381 596 96	1.502 005 154
$4C'(1.5\varepsilon)$	−0.500 000 000 20	1.639 927 912 94	1.534 381 584 17	1.502 005 637
$4C_{opt}(1.5\varepsilon)$	−0.500 000 000 05	1.639 927 912 96	1.534 381 585 29	1.502 005 613
$OMF18(1.5\varepsilon)$	−0.499 999 999 43	1.639 927 912 94	1.534 381 583 51	1.502 005 644
$RK(2\varepsilon)$	−0.499 999 961 58	1.639 927 907 62	1.534 381 176 25	1.502 004 936
$M(2\varepsilon)$	−0.500 000 009 24	1.639 927 911 31	1.534 381 453 29	1.502 005 456
$OMF29(2\varepsilon)$	−0.499 999 999 52	1.639 927 912 94	1.534 381 583 87	1.502 005 640
$OMF36(2.5\varepsilon)$	−0.499 999 999 67	1.639 927 912 94	1.534 381 583 89	1.502 005 579
$BM(3\varepsilon)$	−0.499 999 997 91	1.639 927 912 69	1.534 381 564 00	1.502 005 496
"Exact"[a]	−0.500 000 000 00	1.639 927 912 96	1.534 381 585 45	1.502 005 626

[a]References 36–38.

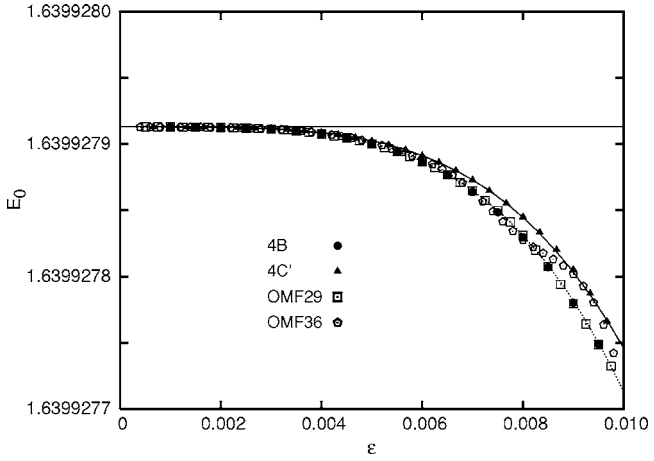

FIG. 4. Equal effort comparison of various fourth-order gradient symplectic algorithms in solving for the ground-state energy of the spiked harmonic oscillator of Fig. 3.

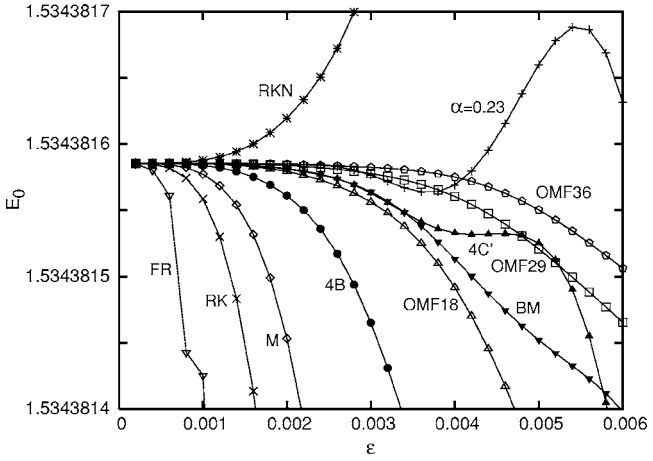

FIG. 5. The convergence of various fourth-order algorithms in solving for the ground-state energy of the spiked harmonic oscillation (5.2) with $M=4$ and $\lambda=0.001$.

based algorithms converge better than nongradient algorithms. Even BM is no better than algorithm 4C′. Since 4C and 4C″ have errors of opposite sign, one can again vary α to minimize the fourth-order error. The optimized algorithm at $\alpha=0.22$ converges better than all other algorithms regardless of the number of force evaluations and can be best fitted by a *fifth-order* power law.

To compare the efficiency of gradient algorithms, we again normalize each algorithm to the computation effort of 4B. In Fig. 4, we plot the convergence curve of $4B(\varepsilon)$, $4C'(1.5\varepsilon)$, $OMF29(2\varepsilon)$, and $OMF36(2.5\varepsilon)$. The solid line is the fourth-order monomial $E(\varepsilon)-E_0=c\varepsilon^4$ which goes through 4C′'s result with $c=16.7$. The other three algorithms can be fitted with the dotted line with $c=20.0$. Thus all gradient algorithms are essentially similar, with 4C′ marginally better. Again, algorithms OME29 and OME36, which use four and five effective force evaluations with complex numeric coefficients, are not more efficient than the simpler algorithms 4B and 4C′ with analytical coefficients.

At $\varepsilon=0.001$, the weighted result of each algorithm is given in the second column of Table I. All are in excellent agreement with the value found in the literature.[36–38] At this step size, only gradient algorithms are accurate to ten or more decimal places. For even greater accuracy, one can simply reduce ε.

The algorithms are equally effective in the case of $M=4$ and $\lambda=0.001$. This is shown in Fig. 5. All algorithms showed fourth-order convergence, except for $4C_{\text{opt}}$, which can be better fitted with a fifth-order power law. Their energy values at $\varepsilon=0.001$ are listed in the third column of Table I. The optimized 4C algorithm is accurate to nine decimal places. Note that once algorithm 4C is optimized for $M=6$, it can also be used for $M=4$. The change in α's value is slight.

In the most difficult, "supersingular" case of $M=5/2$, with λ remained small at 0.001, the power-law behavior seems to require $\varepsilon<10^{-5}$. This is shown in Fig. 6. (If λ were not too small, such as 0.1 or 0.01, the power-law behavior would remain observable in the range of ε considered.) The energy obtained at $\varepsilon=0.0002$ is listed in the fourth column of Table I. The variable 4C algorithm uses $\alpha=0.23$ inherited from the $M=4$ case. All algorithms are less efficient in dealing with this "supersingular" case, but gradient algorithms can still maintain an eight-digit accuracy.

VI. HIGHER-ORDER ALGORITHMS

In general, if $\mathcal{T}_A$ is a left-right symmetric approximation of the short time evolution operator $e^{\varepsilon(T+V)}$,

$$\mathcal{T}_A=\prod_{i=1}^{N} e^{t_i\varepsilon T}e^{v_i\varepsilon V}=e^{\varepsilon H_A}, \tag{6.1}$$

such that $\Sigma_i^N t_i=1$ and $\Sigma_i^N v_i=1$, then the approximate Hamiltonian operator is of the form

$$\begin{aligned}H_A=T+V&+\varepsilon^2(e_{TTV}[T^2V]+e_{VTV}[VTV])\\&+\varepsilon^4(e_{TTTTV}[TT^3V]+e_{VTTTV}[VT^3V]\\&+e_{TTVTV}[TTVTV]+e_{VTVTV}[VTVTV])+\cdots,\end{aligned} \tag{6.2}$$

where e_{TTV}, e_{VTTTV}, etc., are coefficients specific to a particular algorithm and where we have used the condensed commutator notation $[T^2V]\equiv[T,[T,V]]$, etc. Symmetric factor-

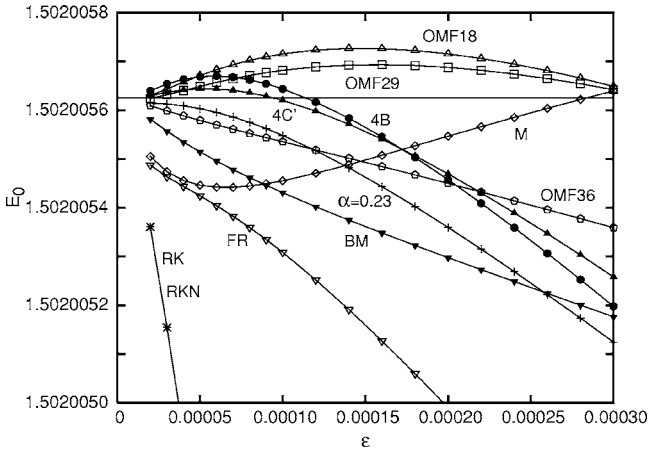

FIG. 6. The convergence of various fourth-order algorithms in solving for the ground-state energy of the spiked harmonic oscillation (5.2) with $M=5/2$ and $\lambda=0.001$. In this "supersingular" case, the power-law behavior is not observed within the range of ε considered.

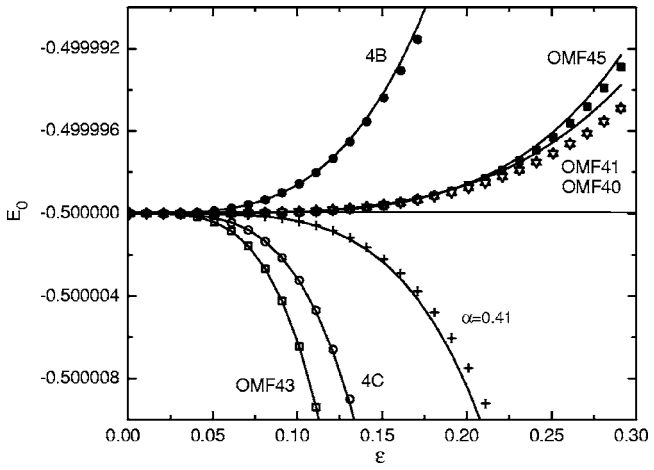

FIG. 7. The convergence of various fourth- and sixth-order gradient algorithms in solving for the ground-state energy of the Coulomb potential. The solid lines here are fitted power laws power of 3.5 (4B), 4 (4C, OMF40, OMF41, OMF43), and 4.5 (4C with $\alpha=0.41$, OMF45). None is showing sixth-order convergence.

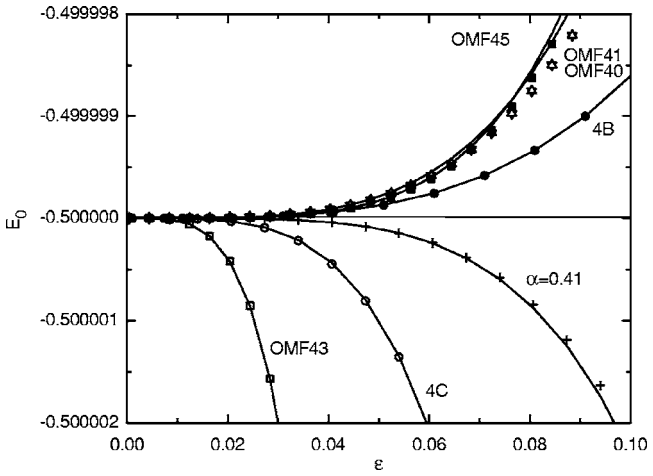

FIG. 8. Equal effort comparison in solving for the ground-state energy of the Coulomb potential. The OMF algorithms are nominally sixth-order algorithms. However, their convergence is no better than that of algorithm 4C with $\alpha=0.41$.

izations give rise to time-reversible algorithms and have only even-order error terms. For a constant $k=\omega^2$, the fundamental identity (3.3) implies that $[TT^3V]=0$ and $[VT^3V]=0$. This crucial simplification is no longer true in the time-dependent case when T is replaced by $\tilde{T}$. From this perspective, one can understand why the ability to integrate the time-independent harmonic oscillator exactly does not help in solving the time-dependent case. Exponentially or sinusoidally fitted algorithms are therefore not necessarily more efficient. In the time-dependent case, the problem is fundamentally different because some commutators no longer vanish. Note that the commutators

$$[V\tilde{T}V\tilde{T}V]=4f^2(t)V \tag{6.3}$$

can be moved back to the LHS. However, the saving here is marginal since this error term can be easily eliminated by incorporating more gradient terms $[VTV]$ in the factorization process.

It has been shown[39] that in order to derived a general sixth-order forward algorithm, one must retain both $[VTV]$ and $[VT^3V]$ in the factorization process. Unfortunately, since $[VT^3V]$ cannot be evaluated easily, there is currently no practical sixth-order forward algorithm. However, Omelyan *et al.*[24] have derived a number sixth-order gradient-based, but nonforward, algorithms of the form

$$\mathcal{T}_B^{(6)}(\varepsilon) \equiv \cdots e^{\varepsilon(v_0V+\varepsilon^2u_0U)}e^{\varepsilon t_1T}e^{\varepsilon(v_1V+\varepsilon^2u_1U)} \times e^{\varepsilon t_2T}e^{\varepsilon(v_2V+\varepsilon^2u_2U)}e^{\varepsilon t_3T}. \tag{6.4}$$

Since the factorization is left-right symmetric, we only list the operators from the center to the right. These sixth-order algorithms all require a mixture of five force or effective force evaluations. Figure 7 shows the convergence of four of their position-type, sixth-order algorithms OMF40, OMF41, OMF43, and OMF45 when solving the Coulomb potential. None exhibited sixth-order convergence. The best is OMF45, which converges with a power of 4.5, same as the optimized, supposedly sixth-order algorithm $4C_{\text{opt}}$ with $\alpha=0.41$. Why these sixth-order algorithms are so downgraded in the Coulomb case is not understood.

In Fig. 8, we compare all algorithms on an equal effort basis as discussed earlier. In this case, the optimized fourth-order 4C algorithm has the smallest error, even when compared with OMF's sixth-order algorithms.

In Fig. 9, we show the convergence of these four sixth-order algorithms in solving for the ground-state energy of the spiked harmonic oscillator with $M=6$ and $\lambda=0.001$. All OMF algorithms can now be well fitted with sixth-order power laws as indicated by solid lines. In the case of OMF40 and OMF41, the "glitch" in the convergence curve near $\varepsilon=0.011$ is real. The convergence curve for these two algorithms contains a singular term of the form $\approx 1/(\varepsilon-0.011)$, which blows up near $\varepsilon\approx 0.011$. Why only algorithms OMF40 and OMF41 exhibit such a singular behavior is also not understood.

In Fig. 10, we compare these gradient algorithms in an

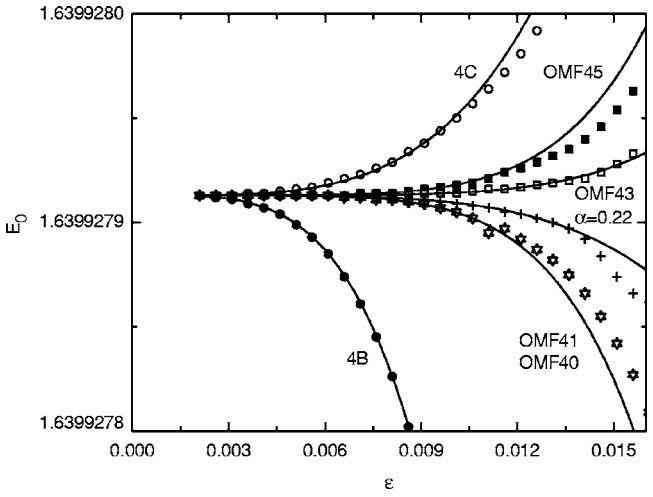

FIG. 9. The convergence of various fourth- and sixth-order algorithms in solving for the ground-state energy of the spiked harmonic oscillator (5.2) with $M=6$ and $\lambda=0.001$. The solid lines are fitted power laws power of 4 (4B, 4C), 5 (4C with $\alpha=0.22$), and 6 (all OMF algorithms).

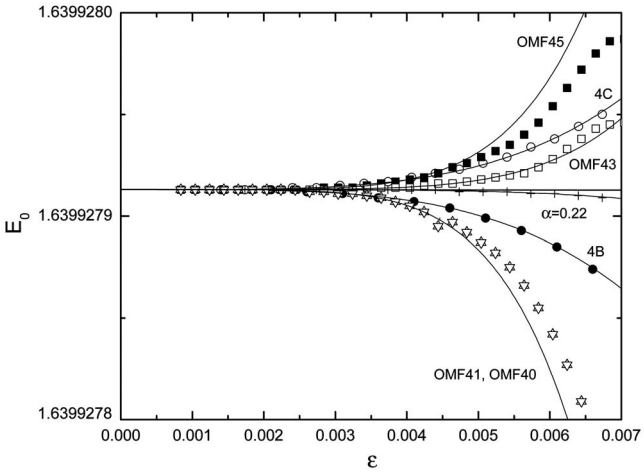

FIG. 10. Equal effort comparison of various fourth- and sixth-order gradient symplectic algorithms in solving for the ground-state energy of the spiked harmonic oscillator of Fig. 9. The optimized algorithm 4C with $\alpha=0.22$ has the smallest error for $\varepsilon \gtrsim 0.002$.

equal effort basis. The convergence range of sixth-order algorithms are not greater than those of fourth-order algorithms. For $\varepsilon \gtrsim 0.002$, the optimized fourth-order algorithm 4C with $\alpha=0.22$ has smaller errors than all the sixth-order algorithms. However, for very high accuracy, sixth-order algorithms are better when ε is very small.

VII. CONCLUSIONS

In this work, by regarding the radial Schrödinger equation as a classical time-dependent force problem, we have shown that the entire literature of symplectic integrators can be used to find its solution. Among symplectic integrators, factorized algorithms are favored because Suzuki's rule can be applied easily to solve the time-dependent force problem. Among factorized algorithms, gradient or forward algorithms are particularly suited because they take advantage of the harmonic character of the Schrödinger equation. We demonstrated the unique effectiveness of fourth-order gradient symplectic algorithms in solving the radial Schrödinger equation via Killingbeck's backward iteration. Even for very singular potentials, these algorithms are highly effective in computing the eigenvalue-function pair. There is also no difficulty in obtaining excited states. These gradient algorithms can form the core basis for solving nonlinear Schrödinger equations such as the Hartree-Fock and the Kohn-Sham equations. However, due to the unique identification of the one-dimensional spatial coordinate as time, the current method does not appear to be generalizable to higher dimension for solving the general Schrödinger equation in two or three dimension.

Among gradient algorithms, algorithm 4C with a tunable parameter α is the most efficient in solving a variety of different potentials. Despite the fact that there are more complex fourth- or sixth-order algorithms which use more effective force evaluations, none are really better than 4C. More force evaluations do not necessarily enhance the efficiency of an algorithm, specially in solving the radial Schrödinger equation.

In solving the Coulomb potential, some gradient algorithms are downgraded to lower order while others are not. Even more surprising is the fact that none of the sixth-order algorithms exhibited sixth-order convergence in the range of ε considered. These findings are not understood and should be studied further.

By regarding the radial Schrödinger equation as a classical dynamical problem, one can now use the same set of symplectic algorithms for solving both classical and quantum-mechanical problems.

ACKNOWLEDGMENTS

This work was supported, in part, by a National Science Foundation Grant (to SAC) No. DMS-0310580 and ONR Grant No. N00014-04-1-0336 to Marlan Scully.

[1] D. R. Hartree, *The Calculation of Atomic Structures* (Pergamon, London, 1957).
[2] R. Chen, Z. Xu and L. Sun, Phys. Rev. E **47**, 3799 (1993).
[3] A. Raptis and A. C. Allison, Comput. Phys. Commun. **14**, 1 (1978).
[4] T. E. Simos, IMA J. Numer. Anal. **21**, 919 (2001).
[5] H. Van de Vyver, Comput. Phys. Commun. **166**, 109 (2005).
[6] R. H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics*, revised edition (AIAA, Reston, VA, 1999).
[7] A. J. Dragt and J. M. Finn, J. Math. Phys. **17**, 2215 (1976).
[8] H. Yoshida, Celest. Mech. 56, 27 (1993).
[9] R. I. McLachlan and G. R. W. Quispel, Acta Numerica **11**, 241 (2002).
[10] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration* (Springer-Verlag, Berlin, 2002).
[11] B. Gladman, M. Duncan, and J. Candy, Celest. Mech. Dyn. Astron. **52**, 221 (1991).
[12] S. A. Chin, Phys. Lett. A **226**, 344 (1997).
[13] S. A. Chin and D. W. Kidwell, Phys. Rev. E **62**, 8746 (2000).
[14] S. A. Chin and C. R. Chen, Celest. Mech. Dyn. Astron. **91**, 301 (2005).
[15] S. Scuro and S. A. Chin, Phys. Rev. E **71**, 056703 (2005).
[16] X. S. Liu, X. Y. Liu, Z. Y. Zhou, P. Z. Ding, and S. F. Pan, Int. J. Quantum Chem. **79**, 343 (2000).
[17] Z. Kalogiratou, Th. Monovasilis, and T. E. Simos, J. Comput. Appl. Math. **158**, 83 (2003).
[18] J. Wench, M. Däne, W. Hergert, and A. Ernst, Comput. Phys. Commun. **160**, 129 (2004).
[19] D. Baye, G. Goldstein, and P. Capel, Phys. Lett. A **317**, 337 (2003).
[20] G. Goldstein and D. Baye, Phys. Rev. E **70**, 056703 (2004).
[21] M. Suzuki, Proc. Jpn. Acad., Ser. B: Phys. Biol. Sci. **69**, 161 (1993).
[22] S. A. Chin and C. R. Chen, J. Chem. Phys. **117**, 1409 (2002).
[23] I. P. Omelyan, I. M. Mryglod, and R. Folk, Phys. Rev. E **66**, 026701 (2002).
[24] I. P. Omelyan, I. M. Mryglod, and R. Folk, Comput. Phys. Commun. **151**, 272 (2003).
[25] S. A. Chin and C. R. Chen, J. Chem. Phys. **114**, 7338 (2001).
[26] S. A. Chin and E. Krotscheck, Phys. Rev. E **72**, 036705 (2005).
[27] J. Killingbeck, J. Phys. A **18**, 245 (1985).
[28] M. Suzuki, *Computer Simulation Studies in Condensed Matter Physics VIII*, edited by D. Landau, K. Mon, and H. Shuttler (Springer, Berlin, 1996).
[29] H. W. Crater and G. W. Redden, J. Comput. Phys. **19**, 236 (1975).
[30] B. R. Johnson, J. Chem. Phys. **69**, 4676 (1978).
[31] E. Buendia and R. Guardiola, J. Comput. Phys. **60**, 561 (1985).
[32] E. Forest and R. D. Ruth, Physica D **43**, 105 (1990).
[33] R. I. McLachlan, SIAM J. Sci. Comput. (USA) **16**, 151 (1995).
[34] S. Blanes and P. C. Moan, as quoted in Ref. 9, p. 407, algorithm 1(a).
[35] J. P. Killingbeck, G. Jolicard, and A. Grosjean, J. Phys. A **34**, L367 (2001).
[36] A. K. Roy, Phys. Lett. A **321**, 231 (2004).
[37] V. C. Aguilera-Navarro, G. A. Estévez, and R. Guardiola, J. Math. Phys. **31**, 99 (1990).
[38] E. Buendía, F. J. Gálvez, and A. Puertas, J. Phys. A **28**, 6731 (1995).
[39] S. A. Chin, Phys. Rev. E **71**, 016703 (2005).