Complete characterization of fourth-order symplectic integrators with extended-linear coefficients

Siu A. Chin
Department of Physics, Texas A&M University, College Station, Texas 77843, USA
(Received 8 November 2005; published 17 February 2006)

The structure of symplectic integrators up to fourth order can be completely and analytically understood when the factorization (split) coefficients are related linearly but with a uniform nonlinear proportional factor. The analytic form of these *extended-linear* symplectic integrators greatly simplified proofs of their general properties and allowed easy construction of both forward and nonforward fourth-order algorithms with an arbitrary number of operators. Most fourth-order forward integrators can now be derived analytically from this extended-linear formulation without the use of symbolic algebra.

DOI: 10.1103/PhysRevE.73.026705 PACS number(s): 02.70.Ss, 02.70.Ns, 95.10.Ce, 05.10.−a

I. INTRODUCTION

Evolution equations of the form

$$w(t+\varepsilon)=e^{\varepsilon(T+V)}w(t), \tag{1.1}$$

where T and V are noncommuting operators, are fundamental to all fields of physics ranging from classical mechanics [1–5], electrodynamics [6,7], and statistical mechanics [8–11] to quantum mechanics [12–14]. All can be solved by approximating $e^{\varepsilon(T+V)}$ to the $(n+1)$th order in the product form

$$e^{\varepsilon(T+V)}=\prod_{i=1}^{N}e^{t_i\varepsilon T}e^{v_i\varepsilon V}+O(\varepsilon^{n+1}) \tag{1.2}$$

via a well chosen set of factorization (or split) coefficients $\{t_i\}$ and $\{v_i\}$. The resulting algorithm is then nth order because the algorithm's Hamiltonian is $T+V+O(\varepsilon^n)$. By understanding this single approximation, computational problems in diverse fields of physics can all be solved by applying the same algorithm.

Classically, (1.2) results in a class of composed or factorized symplectic integrators. While the conditions on $\{t_i\}$ and $\{v_i\}$ for producing an nth order algorithm can be stated, these *order conditions* are highly nonlinear and analytically opaque. In many cases [14–17], elaborate symbolic mathematical programs are needed to produce even fairly low order algorithms if N is large. In this work, we show that the structure of most fourth-order algorithms, including nearly all known forward ($\{t_i,v_i\}>0$) integrators, can be understood and derived on the basis that $\{v_i\}$ and $\{t_i\}$ are linearly related but with a uniform nonlinear proportional factor. This class of *extended-linear integrators* is sufficiently complex to be representative of symplectic algorithms in general, but its transparent structure makes it invaluable for constructing integrators up to the fourth-order. In this work we prove three important theorems, on the basis of which many families of fourth-order algorithms can be derived with analytically known coefficients, including all known forward integrators up to $N=4$.

II. THE ERROR COEFFICIENTS

The product form (1.2) has the general expansion

$$\prod_{i=1}^{N}e^{t_i\varepsilon T}e^{v_i\varepsilon V}=\exp\{\varepsilon e_T T+\varepsilon e_V V+\varepsilon^2 e_{TV}[T,V]+\varepsilon^3 e_{TTV}[T,[T,V]]+\varepsilon^3 e_{VTV}[V,[T,V]]+\cdots\}. \tag{2.1}$$

We have previously [18] described in detail how the error coefficients e_T, e_V, e_{TV}, e_{TTV}, and e_{VTV} can be determined from $\{t_i\}$ and $\{v_i\}$:

$$e_T=\sum_{i=1}^{N}t_i,\quad e_V=\sum_{i=1}^{N}v_i, \tag{2.2}$$

$$\tfrac{1}{2}+e_{TV}=\sum_{i=1}^{N}\nabla s_i u_i, \tag{2.3}$$

$$\frac{1}{3!}+\frac{1}{2}e_{TV}+e_{TTV}=\frac{1}{2}\sum_{i=1}^{N}\nabla s_i^2 u_i, \tag{2.4}$$

$$\frac{1}{3!}+\frac{1}{2}e_{TV}-e_{VTV}=\frac{1}{2}\sum_{i=1}^{N}\nabla s_i u_i^2, \tag{2.5}$$

where we have defined useful variables

$$s_i=\sum_{j=1}^{i}t_j,\quad u_i=\sum_{j=i}^{N}v_j, \tag{2.6}$$

with $s_0=0$, $u_{N+1}=0$, and the backward finite differences

$$\nabla s_i^n=s_i^n-s_{i-1}^n, \tag{2.7}$$

with property

$$\sum_{i=1}^{N}\nabla s_i^n=s_N^n(=e_T^n=1). \tag{2.8}$$

We will always assume that the primary constraint $e_T=1$ and $e_V=1$ are satisfied so that (2.8) sums to unity. Satisfying these two primary constraints is sufficient to produce a first-order algorithm. For a second-order algorithm, one must additionally force $e_{TV}=0$. For a third-order algorithm, one further requires that $e_{TTV}=0$ and $e_{VTV}=0$. For a fourth-order algorithm, it is sufficient to satisfy the third-order constraints with coefficients t_i that are left-right symmetric. (The symmetry for v_i will follow and need not be imposed *a priori*.) Once the primary conditions $e_T=1$ and $e_V=1$ are imposed, the constraints equations (2.3)–(2.5) are highly nonlinear and difficult to decipher analytically. In this work, we will show that (2.3) can be satisfied for all N by having $\{v_i\}$ linearly related to $\{t_i\}$ (or vice versa). The coefficients e_{TTV} and e_{VTV} can then be evaluated simply in terms of $\{t_i\}$ (or $\{v_i\}$) alone. This then completely determines the structure of third- and fourth-order algorithms.

III. THE EXTENDED-LINEAR FORMULATION

The constraint $e_{TV}=0$ is satisfied if

$$\sum_{i=1}^{N} \nabla s_i u_i = \tfrac{1}{2}. \tag{3.1}$$

If we view $\{t_i\}$ as given, this is a linear equation for $\{u_i\}$. Knowing (2.8), a general solution for u_i in terms of s_i and s_{i-1} is

$$u_i = \sum_{n=1}^{M} C_n \frac{\nabla s_i^n}{\nabla s_i}, \quad \text{with} \sum_{n=1}^{M} C_n = \tfrac{1}{2}. \tag{3.2}$$

The coefficients C_n represent the intrinsic freedom in $\{v_i\}$ to satisfy any constraint as expressed through its relationship to $\{t_i\}$. The expansion (3.2) is in increasing powers of s_i and s_{i-1}. If we truncated the expansion at $M=2$, then for $i\neq 1$, u_i is linearly related to $\{s_i\}$, i.e.,

$$u_i = C_1 + C_2\frac{\nabla s_i^2}{\nabla s_i} = C_1 + C_2(s_i + s_{i-1}). \tag{3.3}$$

For $i=1$, since we must satisfy the primary constraint $e_V=1$, we must have

$$u_1 = 1. \tag{3.4}$$

In this case, the constraint (3.1) takes the form

$$\sum_{i=1}^{N} \nabla s_i u_i = t_1 + C_1(1-t_1) + C_2(1-t_1^2) = \tfrac{1}{2}. \tag{3.5}$$

The complication introduced by $u_1=1$, in this, and in other similar sums, can be avoided without any loss of generality by decreeing

$$t_1 = 0, \tag{3.6}$$

so that (3.5) remains

$$C_1 + C_2 = \tfrac{1}{2}. \tag{3.7}$$

For $i\neq 1\neq N$, (3.3) implies that

$$v_i = -C_2(t_i + t_{i+1}). \tag{3.8}$$

Since $v_1=u_1-u_2=1-C_1-C_2t_2$, by virtue of (3.7),

$$v_1 = \tfrac{1}{2} + C_2(1-t_2). \tag{3.9}$$

Similarly, since $v_N=u_N=C_1+C_2(2-t_N)$, we also have

$$v_N = \tfrac{1}{2} + C_2(1-t_N). \tag{3.10}$$

Given $\{t_i\}$ such that $t_1=0$, the set of $\{v_i\}$ defined by (3.8)–(3.10) automatically satisfies $e_V=1$ and $e_{TV}=0$. If C_2 were a real constant, then $\{v_i\}$ is linearly related to $\{t_i\}$. However, in most cases C_2 will be a function of $\{t_i\}$ and the actual dependence is nonlinear. But the nonlinearity is restricted to C_2, which is the same for all v_i. We will call this special form of dependence of v_i on $\{t_i\}$, *extended-linear*. For a given set of t_i, (3.8)–(3.10) defines our class of extended-linear integrators with one remaining parameter C_2.

For extended-linear integrators as described above, one can easily check that the sums in (2.4) and (2.5) can be evaluated as

$$\sum_{i=1}^{N} \nabla s_i^2 u_i = C_1 + C_2 + gC_2 = \tfrac{1}{2} + gC_2, \tag{3.11}$$

$$\sum_{i=1}^{N} \nabla s_i u_i^2 = (C_1 + C_2)^2 + gC_2^2 = \tfrac{1}{4} + gC_2^2. \tag{3.12}$$

Again the complication introduced by $u_1=1$ is avoided by decreeing $t_1=0$. The quantity g is a frequently occuring sum defined via

$$\sum_{i=1}^{N} \frac{\nabla s_i^2\,\nabla s_i^2}{\nabla s_i} = 1 + g, \tag{3.13}$$

with explicit form

$$g = \sum_{i=1}^{N} s_i s_{i-1}(s_i - s_{i-1}) = \tfrac{1}{3}(1-\delta g), \tag{3.14}$$

where

$$\delta g = \sum_{i=1}^{N} t_i^3. \tag{3.15}$$

Much of the mechanics of dealing with these sums have been worked out in Ref. [18]. However, their use and interpretation here are very different. From (2.4) and (2.5), we have

$$e_{TTV} = \tfrac{1}{12} + \tfrac{1}{2}gC_2, \tag{3.16}$$

$$e_{VTV} = \tfrac{1}{24} - \tfrac{1}{2}gC_2^2. \tag{3.17}$$

Both are now only functions of $\{t_i\}$ through g.

IV. FUNDAMENTAL THEOREMS

We can now prove a number of important results:

Theorem 1. For the class of extended-linear symplectic integrators defined by $t_1=0$ and (3.8)–(3.10), if $\{t_i\}>0$ for $i\neq 1$ such that $e_T=1$, then $e_{TTV}\neq e_{VTV}$.

Proof. Setting $e_{TTV}=e_{VTV}$ produces a quadratic equation for C_2,

$$C_2^2 + C_2 + \frac{1}{12g} = 0 \tag{4.1}$$

whose discriminant

$$D = b^2 - 4ac = -\frac{\delta g}{1-\delta g} \tag{4.2}$$

is strictly negative (since if $e_T=1$, then $1>\delta g>0$). Hence no real solution exists for C_2. This is a fundamental theorem about positive-coefficient factorizations. This was proved generally in the context of symplectic corrector (or process) algorithms by Chin [11] and by Blanes and Casas [19]. If e_{TTV} can never equal e_{VTV}, then no second-order algorithm with positive coefficients can be corrected beyond second order with the use of a corrector.

As a corollary, for $\{t_{i>1}\}>0$, e_{TTV} and e_{VTV} cannot both vanish. This is the content of the Sheng-Suzuki Theorem [20,21]: there are no integrators of an order greater than 2 of the form (2.1) with only positive factorization coefficients. Our proof here is restricted to extended-linear integrators, but can be interpreted more generally as it is done in Ref. [18]. Blanes and Casas [19] have also given elementary proof of this using a very weak *necessary* condition. Here, for extended-linear integrators, we can be very precise in stating how both e_{TTV} and e_{VTV} fail to vanish. We have, from (3.16), if $e_{TTV}=0$, then

$$C_2 = -\frac{1}{2(1-\delta g)}, \quad e_{VTV} = -\frac{1}{24}\frac{\delta g}{(1-\delta g)}. \tag{4.3}$$

Similarly, from (3.17), if $e_{VTV}=0$, then

$$C_2 = -\frac{1}{2\sqrt{1-\delta g}}, \quad e_{TTV} = \tfrac{1}{12}(1-\sqrt{1-\delta g}). \tag{4.4}$$

Satisfying either condition forces C_2 to be a function of $\{t_i\}$ through δg. From Ref. [18], we have learned that the value given by (4.3) is actually an upperbound for e_{VTV} if $\{t_{i>1}\}>0$ and $e_{TTV}=0$. Similarly, in general, the value given by (4.4) is a lower bound for e_{TTV} if $\{t_{i>1}\}>0$ and $e_{VTV}=0$. Our class of extended-linear integrators are all algorithms that attain these bounds for positive $t_{i>1}$. Note that in (4.4) we have discarded the positive solution for C_2 which would have led to negative values for the v_i coefficients.

For the study of forward integrators where one requires $\{t_{i>1}\}>0$, it is useful to state (4.3) as a theorem:

Theorem 2a. For the class of extended-linear symplectic integrators defined by

$$\begin{aligned} v_1 &= \tfrac{1}{2} + C_2(1-t_2), \quad v_N = \tfrac{1}{2} + C_2(1-t_N), \\ v_i &= -C_2(t_i + t_{i+1}), \end{aligned} \tag{4.5}$$

with $t_1=0$, $e_T=1$, and C_2, e_{VTV} given by

$$C_2 = -\frac{1}{2\phi}, \quad e_{VTV} = -\frac{1}{24}\left(\frac{1}{\phi}-1\right), \tag{4.6}$$

where

$$\phi = 1-\delta g \quad \text{and} \quad \delta g = \sum_{i=1}^{N} t_i^3, \tag{4.7}$$

one has

$$\prod_{i=1}^{N} e^{t_i \varepsilon T} e^{v_i \varepsilon V} = \exp\{\varepsilon(T+V) + \varepsilon^3 e_{VTV}[V,[T,V]] + \cdots\}. \tag{4.8}$$

For $t_1=0$, the first operator $e^{v_1\varepsilon V}$ classically updates the velocity (momentum) variable. *Theorem 2a* completely described the structure of these *velocity*-type algorithms.

If one now interchanges $T \leftrightarrow V$ and $\{t_i\}\leftrightarrow\{v_i\}$, then $[T,[T,V]]$ transforms into $[V,[T,V]]$ with a sign change. Hence, one needs to interpret e_{TTV} in (4.4) as $-e_{VTV}$, yielding as follows:

Theorem 2b. For the class of extended-linear symplectic integrators defined by

$$\begin{aligned} t_1 &= \tfrac{1}{2} + C_2(1-v_2), \quad t_N = \tfrac{1}{2} + C_2(1-v_N), \\ t_i &= -C_2(v_i + v_{i+1}), \end{aligned} \tag{4.9}$$

with $v_1=0$, $e_V=1$, and C_2, e_{VTV} given by

$$C_2 = -\frac{1}{2\phi'}, \quad e_{VTV} = -\tfrac{1}{12}(1-\phi'), \tag{4.10}$$

where

$$\phi' = \sqrt{1-\delta g'} \quad \text{and} \quad \delta g' = \sum_{i=1}^{N} v_i^3, \tag{4.11}$$

one has

$$\prod_{i=1}^{N} e^{v_i \varepsilon V} e^{t_i \varepsilon T} = \exp\{\varepsilon(T+V) + \varepsilon^3 e_{VTV}[V,[T,V]] + \cdots\}. \tag{4.12}$$

For $v_1=0$, the first operator $e^{t_1\varepsilon T}$ classically updates the position variable. *Theorem 2b* completely described the structure of these *position*-type algorithms.

In both *Theorems 2a* and *2b*, one obtains fourth-order forward algorithms by simply moving the commutator $[V,[T,V]]$ term back to the left-hand side and distribute it symmetrically among all the V operators [28].

If some t_i were allowed to be negative, then both e_{TTV} and e_{VTV} can be zero for $\delta g=0$. For both (4.3) and (4.4) we have

$$C_2 = -\tfrac{1}{2} \tag{4.13}$$

and

$$v_i = \tfrac{1}{2}(t_i + t_{i+1}). \tag{4.14}$$

The latter is now true even for $i=1$ and $i=N$. This is not a coincident, from (3.16) and (3.17), if we set $C_2=-\frac{1}{2}$, then

$$e_{TTV} = 2e_{VTV} = \frac{1}{12} - \frac{g}{4} = \frac{1}{12}\delta g. \tag{4.15}$$

Since C_2 here is a true constant, $\{v_i\}$ is linearly related to $\{t_i\}$. We can formulate this explicitly as a theorem for the negative-coefficient factorization yielding truly linear algorithms:

Theorem 3. For the class of truly *linear* algorithms defined by

$$v_1 = \tfrac{1}{2}t_2, \quad v_N = \tfrac{1}{2}t_N, \quad v_i = \tfrac{1}{2}(t_i + t_{i+1}), \tag{4.16}$$

and $t_1=0$, one has

$$\prod_{i=1}^{N} e^{t_i\varepsilon T}e^{v_i\varepsilon V} = \exp\left\{ \varepsilon e_T(T+V) + \frac{\varepsilon^3}{24}\delta g(2[T,[T,V]] + [V,[T,V]]) + \cdots \right\}. \tag{4.17}$$

Both commutators now vanish simultaneously if $\delta g=0$.

Theorem 3 can be proven more directly by noting that

$$\mathcal{T}_2(t_i) = e^{1/2t_i\varepsilon V}e^{t_i\varepsilon T}e^{1/2t_i\varepsilon V} = \exp\left[t_i\varepsilon(T+V) + t_i^3\frac{\varepsilon^3}{24}(2[T,[T,V]] + [V,[T,V]]) + O(\varepsilon^5) \right], \tag{4.18}$$

the product $\Pi_{i=2}^{N}\mathcal{T}_2(t_i)$ then reproduces (4.17). This has been derived by Blanes and Casas [19] in a discussion before their *Theorem 5*. However, they were more interested in using *Theorem 3* above to discuss the distribution of negative coefficients than in deriving fourth-order algorithms. For example, an immediate corollary of *Theorem 3* is that if δg were to vanish, then there must be at least one $t_k<0$ such that $t_k^3+t_{k+1}^3<0$ *or* $t_k^3+t_{k-1}^3<0$. Since

$$(x^3+y^3) = (x+y)\left[\tfrac{3}{4}y^2 + \left(x-\tfrac{1}{2}y\right)^2\right],$$

$x^3+y^3<0 \Rightarrow x+y<0$. We therefore must have $t_k+t_{k+1}<0$ or $t_k+t_{k-1}<0$. From (4.16), this implies that v_k or v_{k-1} must be negative. Thus an algorithm of order greater than 2 of the form (4.17) must contain at least one pair of negative coefficients t_i and v_j. In its general context, this is the Goldman-Kaper result [22]. Our linear formulation here, as well as Blanes and Casas' *Theorem 5*, is more precise: if t_i is negative, then at least one of its adjacent v_i must be negative. If only one t_k is negative, then both of its adjacent v_i must be negative. For further discussions on the distribution of negative coefficients, see Blanes and Casas [19].

V. THE STRUCTURE OF FORWARD INTEGRATORS

Theorems 2a and *2b* can be used to construct fourth-order forward algorithms with only positive factorization coefficients. These forward integrators are the only fourth-order factorized symplectic algorithms capable of integrating *time-irreversible* equations such as the Fokker-Planck [10,23] or the imaginary time Schrödinger equation [24–26]. Since it has been shown that [18] currently there are no practical ways of constructing sixth-order forward integrators, these fourth-order algorithms enjoy a unique status.

For $N=3$, for a fourth-order algorithm, we must require $t_2=t_3=\frac{1}{2}$. *Theorem 2a* then implies that

$$v_1 = v_3 = \tfrac{1}{6}, \quad v_2 = \tfrac{2}{3}, \quad \text{and } e_{VTV} = -\tfrac{1}{72}. \tag{5.1}$$

By moving the term $\varepsilon^3 e_{VTV}[V,[T,V]]$ back to the left-hand side of (1.2) and combining it with the central V, one recovers forward algorithm 4A [27,28]. For $N=4$ with $t_2=t_3=t_4=\frac{1}{3}$, we have

$$v_1 = v_4 = \tfrac{1}{8}, \quad v_2 = v_3 = \tfrac{3}{8}, \quad \text{and } e_{VTV} = -\tfrac{1}{192}, \tag{5.2}$$

which corresponds to forward algorithm 4D [13]. These are special cases of the general *minimal* $|e_{VTV}|$, velocity-type algorithm given by $t_1=0$, $t_i=1/(N-1)$,

$$v_1 = v_N = \frac{1}{2N}, \quad v_i = \frac{(N-1)}{N(N-2)}, \quad \text{with } e_{VTV} = -\frac{1}{24}\frac{1}{N(N-2)}. \tag{5.3}$$

This arbitrary N algorithm can serve as a useful check for any general fourth-order, velocity-type algorithm.

Alternatively, for $N=4$, we can allow t_2 to be a free parameter so that

$$t_4 = t_2, \quad t_3 = 1-2t_2. \tag{5.4}$$

Theorem 2a then fixes C_2 and e_{VTV} with

$$\phi = 6t_2(1-t_2)^2 \tag{5.5}$$

and

$$v_2 = v_3 = \frac{1}{12t_2(1-t_2)}, \quad v_1 = v_4 = \tfrac{1}{2} - v_2. \tag{5.6}$$

One recognizes that this is the one-parameter algorithm 4BDA first found in Ref. [14] using symbolic algebra. For $t_2=\frac{1}{2}$, one recovers the integrator 4A; for $t_2=\frac{1}{3}$, one gets back 4D. The advantage of using a variable t_2 is that one can use it to minimize the resulting fourth-order error (oftentime to zero) in any specific application. All the seven-stage, forward integrators in the velocity form described by Omelyan, Mryglod, and Folk (OMF) [17] correspond to different ways of choosing t_2 and distributing the commutator term in 4BDA.

For $N=5$, again using t_2 as a parameter, we have $t_1=0$, $t_5=t_2$, $t_4=t_3=\frac{1}{2}-t_2$, (4.6) with

$$\phi = \tfrac{15}{16} - 3\left(t_2 - \tfrac{1}{4}\right)^2, \tag{5.7}$$

$v_5=v_1$, $v_4=v_2$, $v_3=1-2(v_1+v_2)$, and

$$v_1 = \tfrac{1}{2} + C_2(1-t_2), \quad v_2 = -\tfrac{1}{2}C_2. \tag{5.8}$$

This is a new one-parameter family of fourth-order algorithms with nine stages or operators.

To generate position-type algorithms, one can apply *Theorem 2b*. For $N=3$, with $v_1=0$, $v_1=v_2=\frac{1}{2}$, we have

$$t_1 = t_3 = \frac{1}{2}\left(1 - \frac{1}{\sqrt{3}}\right), \quad t_2 = \frac{1}{\sqrt{3}}, \quad \text{and } e_{VTV} = -\tfrac{1}{12}\left(1 - \tfrac{1}{2}\sqrt{3}\right). \tag{5.9}$$

This produces forward algorithm 4B [27,28] corresponding to $t_2=(1-1/\sqrt{3})/2$ in 4BDA. Again, this is a special case of the general fourth-order, minimal $|e_{VTV}|$ algorithm with $v_1=0$, $v_i=1/(N-1)$,

$$t_1 = t_N = \frac{1}{2}\left(1 - \sqrt{\frac{N-2}{N}}\right), \quad t_i = \frac{1}{\sqrt{N(N-2)}}, \tag{5.10}$$

and

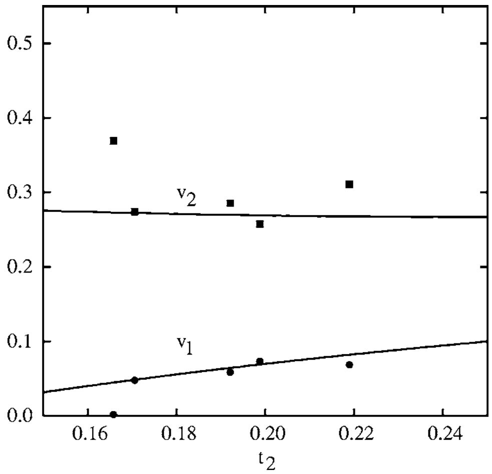

FIG. 1. Comparing the coefficients of five, 9-stage, *velocity*-type, fourth-order forward integrators of Omelyan, Mryglod, and Folk [17] (filled circles and squares), with the analytical prediction of extended-linear symplectic integrators (solid lines).

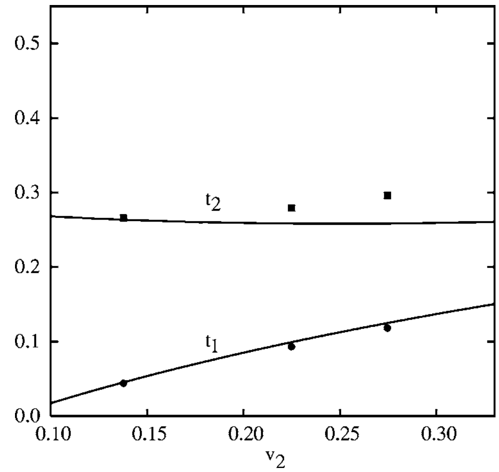

FIG. 2. Comparing the coefficients of three, 9-stage, *position*-type, fourth-order forward integrators of Omelyan, Mryglod, and Folk [17] (filled circles and squares), with the analytical prediction of extended-linear symplectic integrators (solid lines).

$$e_{VTV} = -\frac{1}{12}\left(1 - \frac{\sqrt{N(N-2)}}{(N-1)}\right). \tag{5.11}$$

For $N=4$, $v_1=0$ and v_2 as the free parameter, invoking *Theorem 2b* gives

$$v_4 = v_2, \quad v_3 = 1 - 2v_2, \tag{5.12}$$

$$t_2 = t_3 = \frac{1}{2\sqrt{6v_2}}, \quad t_1 = t_4 = \tfrac{1}{2} - t_2 \tag{5.13}$$

and

$$e_{VTV} = -\tfrac{1}{12}[1 - (1 - v_2)\sqrt{6v_2}]. \tag{5.14}$$

For $v_2=\frac{1}{6}$ and $v_2=\frac{3}{8}$, this reproduces algorithms 4A and 4C [28], respectively. One again recognizes that the above is the one-parameter algorithm 4ACB first derived in Ref. [14], but now with a much simpler parametrization. Algorithm 4ACB covers all of the seven-stage, forward fourth-order position-type integrators described by OMF [17].

For $N=5$, with v_2 as a free parameter, we have $v_1=0$, $v_5=v_2$, $v_4=v_3=\frac{1}{2}-v_2$, and *Theorem 2b* produces another 9-stage fourth-order algorithm with

$$\phi' = \sqrt{\tfrac{15}{16} - 3\left(v_2 - \tfrac{1}{4}\right)^2}. \tag{5.15}$$

$t_5=t_1$, $t_4=t_2$, $t_3=1-2(t_1+t_2)$, and

$$t_1 = \tfrac{1}{2} + C_2(1 - v_2), \quad t_2 = -\tfrac{1}{2}C_2. \tag{5.16}$$

For $N<5$, we have shown above that all fourth-order algorithms are necessarily extended-linear. For $N \geqslant 5$, this is not necessarily the case. Nevertheless we find that, remarkably, most known $N=5$ (9 stages) forward algorithms are very close to being extended-linear. For velocity-type, $N=5$ extended-linear algorithms, v_1 and v_2 are functions of t_2 fixed by (5.8). In Fig. 1, we compare this predicted relationship with the actual values of v_1, v_2, and t_2 of five forward, velocity-type, fourth-order algorithms found by OMF [17]. These are their Eqs. (52)–(56), and with their θ, ϑ, and λ correspond to t_2, v_1, and v_2, respectively. Four of their five algorithms, with v_1 in particular, are well described by (5.8).

In Fig. 2, we compare the coefficients of all three of OMF's forward, position-type algorithms, Eqs. (59)–(61), with (5.16) which fixes t_1, t_2 as a function of v_2. Here, their parameters λ, ρ, θ correspond to v_2, t_1, t_2, respectively. Again, t_1 is particularly well predicted by (5.16).

For 11-stage algorithms with $N=6$, we have two free parameters t_2, t_3 for velocity-type algorithms with

$$\phi = 1 - 2t_2^3 - 2t_3^3 - (1 - 2t_2 - 2t_3)^3 \tag{5.17}$$

and two free parameters v_2, v_3 for position-type algorithms with

$$\phi' = \sqrt{1 - 2v_2^3 - 2v_3^3 - (1 - 2v_2 - 2v_3)^3}. \tag{5.18}$$

Once ϕ and ϕ' are known, we can determine v_1 and v_2 in the case of velocity-type algorithms and t_1 and t_2 in the case of position-type algorithms. There is one 11-stage velocity algorithm with positive coefficients found by OMF; their Eq. (68) with $\rho(=t_2)=0.2029$, $\theta(=t_3)=0.1926$,

$$\vartheta(=v_1) = 0.0667, \quad \text{and } \lambda(=v_2) = 0.2620. \tag{5.19}$$

The last two values are to be compared with the values given by *Theorem 2a* below at the same values of t_2 and t_3,

$$v_1 = 0.0848, \quad \text{and } v_2 = 0.2060. \tag{5.20}$$

For OMF's 11-stage, position-type algorithm Eq. (78), with $\vartheta(=v_2)=0.1518$, $\lambda(=v_3)=0.2158$,

$$\rho(=t_1) = 0.0642, \quad \text{and } \theta(=t_2) = 0.1920. \tag{5.21}$$

For the same values of v_2 and v_3, *Theorem 2b* gives

$$t_1 = 0.0659, \quad \text{and } t_2 = 0.1881. \tag{5.22}$$

It is remarkable that these 11-stage, fourth-order algorithms derived by complex symbolic algebra, remained very close to the values predicted by our extended-linear algorithms.

VI. THE STRUCTURE OF NONFORWARD INTEGRATORS

Theorem 3 can be used to derive two distinct families of nonforward, fourth-order algorithms. Consider first the case of $N=4$. For $t_1=0$ with symmetric coefficients $t_4=t_2$, the constraints

$$2t_2 + t_3 = 1, \tag{6.1}$$

$$2t_2^3 + t_3^3 = 0, \tag{6.2}$$

have unique solutions

$$t_2 = \frac{1}{2-2^{1/3}} \text{ and } t_3 = -\frac{2^{1/3}}{2-2^{1/3}}. \tag{6.3}$$

Equation (4.16) then yields

$$v_1 = v_4 = \frac{1}{2}\frac{1}{2-2^{1/3}}, \quad v_2 = v_3 = -\frac{1}{2}\frac{(2^{1/3}-1)}{2-2^{1/3}}. \tag{6.4}$$

One recognizes that we have just derived the well-known fourth-order Forest-Ruth integrator [29]. Note that there is complete symmetry between $\{t_i\}$ and $\{v_i\}$. For position-type algorithms, we simply interchange the values of t_i and v_i.

There are no symmetric solutions for $N=5$, for the same reason that there are also no solutions for $N=3$. For $N=2k$, we have the general condition

$$2\sum_{i=2}^{k} t_i + t_{k+1} = 1,$$

$$2\sum_{i=2}^{k} t_i^3 + t_{k+1}^3 = 0, \tag{6.5}$$

which can be solved by introducing real parameters α_i for $i=2$ to k with $\alpha_2=1$,

$$t_i = \alpha_i t_2, \tag{6.6}$$

so that

$$t_{k+1} = -2^{1/3}\left(\sum_{i=2}^{k}\alpha_i^3\right)^{1/3} t_2, \tag{6.7}$$

$$t_2 = \frac{1}{2\left(\sum_{i=2}^{k}\alpha_i\right) - 2^{1/3}\left(\sum_{i=2}^{k}\alpha_i^3\right)^{1/3}}. \tag{6.8}$$

These solutions generalize the fourth-order Forest-Ruth integrator to arbitrary N.

The general fourth-order condition (6.5) has been derived previously by McLachlan [30] using the generalized triplet construction published by Creutz and Gocksch [31]. However, the invocation of *Theorem 3* is more general and much simpler. McLachlan suggests that one should just set all $\alpha_i=1$.

For $N=2k+1$, $k>2$, again introducing (6.6) for $i=2$ to k with $\alpha_2=1$, we have

$$t_{k+1} = -\left(\sum_{i=2}^{k}\alpha_i^3\right)^{1/3} t_2, \tag{6.9}$$

$$t_2 = \frac{1}{2\left(\sum_{i=2}^{k}\alpha_i\right) - 2\left(\sum_{i=2}^{k}\alpha_i^3\right)^{1/3}}. \tag{6.10}$$

This is a new class of a fourth-order algorithm possible only for N odd greater than 5 and is not derivable from the triplet construction.

VII. CONCLUSIONS

Most of the machinery for tracking coefficients was developed in Ref. [18] for the purpose of providing a constructive proof of the Sheng-Suzuki theorem. The advantage of this constructive approach is that we can obtain explicit lower bounds on the second-order error coefficients. Here, by imposing the extended-linear relationship between $\{t_i\}$ and $\{v_i\}$, these bounds become the actual error coefficients and provide a complete characteriation for all fourth-order symplectic integrators for an arbitrary number of operators. The most satisfying aspect of this work is that most fourth-order integrators can now be derived analytically without recourse to symbolic algebra or numerical root finding. We have also provided explicit construction of many new classes of fourth-order algorithms.

For $N=5,6$, corresponding to 9 and 11 operators, we have shown that many fourth-order algorithms found by Omelyan, Mryglod, and Folk [17] are surprisingly close to the predicted coefficients of our theory, suggesting that the extended-linear relation between coefficients may be the dominate solution of the order condition.

The expansion (3.2) may hold similar promise for characterizing sixth-order algorithms by introducing extended-quadratic or higher order relationships between the two sets of coefficients.

ACKNOWLEDGMENTS

I would like to thank Dr. Blanes and Dr. Casas for an interesting comment which led to this work. This work was supported, in part, by a National Science Foundation Grant No. DMS-0310580.

[1] H. Yoshida, Celest. Mech. **56**, 27 (1993).
[2] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration* (Springer-Verlag, Berlin, 2002).
[3] R. I. McLachlan and G. R. W. Quispel, Acta Numerica **11**, 241 (2002).
[4] S. A. Chin and C. R. Chen, Celest. Mech. Dyn. Astron. **91**, 301 (2005).
[5] S. R. Scuro and S. A. Chin, Phys. Rev. E **71**, 056703 (2005).
[6] T. Hirono, W. Lui, S. Seki, and Y. Yoshikuni, IEEE Trans. Microwave Theory Tech. **49**, 1640 (2001).
[7] J. Lee and B. Fornberg, J. Comput. Appl. Math. **158**, 485 (2003).
[8] M. Takahashi and M. Imada, J. Phys. Soc. Jpn. **53**, 3765 (1984).
[9] S. Jang, S. Jang, and G. A. Voth, J. Chem. Phys. **115**, 7832 (2001).
[10] H. A. Forbert and S. A. Chin, Phys. Rev. E **63**, 016703 (2001).
[11] S. A. Chin, Phys. Rev. E **69**, 046118 (2004).
[12] D. Feit, J. A. Fleck, Jr., and A. Steiger, J. Comput. Phys. **47**, 412 (1982).
[13] S. A. Chin and C. R. Chen, J. Chem. Phys. **114**, 7338 (2001).
[14] S. A. Chin and C. R. Chin, J. Chem. Phys. **117**, 1409 (2002).
[15] P. V. Koseleff, *Integration Algorithms and Classical Mechanics*, Fields Institute Communications, Vol. 10 (American Mathematical Society, Providence, RI, 1996), p. 103.
[16] I. P. Omelyan, I. M. Mryglod, and R. Folk, Phys. Rev. E **66**, 026701 (2002).
[17] I. P. Omelyan, I. M. Mryglod, and R. Folk, Comput. Phys. Commun. **151**, 272 (2003).
[18] S. A. Chin, Phys. Rev. E **71**, 016703 (2005).
[19] S. Blanes and F. Casas, Appl. Numer. Math. **54**, 23 (2005).
[20] Q. Sheng, IMA J. Numer. Anal. **9**, 199 (1989).
[21] M. Suzuki, J. Math. Phys. **32**, 400 (1991).
[22] D. Goldman and T. J. Kaper, IMA J. Numer. Anal. **33**, 349 (1996).
[23] H. A. Forbert and S. A. Chin, Phys. Rev. B **63**, 144518 (2001).
[24] J. Auer, E. Krotscheck, and S. A. Chin, J. Chem. Phys. **115**, 6841 (2001).
[25] O. Ciftja and S. A. Chin, Phys. Rev. B **68**, 134510 (2003).
[26] S. A. Chin and E. Krotscheck, Phys. Rev. E **72**, 036705 (2005).
[27] M. Suzuki, in *Computer Simulation Studies in Condensed Matter Physics VIII*, edited by D. Landau, K. Mon, and H. Shuttler (Springler, Berlin, 1996).
[28] S. A. Chin, Phys. Lett. A **226**, 344 (1997).
[29] E. Forest and R. D. Ruth, Physica D **43**, 105 (1990).
[30] R. I. McLachlan, Numer. Algorithms **31**, 233 (2002).
[31] M. Creutz and A. Gocksch, Phys. Rev. Lett. **63**, 9 (1989).