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 simplifies 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

\[ w(t + \varepsilon) = e^{(T+V)t}w(t), \]

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 \( e^{(T+V)t} \) to the \((n+1)\)th order in the product form

\[ e^{\varepsilon(T+V)} = \prod_{i=1}^{N} e^{\varepsilon T} e^{\varepsilon V} + O(\varepsilon^{n+1}) \] (1.2)

via a well chosen set of factorization (or split) coefficients \( \{t_i\} \) and \( \{v_i\} \). The resulting algorithm is then \( n \)th 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 \( n \)th 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

\[ \prod_{i=1}^{N} e^{\varepsilon T} e^{\varepsilon V} = \exp\{\varepsilon e_T T + \varepsilon e_V V + \varepsilon^2 e_{TTV} [T,V] + \varepsilon^3 e_{TTTV} [T,[T,V]] + \cdots \}. \] (2.1)

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

\[ e_T = \sum_{i=1}^{N} t_i, \quad e_V = \sum_{i=1}^{N} v_i, \] (2.2)

\[ \frac{1}{2} + e_{TV} = \sum_{i=1}^{N} \nabla s_i t_i, \] (2.3)

\[ \frac{1}{3!} + \frac{1}{2} e_{TV} + e_{TTV} = \frac{1}{2} \sum_{i=1}^{N} \nabla s_i v_i, \] (2.4)

\[ \frac{1}{3!} + \frac{1}{2} e_{TTV} - e_{TTTV} = \frac{1}{2} \sum_{i=1}^{N} \nabla s_i t_i^2, \] (2.5)

where we have defined useful variables

\[ s_i = \sum_{j=1}^{i} t_j, \quad u_i = \sum_{j=1}^{N} v_j, \] (2.6)

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

\[ \nabla s_i^\alpha = s_i^\alpha - s_{i-1}^\alpha, \] (2.7)

with property

\[ \sum_{i=1}^{N} \nabla s_i^\alpha = s_N^\alpha (= e_T^\alpha = 1). \] (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 \( e_{TTTV}=0 \). 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 \textit{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 \( e_{TV} \) and \( e_{VT} \) 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

\[
\sum_{i=1}^{N} \nabla s \mu_i = \frac{1}{2}.
\]  

(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

\[
u_i = \sum_{n=1}^{M} C_n \nabla s_i^{n}, \quad \text{with} \quad \sum_{n=1}^{M} C_n = \frac{1}{2}.
\]  

(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.,

\[
\nu_i = C_1 + C_2 \frac{\nabla s_i^2}{\nabla s_i} = C_1 + C_2 (s_i + s_{i-1}).
\]  

(3.3)

For \( i=1 \), since we must satisfy the primary constraint \( e_V=1 \), we must have

\[
\nu_1 = 1.
\]  

(3.4)

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

\[
\sum_{i=1}^{N} \nabla s_i \mu_i = t_1 + C_1 (1 - t_1) + C_2 (1 - t_1^2) = \frac{1}{2}.
\]  

(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

\[
\nu_1 = 0,
\]  

(3.6)

so that (3.5) remains

\[
C_1 + C_2 = \frac{1}{2}.
\]  

(3.7)

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

\[
\nu_i = -C_2 (t_i + t_{i+1}).
\]  

(3.8)

Since \( v_1 = u_1 - u_2 = 1 - C_1 - C_2 t_2 \), by virtue of (3.7),

\[
\nu_1 = \frac{1}{2} + C_2 (1 - t_2).
\]  

(3.9)

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

\[
v_N = \frac{1}{2} + C_2 (1 - t_N).
\]  

(3.10)

Given \( \{t_i\} \) such that \( t_1=0 \), the set of \( \{v_i\} \) defined by (3.8)–(3.10) automatically satisfies \( e_T=1 \) and \( e_V=0 \). 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 \( \{t_i\} \), \textit{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

\[
\sum_{i=1}^{N} \nabla s_i^2 \nabla s_i = 1 + g,
\]  

(3.11)

\[
\sum_{i=1}^{N} \nabla s_i^2 \nabla s_i = 1 + g,
\]  

(3.12)

Again the complication introduced by \( u_1=1 \) is avoided by decreeing \( t_1=0 \). The quantity \( g \) is a frequently occurring sum defined via

\[
\sum_{i=1}^{N} \nabla s_i^2 \nabla s_i = 1 + g,
\]  

(3.13)

with explicit form

\[
\nu = \sum_{i=1}^{N} s_i s_{i-1} (s_i - s_{i-1}) = \frac{1}{3} (1 - \delta g),
\]  

(3.14)

where

\[
\delta g = \sum_{i=1}^{N} r_i^2.
\]  

(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

\[
e_{TV} = \frac{1}{12} - \frac{1}{2} g C_2,
\]  

(3.16)

\[
e_{VT} = \frac{1}{24} - \frac{1}{2} g C_2
\]  

(3.17)

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

### IV. FUNDAMENTAL THEOREMS

We can now prove a number of important results:

\textit{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 \( e_{TV} \neq e_{VT} \).

\textit{Proof.} Setting \( e_{TV}=e_{VT} \) produces a quadratic equation for \( C_2 \),

\[
C_2 + C_2 + \frac{1}{12} g = 0
\]  

(4.1)

whose discriminant
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 factorization. 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 \( \{t_i>1\} \), \( e_{TTV} \) 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

\[
C_2 = -\frac{1}{2(1-\delta g)}, \quad e_{VTV} = -\frac{1}{24} \frac{\delta g}{1-\delta g}.
\]

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

\[
C_2 = -\frac{1}{2\sqrt{1-\delta g}}, \quad e_{TTV} = \frac{1}{12} \left(1 - \sqrt{1-\delta g}\right).
\]

Satisfying either condition forces \( C_2 \) to be a function of \( t_i \) through \( \delta g \). From Ref. [18], we have learned that the value given by (4.3) is actually an upper bound for \( e_{VTV} \) if \( \{t_i>1\} \) and \( e_{TTV}=0 \). Similarly, in general, the value given by (4.4) is a lower bound for \( e_{TTV} \) if \( \{t_i>1\} \) 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 \( \{t_{i>1}\} \), it is useful to state (4.3) as a theorem:

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

\[
v_1 = \frac{1}{2} + C_2(1 - t_2), \quad v_N = \frac{1}{2} + C_2(1 - t_N),
\]

\[
v_i = - C_2(t_i + t_{i+1}),
\]

with \( t_{i=0} \), \( e_T=1 \), and \( C_2, e_{VTV} \) given by

\[
C_2 = -\frac{1}{2\phi'}, \quad e_{VTV} = -\frac{1}{24} \left(1 - \frac{1}{\phi} - 1\right),
\]

where

\[
\phi = 1 - \delta g \quad \text{and} \quad \delta g = \sum_{i=1}^{N} t_i^3,
\]

one has

\[
\prod_{i=1}^{N} e^{e_{TTV} e^{i\epsilon} e^{T}} = \exp(e(T + V) + e^{3} e_{VTV}[V,[T,V]] + \cdots).
\]

For \( t_{i=0} \), the first operator \( e^{t_0 e^{T}} \) classically updates the velocity (momentum) variable. **Theorem 2b** completely describes the structure of these velocity-type algorithms.

If one now interchanges \( T \mapsto V \) and \( \{t_i\} \mapsto \{v_i\} \), 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

\[
t_1 = \frac{1}{2} + C_2(1 - v_2), \quad t_N = \frac{1}{2} + C_2(1 - v_N),
\]

\[
t_i = - C_2(v_i + v_{i+1}),
\]

with \( v_{i=0} \), \( e_V=1 \), and \( C_2, e_{VTV} \) given by

\[
C_2 = -\frac{1}{2\phi'}, \quad e_{VTV} = -\frac{1}{12} (1 - \phi'),
\]

where

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

one has

\[
\prod_{i=1}^{N} e^{e_{VTV} e^{i\epsilon} e^{T}} = \exp(e(T + V) + e^{3} e_{VTV}[V,[T,V]] + \cdots).
\]

For \( v_{i=0} \), the first operator \( e^{t_0 e^{T}} \) classically updates the position variable. **Theorem 2b** completely describes 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

\[
C_2 = -\frac{1}{2}
\]

and

\[
v_i = \frac{1}{2}(t_i + t_{i+1}).
\]

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

\[
e_{TTV} = 2e_{VTV} = \frac{1}{12} - \frac{g}{4} = \frac{1}{12} \delta g.
\]

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
were to vanish, then there must be at least one side of $x$ or $tk$ that must contain at least one pair of negative coefficients than in deriving fourth-order algorithms. For example, an immediate corollary of Theorem 3 is that if $\delta g$ 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

$$x^3 + y^3 < 0 \Rightarrow x + y < 0.$$  

We therefore must have $t_k^3 + t_{k+1}^3 < 0$ or $t_k^3 + t_{k-1}^3 < 0$. 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 $t$ and $v_i$. 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

$$v_1 = v_4 = \frac{1}{8}, \quad v_2 = v_3 = \frac{3}{8}, \quad \text{and} \quad e_{VTV} = -\frac{1}{12},$$  

(5.1)

By moving the term $\epsilon^3 e_{VTV}[V, [T, V]]$ 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

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

(5.2)

which corresponds to forward algorithm 4D [13]. These are special cases of the general minimal $|e_{VTV}|$, velocity-type algorithm given by $t_1 = 0$, $t_2 = 1/(N-1)$,

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

(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

$$t_4 = t_2, \quad t_5 = 1 - 2t_2.$$  

(5.4)

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

$$\phi = 6t_2(1-t_2)^2$$  

(5.5)

and

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

(5.6)

One recognizes that this is the one-parameter algorithm 4BDA first found in Ref. [14] using symbolic algebra. For $t_2 = \frac{1}{3}$, one recovers the integrator 4A; for $t_2 = \frac{1}{2}$, 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 (oftentimes to zero) in any specific application. All the seven-stage, forward integrators in the velocity form described by Omelyan, Myrglod, and Folk (OMF) [17] correspond to different ways of choosing $t_2$ and separating the commutator term in 4BDA.

For $N=5$, again using $t_2$ as a parameter, we have $t_1 = 0$, $t_5 = t_2$, $t_4 = t_3 = \frac{2}{3} - t_2$, (4.6) with

$$\phi = \frac{15}{16} - 3\left(t_2 - \frac{1}{2}\right)^2,$$  

(5.7)

$$v_5 = v_1, \quad v_4 = v_2, \quad v_3 = 1 - 2(v_1 + v_2), \quad \text{and} \quad v_1 = \frac{1}{3} + C_2(1 - t_2), \quad v_2 = -\frac{1}{3}C_2.$$  

(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 $v_1 = 0$, $v_i = v_{i-1} + \frac{1}{8}$, we have

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

(5.9)

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

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

(5.10)

and
For $N=5$, with $v_2$ as a free parameter, we have $v_1=0$, $v_3=v_2$, $4=v_3-\frac{1}{2}$ and $Theorem 2b$ produces another 9-stage fourth-order algorithm with

$$
\phi' = \sqrt{\frac{15}{16} - 3(v_2 - \frac{1}{4})^2.}
$$

(5.15)

$$
t_5 = t_1, \quad t_4 = t_2, \quad t_3 = 1 - 2(t_1 + t_2), \quad \text{and}
$$

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

For $N<5$, we have shown above that all fourth-order algorithms are necessarily extended-linear. For $N\geq 5$, 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 $\theta$, $\theta$, and $\lambda$ 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 $\lambda$, $\rho$, $\theta$ 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

$$
\phi = 1 - 2t_2^3 - 2t_3^3 - (1 - 2t_2 - 2t_3)^3
$$

(5.17)

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

$$
\phi' = (1 - 2v_2^3 - 2v_3^3 - (1 - 2v_2 - 2v_3)^3).
$$

(5.18)

Once $\phi$ and $\phi'$ 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 $\rho(=t_2)=0.2029, \theta(=t_3)=0.1926$,

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

The last two values are to be compared with the values given by Theorem 2a below at the same values of $t_2$ and $t_3$, $v_1=0.0848, \quad v_2=0.2060. \quad \text{(5.20)}$

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

$$
\rho(=t_1)=0.0642, \quad \theta(=t_3)=0.1920. \quad \text{(5.21)}$

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

$$
t_1=0.0659, \quad t_2=0.1881. \quad \text{(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 \( t_2=t_3 \), the constraints

\[
2t_2 + t_3 = 1, \quad (6.1)
\]

\[
2t_2^3 + t_3^3 = 0, \quad (6.2)
\]

have unique solutions

\[
t_2 = \frac{1}{2 - 2^{1/3}} \quad \text{and} \quad t_3 = -\frac{2^{1/3}}{2 - 2^{1/3}}. \quad (6.3)
\]

Equation (4.16) then yields

\[
v_1 = v_4 = \frac{1}{2} \left( \frac{1}{2 - 2^{1/3}} \right), \quad v_2 = v_3 = -\frac{1}{2} \left( \frac{2^{1/3} - 1}{2 - 2^{1/3}} \right). \quad (6.4)
\]

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

\[
2 \sum_{i=2}^{k} t_i + t_{k+1} = 1, \quad (6.5)
\]

\[
2 \sum_{i=2}^{k} t_i^3 + t_{k+1}^3 = 0,
\]

which can be solved by introducing real parameters \( \alpha_i \) for \( i=2 \) to \( k \) with \( \alpha_2=1 \),

\[
t_i = \alpha_i t_2, \quad (6.6)
\]

so that

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

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

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 \( \alpha_i \) equal to 1.

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

\[
t_k+1 = -\left( \sum_{i=2}^{k} \alpha_i^3 \right)^{1/3} t_2, \quad (6.9)
\]

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

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 characterization 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.