

Higher-order splitting algorithms for solving the nonlinear Schrödinger equation and their instabilities

Siu A. Chin

Department of Physics, Texas A&M University, College Station, Texas 77843, USA

(Received 9 July 2007; revised manuscript received 4 September 2007; published 29 November 2007)

Since the kinetic and potential energy terms of the real-time nonlinear Schrödinger equation can each be solved exactly, the entire equation can be solved to any order via splitting algorithms. We verified the fourth-order convergence of some well-known algorithms by solving the Gross-Pitaevskii equation numerically. All such splitting algorithms suffer from a latent numerical instability even when the total energy is very well conserved. A detail error analysis reveals that the noise, or elementary excitations of the nonlinear Schrödinger equation, obeys the Bogoliubov spectrum and the instability is due to the exponential growth of high-wave-number noises caused by the splitting process. For a continuum wave function, this instability is unavoidable no matter how small the time step. For a discrete wave function, the instability can be avoided only for $\Delta t k_{max}^2 \lesssim 2\pi$, where $k_{max} = \pi/\Delta x$.

DOI: 10.1103/PhysRevE.76.056708

PACS number(s): 02.70.Hm, 03.75.Kk, 02.70.-c

I. INTRODUCTION

Taha and Ablowitz [1] showed some time ago that the first-order pseudospectral, split-operator method is a very fast way of solving the nonlinear Schrödinger equation. Bandrauk and Shen [2] later applied higher-order splitting algorithms with negative coefficients to solve the same equation. They regarded the nonlinear potential as time dependent. Since they can only estimate the intermediate-time nonlinear potential to second order, it is not proven that their higher-order algorithms actually converge at fourth or sixth order. Recently Javanainen and Ruostekoski [3] have shown by symbolic calculations that fourth-order algorithms are possible by use of the “latest” intermediate wave function in evaluating the nonlinear potential. Strauch [4], by constructing a special operator that correctly propagates the nonlinear potential term, proved that this use of the “latest” intermediate wave function is valid.

This work shows that (i) Javanainen and Ruostekoski’s finding is a direct consequence of Taha and Ablowitz’ original work and a much simpler proof than that of Strauch is possible. (ii) The time-dependent potential method of Bandrauk and Shen and the time-independent approach suggested by Javanainen and Ruostekoski both yielded identical second-order algorithms but different higher-order algorithms. (iii) It is verified numerically that algorithms derived by the time-independent method do converge to fourth order when solving the Gross-Pitaevskii equation. (iv) All such splitting algorithms possess a latent numerical instability which causes the energy of the wave function to eventually blow up despite excellent total energy conservation for a long time. (v) The instability is shown to be due to the exponential growth of high-wave-number noises intrinsic to the splitting process. For a continuum wave function, this instability is unavoidable no matter how small the time step is. For a discrete wave function, this can only be avoided if $\Delta t \lesssim 2\pi/k_{max}^2$, which forces Δt to be very small if the discretization is very fine with a large $k_{mas} = \pi/\Delta x$. The next three sections summarize how higher-order algorithms can be systematically derived, and Sec. V discusses the instability in detail.

II. SOLVING THE NONLINEAR SCHRÖDINGER EQUATION

Consider the nonlinear Schrödinger equation defined by

$$i\frac{\partial\psi}{\partial t} = \left(-\frac{1}{2}\nabla^2 + g|\psi|^2\right)\psi. \quad (1)$$

The free particle propagation can be solved exactly in operator form

$$\psi(\Delta t) = e^{-i\Delta t\hat{T}}\psi(0), \quad (2)$$

where the operator $\hat{T} = -\frac{1}{2}\nabla^2$. Since \hat{T} is diagonal in k -space, Eq. (2) is usually solved by fast Fourier transforms (FFTs). Surprisingly, as shown by Taha and Ablowitz, the potential part of the equation

$$i\frac{\partial\psi}{\partial t} = g|\psi|^2\psi \quad (3)$$

can also be solved exactly:

$$\psi(\Delta t) = e^{-i\Delta t g|\psi(0)|^2}\psi(0). \quad (4)$$

This is because Eq. (3) exactly conserves $|\psi|^2$ [multiply Eq. (3) by ψ^* , the complex-conjugated equation by ψ , and subtract] and the nonlinear potential is just a constant in Eq. (3). This is also clear from Eq. (4),

$$|\psi(\Delta t)|^2 = |\psi(0)|^2, \quad (5)$$

since $\psi(0)$ is only multiplied by a phase. Equations (2) and (4) are the basic building blocks for constructing splitting algorithms for solving the nonlinear Schrödinger equation. Equation (4) is the fundamental justification for using the “latest” wave function in computing the nonlinear potential [3]. (See also below.) Define a *time-independent* operator \hat{V} such that

$$\hat{V}|\psi(t)\rangle = g|\psi(t)|^2|\psi(t)\rangle. \quad (6)$$

That is, we define a time-independent operator \hat{V} , whose eigenvalue is the nonlinear potential $g|\psi(t)|^2$. Note that \hat{V} only

acts on $|\psi(t)\rangle$ and does not act on its own eigenvalue $g|\psi(t)\rangle^2$. It follows that

$$e^{-i\Delta t\hat{V}}|\psi(t)\rangle = e^{-i\Delta t g|\psi(t)\rangle^2}|\psi(t)\rangle. \quad (7)$$

The crucial point here is that \hat{V} has no time dependence; when it acts on any $|\psi(t)\rangle$, it produces the eigenvalue $g|\psi(t)\rangle^2$. The resulting time dependence of the nonlinear potential is due entirely to the state vector $|\psi(t)\rangle$ and not to the operator \hat{V} . The exact solution can then be written in operator form as

$$|\psi(t)\rangle = e^{-it(\hat{T}+\hat{V})}|\psi(0)\rangle. \quad (8)$$

For our purposes here, we only need to know Eq. (7) and not the explicit form of \hat{V} . For an elegant, but rather abstract construction of \hat{V} , see Strauch's [4] recent work.

III. DERIVING SPLITTING ALGORITHMS

To solve Eq. (8) by splitting algorithms, one factorizes the evolution operator to any order with a suitable set of coefficients $\{t_i, v_i\}$ via

$$e^{\varepsilon(\hat{T}+\hat{V})} = \prod_i e^{\varepsilon t_i \hat{T}} e^{\varepsilon v_i \hat{V}}, \quad (9)$$

where we have denoted $\varepsilon = -i\Delta t$. For example, we can have the second-order algorithm 2A as

$$\psi(\Delta t) = e^{\varepsilon \hat{V}/2} e^{\varepsilon \hat{T}} e^{\varepsilon \hat{V}/2} \psi(0) = e^{\varepsilon g|\phi|^2/2} e^{\varepsilon \hat{T}} e^{\varepsilon g|\psi(0)\rangle^2/2} \psi(0), \quad (10)$$

where according to Eq. (4) or (7), we must take

$$\phi = e^{\varepsilon \hat{T}} e^{\varepsilon g|\psi(0)\rangle^2/2} \psi(0). \quad (11)$$

Algorithm 2A only requires one pair of FFTs (forward and backward) to achieve second-order accuracy, which is the same number of FFTs needed for a first-order algorithm. If the nonlinear potential is treated as a time-dependent potential, as done by Bandrauk and Shen [2], then we would have the algorithm [5,6]

$$\psi(\Delta t) = e^{-i\Delta t V(\Delta t)/2} e^{-i\Delta t \hat{T}} e^{-i\Delta t V(0)/2} \psi(0). \quad (12)$$

In this case, since the last factor is only a phase,

$$V(\Delta t) = g|\psi(\Delta t)\rangle^2 = g|\phi|^2, \quad (13)$$

the result is the same as Eq. (10). If one ignores the time dependence [7] and uses $V(\Delta t) = V(0) = g|\psi(0)\rangle^2$, then algorithm (12) is degraded to first order.

Similarly one has the second-order algorithm 2B,

$$\psi(\Delta t) = e^{\varepsilon \hat{T}/2} e^{\varepsilon \hat{V}} e^{\varepsilon \hat{T}/2} \psi(0) = e^{\varepsilon \hat{T}/2} e^{\varepsilon g|\phi|^2} e^{\varepsilon \hat{T}/2} \psi(0), \quad (14)$$

where here

$$\phi = e^{\varepsilon \hat{T}/2} \psi(0). \quad (15)$$

In the time-dependent potential approach, one would have, instead,

$$\psi(\Delta t) = e^{-i\Delta t \hat{T}/2} e^{-i\Delta t V(\Delta t/2)} e^{-i\Delta t \hat{T}/2} \psi(0). \quad (16)$$

One must now evaluate $V(\Delta t/2) = g|\psi(\Delta t/2)\rangle^2$. Since the algorithm is only second order, one can simply approximate the midpoint wave function to first order,

$$\psi(\Delta t/2) = e^{-i\Delta t V(\Delta t/2)/2} e^{-i\Delta t \hat{T}/2} \psi(0), \quad (17)$$

and therefore

$$|\psi(\Delta t/2)\rangle^2 = |e^{-i\Delta t \hat{T}/2} \psi(0)\rangle^2. \quad (18)$$

Again, the result is the same as Eq. (14)

For fourth- and higher-order algorithms, the time-dependent potential approach cannot be easily implemented. It is much more efficient to use the "latest" intermediate wave function than to estimate the intermediate-time wave function to third or higher order. Thus higher-order algorithms are currently possible only with the use of the time-independent formalism based on the original finding of Taha and Ablowitz.

The fourth-order Forest-Ruth (FR) [9] algorithm, which is the triplet concatenation [8,10,11] of algorithm 2A,

$$\mathcal{T}_{FR}(\varepsilon) = \mathcal{T}_{2A}(c_1\varepsilon)\mathcal{T}_{2A}(c_0\varepsilon)\mathcal{T}_{2A}(c_1\varepsilon), \quad (19)$$

with $c_1 = 1/(2-2^{1/3})$ and $c_0 = -2^{1/3}/(2-2^{1/3})$, has been verified by Javanainen and Ruostekoski as obeying the "latest" intermediate wave function rule. However, since this triplet concatenation will convert any reversible, second-order exponential-splitting algorithm to fourth order, verifying this algorithm alone does not constitute an independent check on more general fourth-order algorithms. (Recall that algorithm 2A can also be derived from the time-dependent approach without explicitly invoking the "latest" wave function rule.) (Javanainen and Ruostekoski have also verified the "latest" wave function rule on a class of third-order algorithms independent of 2A.) To seal this loophole in our verification process, we also consider more general fourth-order algorithms previously studied by McLachlan [12] with nine operators,

$$\mathcal{T}_M = \cdots \exp(\varepsilon t_0 \hat{V}) \exp(\varepsilon v_1 \hat{T}) \exp(\varepsilon t_1 \hat{V}) \exp(\varepsilon v_2 \hat{T}) \exp(\varepsilon t_2 \hat{V}). \quad (20)$$

The factorization is left-right symmetric, and only operators from the center to the right are indicated. The fourth-order order condition requires [13] that

$$v_1 = \frac{1}{2} - v_2, \quad t_2 = \frac{1}{6} - 4t_1 v_1^2, \quad t_0 = 1 - 2(t_1 + t_2), \quad (21)$$

$$w = \sqrt{3 - 12t_1 + 9t_1^2}, \quad v_2 = \frac{1}{4} \left(1 \mp \sqrt{\frac{9t_1 - 4 \pm 2w}{3t_1}} \right), \quad (22)$$

and that the free parameter $t_1 < 0$. This algorithm requires four pairs of FFTs but has a much smaller energy error and greater stability than that of the FR algorithm. (The coefficient designation does not match the the operators because the algorithm has been adapted from its classical version by

interchanging $\hat{T} \leftrightarrow \hat{V}$.) There are four solution branches for v_2 . The choice of

$$t_1 = \frac{121}{3924}(12 - \sqrt{471}) \approx -0.299,$$

with

$$v_2 = \frac{1}{4} \left(1 + \sqrt{\frac{9t_1 - 4 + 2w}{3t_1}} \right), \quad (23)$$

reproduces McLachlan's [12] recommended algorithm. By varying t_1 and using different branches of v_2 , it is possible to optimize the algorithm for specific applications. For application in the next section, the results are not very sensitive to the branch of v_2 or the choice of t_1 , as long as t_1 is in the range of $[-0.1, -0.4]$. More higher-order splitting algorithms can be found in Refs. [14–17].

IV. NUMERICAL VERIFICATIONS

To verify the order of convergence of these algorithms, we apply them to the Gross-Pitaevskii equation with a harmonic trap in one dimension (1D),

$$i \frac{\partial \psi}{\partial t} = \left(-\frac{1}{2} \frac{d^2}{dx^2} + \frac{1}{2} \omega^2 x^2 + g |\psi|^2 \right) \psi. \quad (24)$$

To gauge the accuracy of any algorithm, we monitor the fluctuation of the total E ,

$$E = \int_{-\infty}^{\infty} dx \psi^*(t) \left(-\frac{1}{2} \frac{d^2}{dx^2} + \frac{1}{2} \omega^2 x^2 + \frac{1}{2} g |\psi(t)|^2 \right) \psi(t). \quad (25)$$

If the time evolution is exact, E would remain a constant. For $\omega=1$, $g=5$, and $\psi(0)=\psi_0(x)$, the ground-state wave function of the harmonic trap, the initial total energy is

$$E = \frac{1}{2} + \frac{5}{2\sqrt{2\pi}} \approx 1.497\,355\,701. \quad (26)$$

The x interval used is $[-20:20]$ with $2^9=512$ grid points. The results are unchanged if one doubles the number of grid points. In Fig. 1 we plot E as a function of time for algorithm 2A at $\Delta t=0.05$ and $\Delta t=0.025$. One observes that the energy fluctuation at $\Delta t=0.025$ is about 1/4 of that at $\Delta t=0.05$, as befitting a second-order algorithm. The results for fourth-order algorithms FR and M (McLachlan) at $\Delta t=0.05$ are also shown. It is clear that even if one take 1/4 of algorithm 2A's error at $\Delta t=0.025$, corresponding to $\Delta t=0.0125$, that error is still much larger than those of fourth-order algorithm FR and M (i.e., running algorithm 2A 4 times at $\Delta t=0.0125$, using four pairs of FFTs, would still be inferior to algorithm FR which uses only three pairs of FFTs).

In Fig. 2 we greatly magnified the scale so that the fluctuations in the fourth-order algorithms are also visible. This time, when the step size of algorithm FR is half, the error in E is reduced by a factor of 16, confirming the fourth-order convergence of the algorithm. The energy error of algorithm

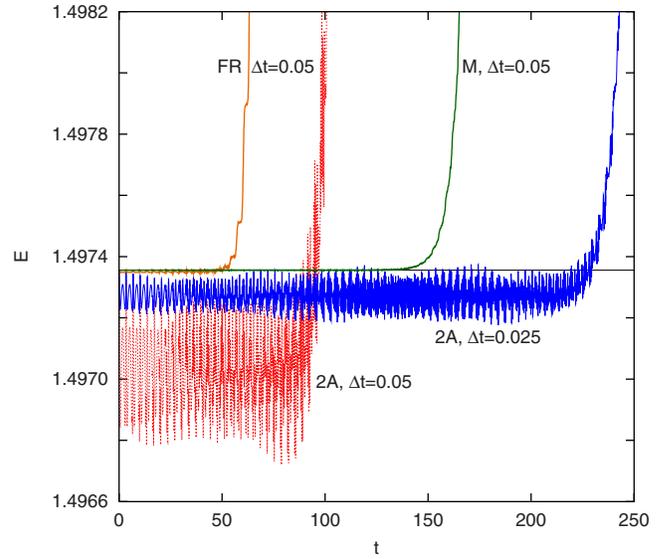


FIG. 1. (Color online) The fluctuation in the total energy E , Eq. (25), when solving the real-time Gross-Pitaevskii equation by second-order algorithm 2A and fourth-order algorithms FR (Forest-Ruth) and M (McLachlan).

M at $\Delta t=0.025$ is $\approx 10^{-6}$, which is too small for a visual comparison.

In both Figs. 1 and 2, the total energy eventually blows up for all calculations, despite the fact that total energy error is only 10^{-6} for McLachlan's algorithm. This instability is directly related to the strength of the nonlinear potential. The rather large value of $g=5$ was chosen so that the instability would show up after a short run. (See further discussion in Sec. VI.)

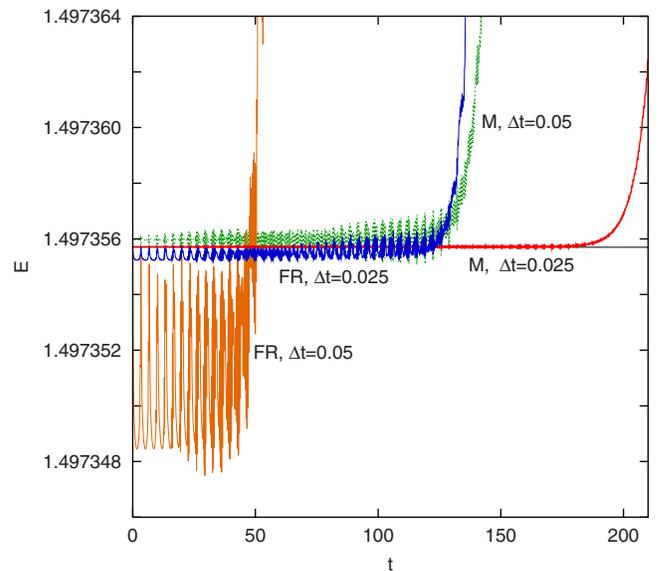


FIG. 2. (Color online) A magnified view of the fluctuation in the total energy of two fourth-order algorithms FR and M at two time-step sizes.

V. CAUSE OF INSTABILITY

The eventual instability as shown in Figs. 1 and 2 demands an understanding of its fundamental cause. To study this, we decompose the general wave function into Fourier components and focus on the propagation of a single component with wave vector p in 1D,

$$\psi(x,t) = A e^{ipx - i\omega t}. \quad (27)$$

This is a solution to Eq. (1) if ω is given by

$$\omega = \frac{1}{2}p^2 + g|A|^2 = E_p + U, \quad (28)$$

where we have denoted $E_p = \frac{1}{2}p^2$ and $U = g|A|^2$. Suppose now the spatial part of ψ is contaminated, due to numerical errors, by very-small-amplitude, sideband wave vectors $p+k$ and $p-k$ so that

$$\psi(x) = A e^{ipx} + a e^{i(p+k)x} + b e^{i(p-k)x}, \quad (29)$$

how will the error amplitudes a and b be propagated by splitting algorithms? (This sideband analysis was inspired by the classical work on Fourier analysis of nonlinearly interacting waves [18].) The effect of $e^{-i\Delta t \hat{T}}$ on $\psi(x)$ is trivial; all amplitudes are multiplied by a phase,

$$\begin{aligned} A' &= e^{-i\Delta t E_p} A, \\ a' &= e^{-i\Delta t E_{p+k}} a, \\ b' &= e^{-i\Delta t E_{p-k}} b. \end{aligned} \quad (30)$$

To compute $e^{-i\Delta t \hat{V}} \psi(x)$, one must compute $|\psi(x)|^2$ using Eq. (29). The result, by keeping terms only to first order in a and b , is

$$\begin{aligned} A' &= e^{-i\Delta t U} A, \\ a' &= e^{-i\Delta t U} [a - i\Delta t (Ua + gA^2 b^*)], \\ b' &= e^{-i\Delta t U} [b - i\Delta t (Ub + gA^2 a^*)]. \end{aligned} \quad (31)$$

Thus the first-order splitting algorithm $e^{-i\Delta t \hat{V}} e^{-i\Delta t \hat{T}} \psi(x)$ modifies the amplitudes by composing Eqs. (30) with Eqs. (31), yielding

$$A_{n+1} = e^{-i\Delta t (E_p + U)} A_n, \quad (32)$$

$$\begin{aligned} a_{n+1} &= e^{-i\Delta t (E_{p+k} - E_k + U)} [a_n e^{-i\Delta t E_k} \\ &\quad - i\Delta t U (a_n e^{-i\Delta t E_k} + (b_n e^{-i\Delta t E_k})^* e^{-i2\delta_n})], \end{aligned} \quad (33)$$

$$\begin{aligned} b_{n+1} &= e^{-i\Delta t (E_{p-k} - E_k + U)} [b_n e^{-i\Delta t E_k} \\ &\quad - i\Delta t U (b_n e^{-i\Delta t E_k} + (a_n e^{-i\Delta t E_k})^* e^{-i2\delta_n})], \end{aligned} \quad (34)$$

where we have defined

$$A_n = |A_n| e^{-i\delta_n}. \quad (35)$$

The algorithm correctly propagates A and preserves the norm $|A|$,

$$A_n = e^{-in\Delta t (E_p + U)} A_0. \quad (36)$$

For notational clarity, we will take A_0 to be real with $\delta_0 = 0$ so that we do not have to keep track of this initial phase, yielding

$$\delta_n = n\Delta t (E_p + U). \quad (37)$$

(Keeping the initial phase simply transfers it to subsequent amplitudes and has no bearing on the issue of instability.) To see the growth in a and b , we factor out their overall phases as follows:

$$\begin{aligned} a_n &= e^{-in\Delta t (E_{p+k} - E_k + U)} \alpha_n, \\ b_n &= e^{-in\Delta t (E_{p-k} - E_k + U)} \beta_n, \end{aligned} \quad (38)$$

and reduce Eqs. (33) and (34) to

$$\alpha_{n+1} = \alpha_n e^{-i\Delta t E_k} - i\Delta t U [\alpha_n e^{-i\Delta t E_k} + (\beta_n e^{-i\Delta t E_k})^*], \quad (39)$$

$$\beta_{n+1} = \beta_n e^{-i\Delta t E_k} - i\Delta t U [\beta_n e^{-i\Delta t E_k} + (\alpha_n e^{-i\Delta t E_k})^*]. \quad (40)$$

These two equations can also be interpreted as a first-order splitting algorithm, with the ‘‘kinetic’’ term giving

$$\begin{aligned} \alpha' &= e^{-i\Delta t E_k} \alpha, \\ \beta' &= e^{-i\Delta t E_k} \beta, \end{aligned} \quad (41)$$

and the ‘‘potential’’ term producing

$$\begin{aligned} \alpha' &= \alpha - i\Delta t U (\alpha + \beta^*), \\ \beta' &= \beta - i\Delta t U (\beta + \alpha^*). \end{aligned} \quad (42)$$

A closer examination reveals that Eqs. (41) and (42) are exact solutions to the following equations:

$$i \frac{d\alpha}{dt} = E_k \alpha, \quad i \frac{d\beta}{dt} = E_k \beta, \quad (43)$$

$$i \frac{d\alpha}{dt} = U (\alpha + \beta^*), \quad i \frac{d\beta}{dt} = U (\beta + \alpha^*). \quad (44)$$

Thus the algorithm is trying to solve the original unsplit equations

$$\begin{aligned} i \frac{d\alpha}{dt} &= (E_k + U) \alpha + U \beta^*, \\ i \frac{d\beta}{dt} &= (E_k + U) \beta + U \alpha^*, \end{aligned} \quad (45)$$

which have general solutions of the form

$$\alpha = c e^{-i\Omega_k t} + d e^{i\Omega_k t}, \quad (46)$$

with

$$\Omega_k = \sqrt{E_k (E_k + 2U)}. \quad (47)$$

This is the famous Bogoliubov spectrum [19] of elementary excitations in a uniform Bose gas. It shows up here because the nonlinear Schrödinger equation is just the Gross-

Pitaevskii equation for describing a uniform Bose-Einstein condensate [20]. The Bogoliubov spectrum in the current context is the background “noise” excitations of the nonlinear Schrödinger equation. If one were able to solve Eqs. (45) exactly via Eq. (46), there would be no instability because the amplitude of α in Eq. (46) is finite. However, when Eqs. (45) are solved by splitting, Eqs. (42) no longer preserve the norm and the modulus of these error terms at selected ranges of k will grow exponentially.

To study this growth, take $\beta_0 = \alpha_0$, so that the splitting forms (41) and (42) simplify to

$$\alpha' = e^{-i\Delta t E_k} \alpha, \quad (48)$$

$$\alpha' = \alpha - i\Delta t U(\alpha + \alpha^*). \quad (49)$$

Now we assert without giving a detailed proof that beyond first order, for any splitting algorithm in solving the nonlinear Schrödinger equation, the error Fourier components will grow correspondingly according to splitting Eqs. (48) and (49) with the same splitting coefficients. For example, corresponding to algorithm 2A, the growth of the error Fourier components is given by

$$\begin{aligned} \alpha_1 &= \alpha_0 - i\frac{1}{2}\Delta t U(\alpha_0 + \alpha_0^*), \\ \alpha_2 &= e^{-i\Delta t E_k} \alpha_1, \\ \alpha_3 &= \alpha_2 - i\frac{1}{2}\Delta t U(\alpha_2 + \alpha_2^*). \end{aligned} \quad (50)$$

The subscripts here simply label the individual steps in the algorithm. The last labeled value is the updated variable after one time step. Denoting this updating as $\mathcal{E}_{2A}(\Delta t)$, the error growth of the Forest-Ruth algorithm is then

$$\mathcal{E}_{FR}(\Delta t) = \mathcal{E}_{2A}(c_1 \Delta t) \mathcal{E}_{2A}(c_0 \Delta t) \mathcal{E}_{2A}(c_1 \Delta t) \quad (51)$$

and McLachlan’s algorithm as

$$\begin{aligned} \alpha_1 &= \alpha_0 - i(t_2 \Delta t) U(\alpha_0 + \alpha_0^*), \\ \alpha_2 &= e^{-i(v_2 \Delta t) E_k} \alpha_1, \\ \alpha_3 &= \alpha_2 - i(t_1 \Delta t) U(\alpha_2 + \alpha_2^*), \\ \alpha_4 &= e^{-i v_1 \Delta t E_k} \alpha_3, \\ &\dots \end{aligned} \quad (52)$$

To verify the validity of our assertion, we run the normal algorithm on an initial wave function having the $p=0$ component with amplitude $A=1$ and all other Fourier components set to e^{-25} at $g=5$ and $\Delta t=0.2$. The resulting Fourier amplitudes are then outputted every time steps for seven time steps. Their modulus are shown as plus signs for the above three algorithms in Figs. 3–5. Instead of plotting the magnitude of these Fourier amplitudes as a function of k , we plot them as a function of $\Delta t E_k / \pi$, which is more revealing. Also plotted as solid lines, are the predicted error amplitudes

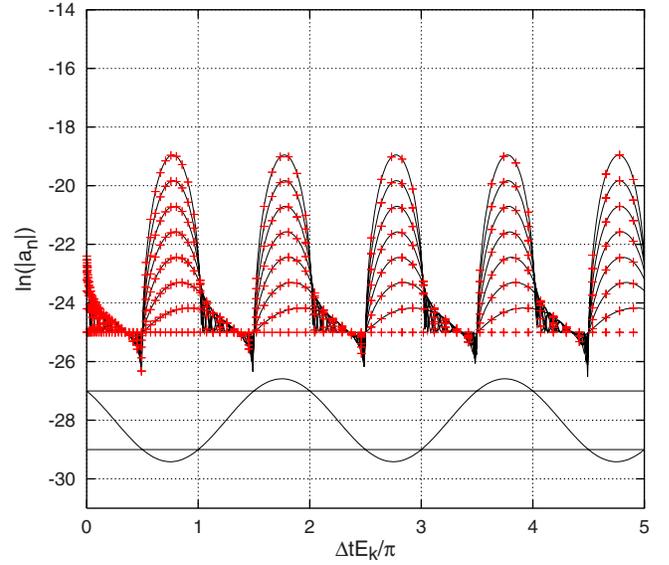


FIG. 3. (Color online) The growth of the error Fourier amplitudes due to algorithm 2A for seven time steps at $g=5$ and $\Delta t=0.2$. The plus signs denotes the algorithm’s actual output; the seven solid lines are the predicted error from the sideband analysis (50) for seven time steps. Centered on -28 is the algorithm’s C function for predicting regions of stability and instability.

given by Eqs. (50)–(52) for seven time steps. The perfect agreement in all three cases confirms our assertion and our sideband analysis.

To understand the pattern of instability as shown in Figs. 3–5, we rewrite the splitting forms (48) and (49) as matrices acting on the real and imaginary part of α ,

$$\begin{pmatrix} \alpha'_R \\ \alpha'_I \end{pmatrix} = \mathbf{T}(\Delta t) \begin{pmatrix} \alpha_R \\ \alpha_I \end{pmatrix}, \quad \begin{pmatrix} \alpha'_R \\ \alpha'_I \end{pmatrix} = \mathbf{V}(\Delta t) \begin{pmatrix} \alpha_R \\ \alpha_I \end{pmatrix}, \quad (53)$$

with

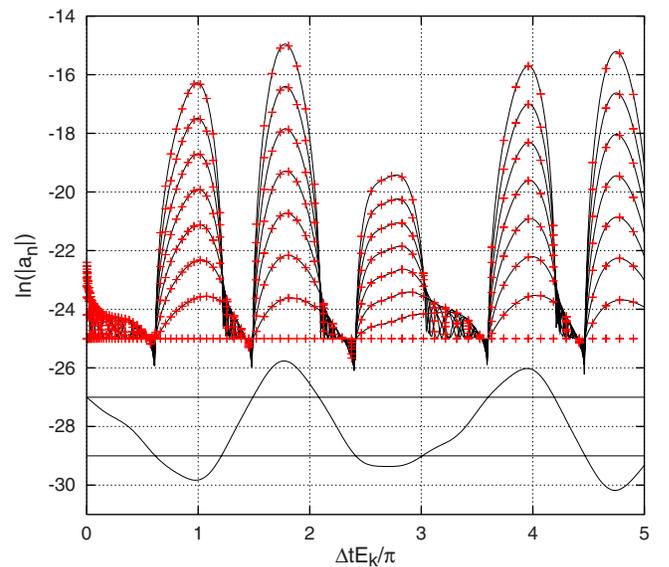


FIG. 4. (Color online) Same as Fig. 3 but for the Forest-Ruth algorithm. The predicted error is given by Eq. (51).

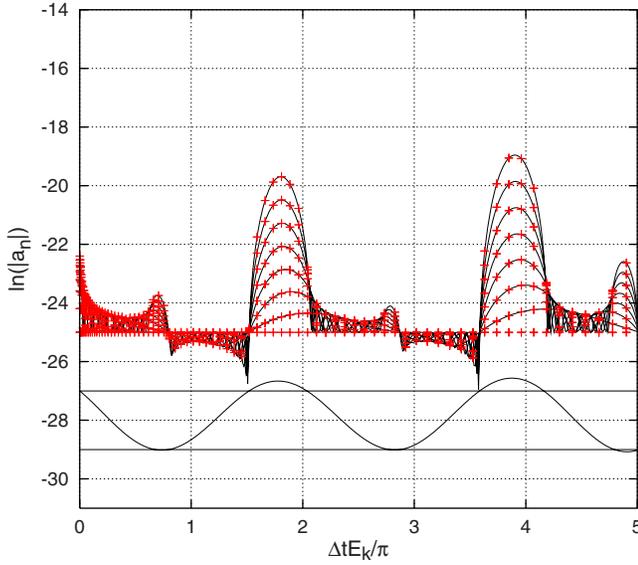


FIG. 5. (Color online) Same as Fig. 3 but for McLachlan's algorithm. The predicted error is given by (52).

$$\mathbf{T}(\Delta t) = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}, \quad \mathbf{V}(\Delta t) = \begin{pmatrix} 1 & 0 \\ -2u & 1 \end{pmatrix}, \quad (54)$$

and where we have defined

$$c = \cos(x), \quad s = \sin(x), \quad x = \Delta t E_k, \quad u = \Delta t U. \quad (55)$$

The updating matrix corresponding to algorithm 2A is therefore

$$\begin{aligned} \mathbf{M}_{2A}(\Delta t) &= \mathbf{V}\left(\frac{1}{2}\Delta t\right)\mathbf{T}(\Delta t)\mathbf{V}\left(\frac{1}{2}\Delta t\right) \\ &= \begin{pmatrix} c - us & s \\ (u^2 - 1)s - 2uc & c - us \end{pmatrix}. \end{aligned} \quad (56)$$

This is a special form of a matrix with equal diagonal elements and unit determinant. This is due to the left-right symmetric form of the matrix product (i.e., the algorithm is time reversible [21]) and that both \mathbf{T} and \mathbf{V} have unit determinant. Such a matrix has the special property that its eigenvalue is given by

$$e_{1,2} = C \pm \sqrt{C^2 - 1}, \quad (57)$$

where C is just the diagonal element (or half of the trace of the matrix). If $|C| < 1$, the eigenvalues are complex with unit modulus and the algorithm is stable. If $|C| > 1$, the eigenvalues are real with one eigenvalue always greater than unity. Thus by just plotting C against $x = \Delta t E_k$, one can immediately determine the regions of instability. For algorithm 2A, we have

$$C(x) = \cos(x) - u \sin(x) = C_0 \cos(x + \delta), \quad (58)$$

with

$$C_0 = \sqrt{1 + u^2} \quad \text{and} \quad \delta = \tan^{-1} u. \quad (59)$$

It is then immediately clear that as long as $u \neq 0$, the algorithm is unstable for x in the interval $[n\pi - 2\delta, n\pi]$ where $n = 1, 2, 3, \dots$. At a fixed U , decreasing Δt reduces u and δ , and hence the width of the instability region, but does not remove the instability (but see further discussion in the next section). In Fig. 3, this C function is plotted and lowered to -28 so that the interval where $|C(x)| > 1$ can be directly compared with the observed regions of instability. The peak instability occurs at $x = n\pi - \delta$ with the maximum eigenvalue

$$|e_{1,2}| = \sqrt{1 + u^2} + \sqrt{u}. \quad (60)$$

For $\Delta t = 0.2$ and $U = 5$, we have $u = 1$, $\delta = \pi/4$, and $|e| = 1 + \sqrt{2}$. After seven iterations, the e -fold increase of the peaks would be $\ln[(1 + \sqrt{2})^7] = 6.16962$, which is the six e -fold increase of amplitude observed in Fig. 3. Thus we have completely accounted for, both qualitatively and quantitatively, the pattern of instability as shown in Fig. 3. The corresponding C functions for the Forest-Ruth and McLachlan algorithms are also plotted in Figs. 4 and 5. Their C functions are too lengthy for a written display. (The analytical expression for McLachlan's C function is more than a page long using Mathematica.)

By comparing Figs. 3 and 4, one sees that the Forest-Ruth algorithm has a greater error growing rate than 2A. We will see in the next section that this is precisely the reason why the FR algorithm blew up earlier than 2A in Fig. 1. Finally, as shown in Fig. 5, McLachlan's algorithm manages to shift the C function is such a way that the error peaks at $x/\pi = 1, 3$ are nearly eliminated.

Further insights into the origin of this instability can be gained by representing $\mathbf{T}(\Delta t)$ and $\mathbf{V}(\Delta t)$ in terms of traceless matrices,

$$\begin{aligned} \mathbf{T}(\Delta t) &= \exp\left[\Delta t \begin{pmatrix} 0 & E_k \\ -E_k & 0 \end{pmatrix}\right], \\ \mathbf{V}(\Delta t) &= \exp\left[\Delta t \begin{pmatrix} 0 & 0 \\ -2U & 0 \end{pmatrix}\right]. \end{aligned} \quad (61)$$

One can then immediately identify the unsplitted evolution operator as

$$\exp\left[\Delta t \begin{pmatrix} 0 & E_k \\ -E_k - 2U & 0 \end{pmatrix}\right] = \begin{pmatrix} \cos(\Omega_k \Delta t) & \sin(\Omega_k \Delta t) \\ -\sin(\Omega_k \Delta t) & \cos(\Omega_k \Delta t) \end{pmatrix}, \quad (62)$$

which is that of a harmonic oscillator with the Bogoliubov spectrum Ω_k . Were one able to split it alternatively as

$$\begin{aligned} \mathbf{T}'(\Delta t) &= \exp\left[\Delta t \begin{pmatrix} 0 & E_k \\ 0 & 0 \end{pmatrix}\right], \\ \mathbf{V}'(\Delta t) &= \exp\left[\Delta t \begin{pmatrix} 0 & 0 \\ -E_k - 2U & 0 \end{pmatrix}\right], \end{aligned} \quad (63)$$

one would recover the stability criterion normally associated with the harmonic oscillator. For example, the corresponding

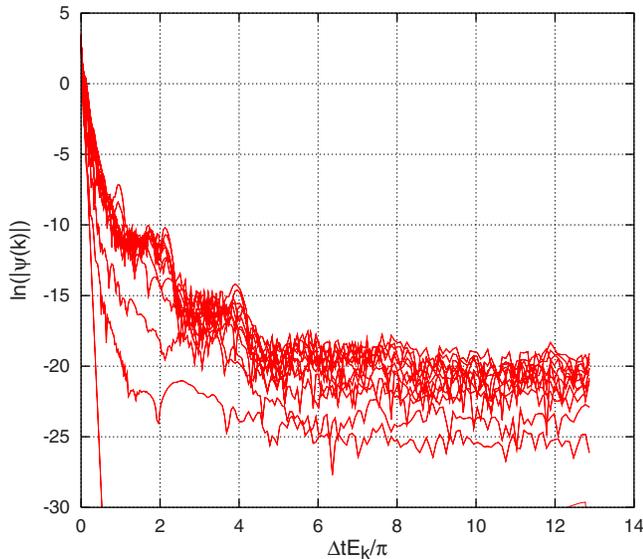


FIG. 6. (Color online) The modulus of the Gross-Pitaevskii momentum wave function $|\psi(k)|$ at every 100th time step due to algorithm 2A. The time-step size is $\Delta t=0.05$.

second-order algorithm 2A, $\mathbf{V}'(\frac{1}{2}\Delta t)\mathbf{T}'(\Delta t)\mathbf{V}'(\frac{1}{2}\Delta t)$, would then yield a C function of

$$C = 1 - \frac{1}{2}\Omega_k^2\Delta t^2, \quad (64)$$

which limits stability to $\Delta t \leq 2/\Omega_k$, a well-known result. This limit is actually worse than $x \leq \pi - 2\delta$, which, as $U \rightarrow 0$, is $\Delta t \leq \pi/E_k$. Our original splitting Eq. (61) is therefore better than the usual harmonic oscillator splitting Eq. (63). Moreover, in contrast to Fig. 3, the usual harmonic oscillator splitting would have *no* stable region whatsoever beyond $\Delta t E_k \geq \pi$!

In this section we have shown that the error-growing pattern of any splitting algorithms when solving the nonlinear Schrödinger can be analytically understood. The instability is due to the exponential amplification of high- k noises at $E_k \geq \pi/\Delta t$.

VI. INSTABILITY OF THE GROSS-PITAEVSKII WAVE FUNCTION

We now repeat the calculations of Fig. 1 at $\Delta t=0.05$ for 1200 time steps to the point where the algorithm FR begins to blow up. We plot in Figs. 6–8, the modulus of the k -space wave function $|\psi(k)|$ as a function of $\Delta t E_k/\pi$ at every 100th time step. The initial Gaussian wave function is the straight line seen plunging down close to vertical axis. Because of limited numerical precision, that line levels off to some random values around $e^{-35} \approx 10^{-16}$ at high E_k . These are the initial random errors of the wave function. When the algorithm acts on the wave function, these random errors are amplified successively and grow in time. For algorithm 2A, Fig. 6 shows error peaks at $x/\pi=1, 2$, and 4, which is in agreement with Fig. 3, but no discernable peak is seen near $x/\pi=3$. For the Forest-Ruth algorithm, Fig. 7 shows a prome-

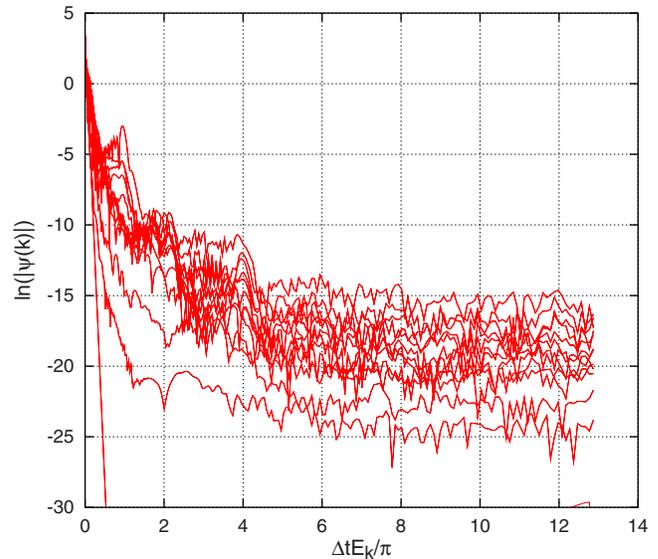


FIG. 7. (Color online) The modulus of the Gross-Pitaevskii momentum wave function $|\psi(k)|$ at every 100th time step due to Forest-Ruth algorithm. This is the momentum wave function corresponding to the energy calculation of Fig. 1 up to $t=60$.

nient peak at $x/\pi=1$, followed by a peak-shoulder structure at $x/\pi=2$ and 4, in agreement with Fig. 4. For McLachlan’s algorithm, Fig. 8 shows that the error peak at $x/\pi=1$ is conspicuously absent and only peaks at $x/\pi=2, 4$ are visible. This is in excellent agreement with the predicted error structure of Fig. 5. In the case of the Forest-Ruth algorithm, the error peak at $x/\pi=1$ has grown sufficiently to distort the wave function and cause the energy to blow up. These exponentially growing error peaks are like ticking time bombs, harmless at first, but eventually overwhelming and destroying the wave function.

For a continuum wave function, this instability is unavoidable as long as Δt is finite. However, for a discrete

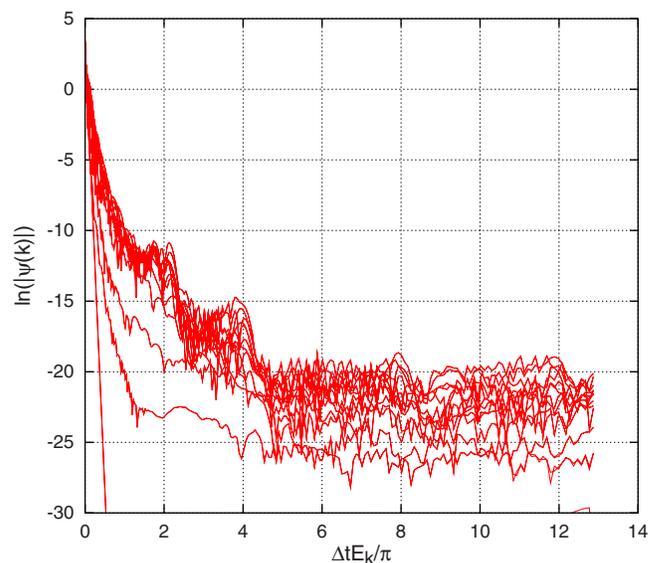


FIG. 8. (Color online) Same as Fig. 7 for McLachlan’s algorithm.

wave function defined at only N grid points, there is a loop hole. For a finite- N -point calculation, the maximum k vector is $k_{max}=N\pi/L$ so that $\Delta t E_k/\pi$ extends only out to $(0.05)0.5(512\pi/40)^2/\pi \approx 12.9$, as shown in Figs. 6–8. Thus one can take advantage of this and force stability by making Δt so small that

$$\Delta t E_k^{max} < x_{min}, \quad (65)$$

where x_{min} is the smallest value of x such that $|C(x)|=1$ and $E_k^{max} = \frac{1}{2}k_{max}^2$. For most algorithms at small Δt , $x_{min} \approx \pi$. This criterion (65) simply shrinks the entire range of E_k values to below the first instability point. Thus the FR calculation would be stable for $\Delta t < \pi/[0.5(512\pi/40)^2]=0.0039$. A more refined calculation at higher N would require an even smaller Δt . Such a small Δt would make long-time simulations very time consuming. On the other hand, Eq. (65) also implies that stability can be achieved by lowering k_{max} —i.e., using fewer grid points. For example, at $N=128$, $\pi/[0.5(128\pi/40)^2]=0.062$. When the FR algorithm is rerun at $\Delta t=0.05$ but with $N=128$, the total energy is indeed stable out to $t=300$. However, the wave function now looked very jagged. Thus, for long-time simulations, one must choose Δt and N judiciously.

The instability observed here is very similar to the “resonance” instability of multiple-time-step algorithms used in biomolecular simulations [22]. There, stability requires that $\Delta t < \pi/\omega$, where ω is the faster physical frequency in the problem. The latency in the energy blowup has also been observed in density functional calculations using split algorithms [23]. The energy blowup there is more gradual, but it is undoubtedly related to the nonlinear Kohn-Sham density used, for which the nonlinear Schrödinger equation is the simplest prototype.

VII. CONCLUSIONS

In this work we have shown how splitting algorithms of any order can be devised to solve the nonlinear Schrödinger equation. The key ingredient is the exact solution of the potential equation (4), as pointed out earlier by Taha and Ablowitz [1]. This explains Javanainen and Ruostekoski’s finding [3] without the need to construct Strauch’s special operator [4]. Solution (4) clearly generalizes to the case where $g|\psi|^2 \rightarrow v(|\psi|)$, implying that this class of general nonlinear equations can also be solved by splitting algorithms.

In the course of verifying these algorithms by solving the Gross-Pitaevskii equation, a latent instability is observed in all the algorithms. This instability persists regardless of the order of the algorithm and despite excellent total energy conservation. A detail error analysis reveals that this instability is intrinsic to splitting algorithms and can only be avoided if Eq. (65) is satisfied.

The main advantage of higher-order algorithms is that a larger Δt can be used for more efficient simulations. However, the stability criterion (65) dictates a small Δt regardless of order, thus negating much of the presumed advantage of using higher-order algorithms. (Of course, higher-order algorithms are useful for short-time simulations, where results can be obtained prior to the blowup.) This work also suggests that one must not use just any higher-order algorithm, such as FR, but a higher-order algorithm with a higher x_{min} , such as McLachlan’s algorithm. How algorithms can be derived systematically with a higher x_{min} is a fitting subject for future study.

ACKNOWLEDGMENT

I thank E. Krotscheck for many valuable discussions over the years on this subject.

-
- [1] T. R. Taha and M. J. Ablowitz, *J. Comput. Phys.* **55**, 203 (1984).
 [2] A. D. Bandrauk and H. Shen, *J. Phys. A* **27**, 7147 (1994).
 [3] J. Javanainen and J. Ruostekoski, *J. Phys. A* **39**, L179 (2006).
 [4] F. W. Strauch, *Phys. Rev. E* **76**, 046701 (2007).
 [5] M. Suzuki, *Proc. Jpn. Acad., Ser. B: Phys. Biol. Sci.* **69**, 161 (1993).
 [6] S. A. Chin and C. R. Chen, *J. Chem. Phys.* **117**, 1409 (2002).
 [7] S. K. Adhikari and P. Muruganandam, *J. Phys. B* **35**, 2831 (2002).
 [8] M. Creutz and A. Gocksch, *Phys. Rev. Lett.* **63**, 9 (1989).
 [9] E. Forest and R. D. Ruth, *Physica D* **43**, 105 (1990).
 [10] M. Suzuki, *Phys. Lett. A* **146**, 319 (1990).
 [11] H. Yoshida, *Phys. Lett. A* **150**, 262 (1990).
 [12] R. I. McLachlan, *SIAM (Soc. Ind. Appl. Math.) J. Sci. Stat. Comput.* **16**, 151 (1995).
 [13] S. A. Chin, *Intl. J. Compt. Math.* **84**, 729 (2007).
 [14] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration* (Springer-Verlag, Berlin, 2002).
 [15] R. I. McLachlan and G. R. W. Quispel, *Acta Numerica* **11**, 241 (2002).
 [16] S. Blanes and P. C. Moan, *J. Comput. Appl. Math.* **142**, 313 (2002).
 [17] B. Leimkuhler and S. Reich, *Simulating Hamiltonian Dynamics* (Cambridge University Press, Cambridge, England, 2004).
 [18] G. B. Whitham, *Linear and Nonlinear Waves* (John Wiley and Sons, New York, 1974), pp. 527–532.
 [19] N. N. Bogoliubov, *J. Phys. (USSR)* **11**, 23 (1947), reprinted in D. Pine, *The Many-Body Problem* (Benjamin, New York, 1961), p. 292.
 [20] C. J. Pethick and H. Smith, *Bose-Einstein Condensation in Dilute Gases* (Cambridge University Press, Cambridge, England, 2002), p. 172.
 [21] S. A. Chin and S. R. Scuro, *Phys. Lett. A* **342**, 397 (2005).
 [22] S. A. Chin, *J. Chem. Phys.* **120**, 8–13 (2004).
 [23] O. Sugino and Y. Miyamoto, *Phys. Rev. B* **59**, 2579 (1999).