Efficient explicit numerical solutions of the time-dependent Schrödinger equation

W. van Dijk[*]
Department of Physics and Astronomy, McMaster University, Hamilton, Ontario, Canada L8S 4M1
and Redeemer University, Ancaster, Ontario, Canada L9K 1J4

(Received 24 August 2021; accepted 20 January 2022; published 4 February 2022)

Explicit numerical solutions of the time-dependent Schrödinger equation are more efficient than those obtained by commonly used implicit approaches. They are more practical, especially for a system with higher spatial dimensions. To that end, we introduce a generalization of an explicit three-level method to obtain solutions with spatial and temporal errors of the order of $O[(\Delta x)^{2r}]$ and $O[(\Delta t)^{2M+3}]$, where Δx and Δt are the spatial and temporal grid elements, and r and M are positive integers. Sample calculations illustrate the efficacy and stability of the algorithm.

DOI: 10.1103/PhysRevE.105.025303

I. INTRODUCTION

Since solutions of the time-dependent Schrödinger equation are fundamental to understanding the behavior of nonrelativistic quantum systems, methods of numerical solutions as well as improvements on existing methods continue to be explored. For example, we have investigated a generalization of the Crank-Nicolson method [1] and compared that to an efficient method involving the Chebyshev expansion of the time-evolution operator [2–4]. The former generalization involves replacing the time-evolution operator by a factorized Padé approximant and ordering the factors so that at each stage, unitarity is maintained. This implicit approach calls for solutions of systems of J linear algebraic equations in J unknowns, where J is the number of spatial grid points. The advantage of the generalized Crank-Nicolson approach is that it preserves unitarity and is unconditionally stable. Spatial integration may be accomplished by spectral, split-operator, difference, or compact-high-order difference methods, as well as other approaches [4].

The explicit, rather than implicit, methods basically involve the determination of the wave function at a time in terms of an algebraic expression of the wave function at earlier times. The simplest forward-time-centered-space algorithm, which is explicit and expresses the wave function at an instant in terms of the wave function a time step earlier, is unconditionally unstable [5]. A noteworthy modification of this procedure, so that the wave function at a particular time is expressed in terms of wave functions at two earlier times, permits a stable scheme. This technique was discussed by Askar and Cakmak [6]; subsequently, it was pointed out [7] that the method was originally proposed by Harmuth [8].

In this paper, we exploit this approach to obtain a numeric wave function that is accurate in time to the order of $(\Delta t)^{2M+3}$ for some non-negative integer M. If, furthermore, we use a generalized forward-time-centered-space expansion, the results will also be accurate to the order of $(\Delta x)^{2r}$ for a positive integral value of r. The approximation involves a polynomial expansion of the time-evolution operator which is factored into binomials. Even though unitarity is not manifestly preserved, this leads to an explicit solution at each stage, with the obvious advantage of a much reduced number of basic operations and a commensurate gain in computational speed compared with the generalized Crank-Nicolson method. As it turns out, the calculations can be very precise, and the lack of assured unitarity may be used to advantage to monitor the stability and reliability of the algorithm.

The method, its stability, and accuracy of the solution are discussed in Sec. II, as well as extensions to higher spatial dimensions. Section III consists of examples that illustrate the efficacy of the approach; conclusions and comments are found in Sec. IV.

II. METHOD OF INTEGRATIONS

The aim is to develop a straightforward, explicit, but accurate, method of solving the time-dependent initial-value Schrödinger equation,

$$\left(i\hbar\frac{\partial}{\partial t} - H\right)\psi(\mathbf{r},t) = 0, \quad \psi(\mathbf{r},t_{\text{init}}) = \phi(\mathbf{r}), \tag{2.1}$$

with the time-independent Hamiltonian

$$H = -\frac{\hbar^2}{2m}\nabla^2 + V(\mathbf{r}), \tag{2.2}$$

where $\phi(\mathbf{r})$ is the wave function at initial time t_{init}. The behavior of the system in time can be expressed in terms of the time-evolution operator acting on the wave function at time t, which yields the wave function at time $t + \Delta t$, i.e.,

$$\psi(\mathbf{r}, t + \Delta t) = e^{-iH\Delta t/\hbar}\psi(\mathbf{r}, t). \tag{2.3}$$

The simplest, lowest-order expansion of the time-evolution operator gives

$$e^{-iH\Delta t/\hbar} = 1 - iH\Delta t/\hbar + O[(\Delta t)^2]. \tag{2.4}$$

[*] vandijk@physics.mcmaster.ca

In this approximation, the wave function evolves according to the forward Euler method,

$$\psi(\mathbf{r}, t+\Delta t) = (1 - iH\Delta t/\hbar)\psi(\mathbf{r}, t). \tag{2.5}$$

This leads to an explicit solution since the evolved wave function is simply an operator acting on the wave function at an earlier time. Unfortunately, if Eq. (2.5) is implemented with a forward-time-centered-space numerical procedure, it is found to be unconditionally unstable [5].

According to Askar and Cakmak [6], if the procedure is extended so that it involves two distinct time steps, or levels, it can lead to conditional stability. Thus, combining Eq. (2.3) with

$$\psi(\mathbf{r}, t-\Delta t) = e^{iH\Delta t/\hbar}\psi(\mathbf{r}, t), \tag{2.6}$$

they obtain

$$\begin{aligned}\psi(\mathbf{r}, t+\Delta t) &= \psi(\mathbf{r}, t-\Delta t) - (e^{iH\Delta t/\hbar} - e^{-iH\Delta t/\hbar})\psi(\mathbf{r}, t)\\ &= \psi(\mathbf{r}, t-\Delta t) - 2i\Delta t H/\hbar\psi(\mathbf{r}, t) + O[(\Delta t)^3].\end{aligned} \tag{2.7}$$

Not only is the possibility of stability secured, the lowest-order approximation has errors of the order of $(\Delta t)^3$ rather than $(\Delta t)^2$.

A. Basic method in one spatial dimension

In order to approximate the solution in one spatial dimension, we discretize the wave function so that $\psi_j^n = \psi(x_j, t_n)$, where the spatial domain $[x_0, x_J]$ is partitioned into intervals $[x_{j-1}, x_j]$ with $x_0 < x_1 < \cdots < x_J$ and $x_{j+1} - x_j = \Delta x$ for $j = 0, 1, \ldots, J-1$. The time is partitioned so that $t_n = n\Delta t$, $n = 0, 1, 2, \ldots, N$. We will also adopt the notation of column vectors $\Psi^n = (\psi_0^n, \psi_1^n, \ldots, \psi_J^n)^T$ to represent the wave function at time t_n. Since the Hamiltonian has the second-order spatial derivative, we approximate the latter with central differences as [1]

$$\frac{\partial^2\psi_j^n}{\partial x^2} = \frac{1}{(\Delta x)^2}\sum_{\ell=-r}^{r} c_\ell^{(r)}\psi_{j+\ell}^n + O[(\Delta x)^{2r}], \tag{2.8}$$

where $c_\ell^{(r)}$ are real rational constants independent of Δx. A simple scheme of obtaining the $c_\ell^{(r)}$ involves the identity

$$\begin{aligned}h^2 f^{(2)}(x) &= \sum_{\ell=-r}^{r} c_\ell^{(r)} f(x+\ell h) = \sum_{\ell=-r}^{r} c_\ell^{(r)}\sum_{i=0}^{\infty}\frac{(\ell h)^i}{i!}f^{(i)}(x)\\ &= \sum_{i=0}^{2r}\frac{h^i}{i!}f^{(i)}(x)\sum_{\ell=-r}^{r}\ell^i c_\ell^{(r)} + O(h^{2r+2}),\end{aligned} \tag{2.9}$$

where superscripts on f refer to the order of the derivative. Equating coefficients of the terms with the same order derivative on both sides of the equation, and noting that $c_{-\ell}^{(r)} = c_\ell^{(r)}$, we determine the $c_\ell^{(r)}$ by solving equations [9]

$$-\frac{c_0^{(r)}}{2}\delta_{0i} + \delta_{2i} = \sum_{\ell=1}^{r}\ell^{2i}c_\ell^{(r)} \quad \text{for} \quad i = 0, 1, \ldots, r. \tag{2.10}$$

This approach can be generalized in a straightforward manner for nonuniform spatial grids; see, for example, Ref. [10, pp. 16-17]. A useful relation is the $i = 0$ equation, which expresses $c_0^{(r)}$ in terms of the other coefficients,

$$c_0^{(r)} = -2\sum_{\ell=1}^{r} c_\ell^{(r)}. \tag{2.11}$$

TABLE I. The coefficients $c_\ell^{(r)}$ up to $r = 6$.

r	$\ell = 0$	1	2	3	4	5	
1	-2	1					
2	$-\frac{5}{2}$	$\frac{4}{3}$	$-\frac{1}{12}$				
3	$-\frac{49}{18}$	$\frac{3}{2}$	$-\frac{3}{20}$	$\frac{1}{90}$			
4	$-\frac{205}{72}$	$\frac{8}{5}$	$-\frac{1}{5}$	$\frac{8}{315}$	$-\frac{1}{560}$		
5	$-\frac{5269}{1800}$	$\frac{5}{3}$	$-\frac{5}{21}$	$\frac{5}{126}$	$-\frac{5}{1008}$	$\frac{1}{3150}$	
6	$-\frac{5369}{1800}$	$\frac{12}{7}$	$-\frac{15}{56}$	$\frac{10}{189}$	$-\frac{1}{112}$	$\frac{2}{1925}$	$-\frac{1}{16632}$
$\vdots$	$\cdots$						

The lowest-order coefficients are listed up to $r = 6$ in Table I. In terms of matrix terminology, the second derivative of the wave function is the product of a $(2r+1)$-diagonally banded matrix and the discrete vector form of the wave function.

The lowest-order in time expression using Eq. (2.5) with $V_j = V(x_j)$ is

$$\psi_j^{n+1} = \psi_j^n - i\frac{\Delta t}{\hbar}\left[-\frac{\hbar^2}{2m(\Delta x)^2}\sum_{\ell=-r}^{r} c_\ell^{(r)}\psi_{j+\ell}^n + V_j\psi_j^n\right]. \tag{2.12}$$

As we show later, this algorithm is unconditionally unstable.

Similarly, the lowest-order approximation in time of Eq. (2.7) is

$$\psi_j^{n+1} = \psi_j^{n-1} - 2i\frac{\Delta t}{\hbar}\left[-\frac{\hbar^2}{2m(\Delta x)^2}\sum_{\ell=-r}^{r} c_\ell^{(r)}\psi_{j+\ell}^n + V_j\psi_j^n\right], \tag{2.13}$$

which leads to a conditionally stable procedure [6,11]. We therefore generalize it to obtain an algorithm of higher order of Δt as well as Δx. To achieve this, we rewrite Eq. (2.7) as

$$\psi(x, t+\Delta t) = \psi(x, t-\Delta t) - 2i\sin(H\Delta t/\hbar)\psi(x, t), \tag{2.14}$$

and consider the expansion of the sine function,

$$\begin{aligned}\sin(z) &= z\left(1 - \frac{1}{3!}z^2 + \frac{1}{5!}z^4 - \cdots + \frac{(-1)^M}{(2M+1)!}z^{2M}\right)\\ &\quad + O(z^{2M+3})\\ &= z\prod_{s=1}^{2M}\left(1 - \frac{z}{z_s^{(2M)}}\right) + O(z^{2M+3}),\end{aligned} \tag{2.15}$$

where the $z_1^{(2M)}, z_2^{(2M)}, \ldots, z_{2M}^{(2M)}$ are the $2M$ zeros of the $2M$th-order polynomial approximation of $\sin z/z$. These can be calculated to the precision needed, a sample of which for $M = 1$ to 3, each rounded to five decimal places, is given in Table II. We use the operator form of the expansion of $\sin(H\Delta t/\hbar)$ to derive a multilevel time algorithm. Define the

TABLE II. The zeros $z_s^{(2M)}$ of the polynomial approximation of $\dfrac{\sin z}{z}$ for $M = 1, 2, 3$, etc.

M	$s = 1$	2	3	4	5	6
1	$-2.44949 + i0.00000$	$244949 + i0.00000$				
2	$3.23685 - i0.69082$	$-3.23685 + i0.69082$	$3.23685 + i0.69082$	$-3.23685 - i0.69082$		
3	$3.07864 + i0.00000$	$-3.07864 + i0.00000$	$4.43401 - i184375$	$-4.43401 + i184375$	$4.43401 + i184375$	$-4.43401 - i184375$
$\vdots$	$\dots$					

operators

$$K_s^{(2M)} \equiv 1 - \frac{H\Delta t/\hbar}{z_s^{(2M)}} \quad \text{for} \quad s = 1, 2, \dots, 2M, \tag{2.16}$$

and, for notational convenience to include the case for which $M = 0$, we also define $K_0^{(2M)} = 1$. In terms of the discretized time, we write Eq. (2.14) as

$$\Psi^{n+1} = \Psi^{n-1} - 2iH\Delta t/\hbar \prod_{s=0}^{2M} K_s^{(2M)} \Psi^n. \tag{2.17}$$

For $M > 0$ given Ψ^n and Ψ^{n-1}, we calculate recursively

$$\Psi^{n,s+1} = K_s^{(2M)} \Psi^{n,s} \quad \text{with} \quad \Psi^{n,0} = \Psi^n, \tag{2.18}$$

for $s = 0, 1, \dots, 2M - 1$, and then evaluate

$$\Psi^{n+1} = \Psi^{n-1} - 2iH\Delta t/\hbar \Psi^{n,2M}. \tag{2.19}$$

The average values of $|z_s^{(2M)}|$ become progressively larger with increasing M, as shown in Fig. 1. Although a greater number of steps is to be calculated as M increases, the size of the time element can be made larger for similar convergence to offset, at least in part, the greater number of arithmetical operations.

It should be pointed out that using Eq. (2.17) as the basic operation requires a numerical determination of $H\Psi$. We have chosen central differencing for the determination of the kinetic energy term, which is convenient for the error analysis. Instead, one could use high-order-compact differencing [12], a fast-Fourier spectral approach [4], differencing on a nonuniform spatial grid [10], or some other approximation of the second-order spatial derivative.

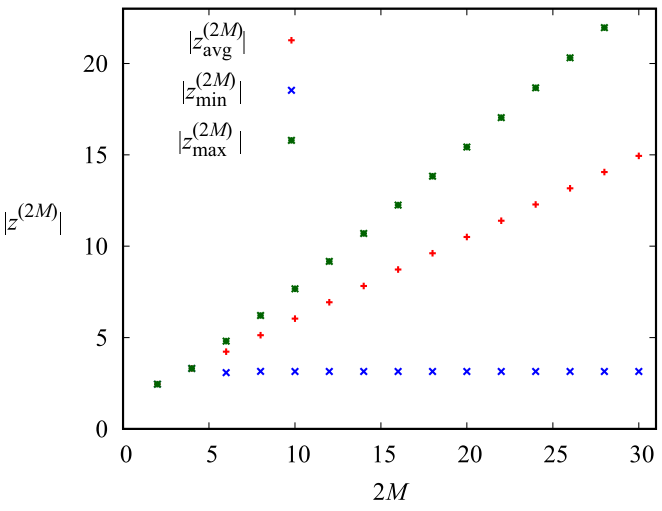

FIG. 1. The average, minimum, and maximum values of $|z_s^{(2M)}|$ for each value of $2M$.

B. Stability

The usefulness of any numerical solution of a partial differential equation depends on its stability, i.e., its ability to avoid amplification of the errors over time.

Consider the lowest-order algorithm of Eq. (2.12). According to von Neumann's stability analysis [5], we can substitute the expression for a Fourier mode,

$$\psi_j^n = \xi^n(k, \Delta t, \Delta x)e^{ikx_j}, \tag{2.20}$$

for any real wave number k in Eq. (2.12) to obtain the growth factor

$$\xi = 1 - ia, \tag{2.21}$$

where

$$a = \frac{\Delta t}{\hbar}\left[\frac{2\hbar^2}{m(\Delta x)^2}\sum_{\ell=1}^{r} c_\ell^{(r)} \sin^2(\ell k\Delta x/2) + V_j\right]. \tag{2.22}$$

Since a is real, the magnitude of the growth factor is greater than one for all k, Δx, Δt; hence the scheme is unconditionally unstable. Strictly speaking, unless the potential is constant, the growth factor depends on j. For all j, however, the growth factor is greater than one and the algorithm is unstable. See, for example, a similar analysis to show the stability of the Crank-Nicolson method in Ref. [5, Sec. 19.2].

Inserting the expression for the Fourier mode in the modified algorithm [Eq. (2.13)], we obtain an equation for the growth factor [6],

$$\xi^2 + 2ia\xi - 1 = 0, \tag{2.23}$$

with solutions $\xi = -ia \pm \sqrt{1 - a^2}$. Thus, for $a^2 \leqslant 1$, $|\xi| = 1$, and for $a^2 > 1$, at least one of the roots yields $|\xi| > 1$. Hence the algorithm is stable when $a^2 \leqslant 1$, and unstable otherwise.

A recent article by Kumari and Donzis [13] generalizes the von Neumann stability analysis to include the possibility that the growth factor depends on time. In our scheme, the time step from $t = n\Delta t$ to $t = (n+1)\Delta t$ consists of $2M$ substeps with distinct effective time elements $\Delta t/z_s^{(2M)}$. It is not so much that the growth factor depends on time, as that it depends on the effective sizes of time substeps. Not only are the effective time substeps of differing magnitudes, they may also be complex. Let us consider the growth factor when $\Psi^{n,2M}$ is calculated from $\Psi^{n,0} = \Psi^n$. The growth factor over one substep is

$$\xi_s^{(2M)} = \frac{\widehat{\Psi}^{n,s+1}}{\widehat{\Psi}^{n,s}} = \left(1 + \frac{a}{z_s^{(2M)}}\right), \tag{2.24}$$

where $\widehat{\Psi}^{n,s}$ is the Fourier amplitude of the wave function with wave number k. The growth factor of the product in the second term of Eq. (2.17) is

$$\xi^{(2M)} = \prod_{s=1}^{2M} \xi_s^{(2M)} = \prod_{s=1}^{2M} \left(1 + \frac{a}{z_s^{(2M)}}\right), \quad \xi^{(0)} = 1. \tag{2.25}$$

Note that

$$\begin{aligned}\lim_{M\to\infty} \xi^{(2M)} &= \lim_{M\to\infty} \prod_{s=1}^{2M} \left(1 + \frac{a}{z_s^{(2M)}}\right) \\ &= \frac{\sin a}{a} < 1 \quad \text{for} \quad a > 0,\end{aligned} \tag{2.26}$$

so that for sufficiently large M, the intermediate growth factor is less than unity irrespective of the value of a. Given that the set of zeros for a particular M includes their complex conjugates, the magnitude of $\xi^{(2M)}$ depends on a^2. Assuming a growth factor of ξ^n in Ψ^n, we obtain an equation for the growth factor,

$$\xi^2 + 2ia\xi^{(2M)}\xi - 1 = 0, \tag{2.27}$$

from which the solutions

$$\xi = -ia\xi^{(2M)} \pm \sqrt{1 - a^2[\xi^{(2M)}]^2} \tag{2.28}$$

determine the condition for stability to be $(a|\xi^{(2M)}|)^2 \leqslant 1$. When that condition is satisfied, $|\xi| = 1$ and stability is assured.

C. Unitarity

Since the scheme of this paper is not strictly unitary, we explore the degree to which the normalization is conserved [3,6,14]. Consider Eq. (2.14) with

$$\sin(H\Delta t/\hbar) = S(H\Delta t/\hbar) + O[(H\Delta t/\hbar)^{2M+3}], \tag{2.29}$$

where $S(H\Delta t/\hbar)$ represents the polynomial part of the sine function; see Eq. (2.15). We express the algorithm as

$$\Psi^{n+1} = \Psi^{n-1} - 2iS(H\Delta t/\hbar)\Psi^n. \tag{2.30}$$

Since $\Psi^{n-1} = e^{iH\Delta t/\hbar}\Psi^n$ exactly,

$$\Psi^{n+1} = [e^{iH\Delta t/\hbar} - 2iS(H\Delta t/\hbar)]\Psi^n. \tag{2.31}$$

Hence,

$$\begin{aligned}\langle\Psi^{n+1}|\Psi^{n+1}\rangle = \langle\Psi^n|&[e^{-iH\Delta t/\hbar} + 2iS] \\ &\times [e^{iH\Delta t/\hbar} - 2iS]|\Psi^n\rangle,\end{aligned} \tag{2.32}$$

and from this equation, it follows that

$$\begin{aligned}\langle\Psi^{n+1}|\Psi^{n+1}\rangle &= \langle\Psi^n|1 - 4S\sin(H\Delta t/\hbar) + S^2|\Psi^n\rangle \\ &= \langle\Psi^n|1 + O[(H\Delta t/\hbar)^{2M+4}]|\Psi^n\rangle \\ &= \langle\Psi^n|\Psi^n\rangle + O[(\Delta t)^{2M+4}],\end{aligned} \tag{2.33}$$

where we have inserted the expression for S from Eq. (2.29) to obtain the result. Thus the normalization is secured to the order of $(\Delta t)^{2M+4}$, a result that is consistent with Ref. [6] when $M = 0$. The deviation is not cumulative when the scheme is stable.

D. Errors

Errors occur primarily because of the truncation of the series expansions and therefore depend on the orders of the method, i.e., r and M. Writing the wave function as a Taylor series about x and t, we obtain

$$\begin{aligned}\psi(x + \Delta x, t + \Delta t) = &\sum_{i=0}^{p-1} \frac{1}{i!}\left(\Delta x\frac{\partial}{\partial x} + \Delta t\frac{\partial}{\partial t}\right)^i \psi(x,t) \\ &+ \frac{1}{p!}\left(\Delta x\frac{\partial}{\partial x} + \Delta t\frac{\partial}{\partial t}\right)^p \psi(\bar{x},\bar{t}),\end{aligned} \tag{2.34}$$

where the point $(\bar{x},\bar{t})$ lies on the line segment that joins (x,t) to $(x + \Delta x, t + \Delta t)$ in the xt plane. If one of the increments is zero or much smaller than the other, this expression reduces to the one-dimensional Taylor series. Neglecting the term after the sum leads to a truncation error whose magnitude can be estimated using the dropped term.

We separate the truncation errors due to the integration over space and those due to integration over time. At a given time t, the spatial integration with the rth-order expansion yields a truncation error

$$e^{(r)} \approx C^{(r)}(\Delta x)^{2r}, \tag{2.35}$$

where $C^{(r)}$ is assumed to be slowly varying with r. Actually, $C^{(r)} = |\psi^{(2r)}(\bar{x},t)|/(2r!)$ for some $\bar{x}$ in the range of the spatial integration, and thus is model, or potential, dependent. If we specify an acceptable error, the step size Δx can be adjusted to obtain that error. Since $\Delta x = (x_0 - x_J)/J$, an adjustment of Δx is equivalent to a change in J. Given that $x_0 - x_J$ is fixed, we obtain

$$\Delta x = \frac{x_0 - x_J}{J} \approx \left(\frac{e^{(r)}}{C^{(r)}}\right)^{1/2r} \tag{2.36}$$

and

$$e^{(r)} \approx \frac{\text{const}}{J^{2r}}. \tag{2.37}$$

The CPU time of the calculation is proportional to the number of basic computer operations. For one time step, this involves multiplying a $(2r + 1)$-banded diagonal matrix and a J-dimensional vector $(2M + 1)$ times in Eq. (2.17). For the time step Δt with a particular M, the number of operations is therefore proportional to rJ. Hence,

$$\text{CPU time} \propto \text{no. operations} \propto Jr \propto \frac{r}{(e^{(r)})^{1/2r}}. \tag{2.38}$$

This form gives a minimum (optimum) CPU time occurring when

$$r \approx -\frac{\ln e^{(r)}}{2}. \tag{2.39}$$

For the time integration, we assume a truncation error independent of r. For a given r and Δx, the error due to finite Δt according to Eq. (2.15) is

$$e^{(2M)} \approx C^{(2M)}(\Delta t)^{2M+3}, \tag{2.40}$$

where again $C^{(2M)}$ is assumed to be a slowly varying function of M depending on the Hamiltonian. In the expansion of $\sin(H\Delta t)$, we have factors involving $\Delta t/z_s^{(2M)}$ rather than the

Δt in the elementary algorithm. For the purpose of estimating the error for a particular M, we replace Δt with $\Delta t/z_{\text{avg}}^{(2M)}$. The relationship of $z_{\text{avg}}^{(2M)}$ with M can be seen in Fig. 1, where $z_{\text{avg}}^{(2M)} \approx M$. Thus we can replace Eq. (2.40) by

$$e^{(2M)} \approx C^{(2M)}(\Delta t/M)^{2M+3}, \tag{2.41}$$

where the constant $C^{(2M)}$ is appropriately adjusted. If we take the total time $t_{\max} = N\Delta t$ to be fixed, then

$$\text{CPU time} \propto \left(e^{(2M)}\right)^{-\dfrac{1}{2M+3}}. \tag{2.42}$$

This CPU time is a decreasing function as M increases.

In order to test the accuracy of the methods, we use models for which the exact solutions are known. The error e_2 is obtained from the discretized spatial wave function using the Euclidean norm,

$$(e_2)^2 = \int_{x_0}^{x_J} dx\, |\psi(x, t_{\text{final}}) - \psi_{\text{exact}}(x, t_{\text{final}})|^2, \tag{2.43}$$

where t_{final} is the final time. A further item of note is that although explicit methods tend not to preserve normalization as, for instance, the (generalized) Crank-Nicolson method does, we find in the sample calculations that when the numerical solution is precise, the normalization is also accurate. Unlike Crank-Nicolson, for the algorithm proposed in this paper, the accuracy of the normalization is a manifestation of the precision of the solution.

E. Two spatial dimensions

For two spatial dimensions, the Hamiltonian is

$$H = -\frac{\hbar^2}{2m}\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) + V(x, y). \tag{2.44}$$

We partition both x and y coordinates into J intervals as in one dimension so that the discretized wave function is

$$\psi_{ij}^n = \psi(x_i, y_j, t_n) \quad \text{for} \quad i, j = 0, 1, \dots, J, \tag{2.45}$$

with $\Delta x = x_{i+1} - x_i$, $\Delta y = y_{j+1} - y_j$ for $i, j = 0, 1, \dots, J-1$. Since Eq. (2.18) holds for any spatial dimension,

$$\begin{aligned}\psi_{ij}^{n,s+1} &= \left\{1 - \frac{\Delta t}{\hbar z_s^{(2M)}}\left[-\frac{\hbar^2}{2m}\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)\right.\right.\\ &\quad \left.\left. + V(x, y)\right]\right\}\psi_{ij}^{n,s}\\ &= \psi_{ij}^{n,s+1} - \frac{\Delta t}{\hbar z_s^{(2M)}}\left[-\frac{\hbar^2}{2m}\sum_{k=-r}^{r} c_k^{(r)}\left(\frac{\psi_{i+k,j}^{n,s}}{(\Delta x)^2}\right.\right.\\ &\quad \left.\left. + \frac{\psi_{i,j+k}^{n,s}}{(\Delta y)^2}\right) + V_{ij}\psi_{ij}^{n,s}\right],\end{aligned} \tag{2.46}$$

where $V_{ij} = V(x_i, y_j)$. For the final step, we set

$$\psi_{ij}^{n+1} = \psi_{ij}^{n-1} - \frac{i\Delta t H}{\hbar}\psi_{ij}^{n'}, \tag{2.47}$$

where $\psi_{ij}^{n'} = \psi_{ij}^{n,2M}$ from Eq. (2.46). Generalization to cases where the size and number of steps in the x and y variables are different and to three or more Cartesian dimensions is straightforward. Reformulation of the problem for different coordinate systems can be done; for example, see Ref. [15] in which the Hamiltonian is expressed in terms of polar coordinates and integration occurs over the radial and polar angle coordinates.

TABLE III. The choice of parameters in addition to $\omega = 0.2, A = 10, x \in [-40, 40]$, and $t_{\text{final}} = 110\pi$ that yields the results shown in Fig. 2.

M	Δt	J	Δx	M	Δt	J	Δx
0	$\pi/800$	400	0.2	4	$\pi/100$	400	0.2
1	$\pi/400$	400	0.2	6	$\pi/80$	400	0.2
2	$\pi/300$	400	0.2	8	$\pi/60$	400	0.2
3	$\pi/200$	400	0.2	10	$\pi/50$	400	0.2

III. NUMERICAL EXAMPLES

A. Oscillation of a coherent wave packet

The oscillations of a coherent state in the harmonic oscillator well are described in Ref. [16]. The reliability and precision of the numerical procedure are assessed by comparing the numerical results of this system with exact ones [1,4]. The time-dependent Schrödinger equation is

$$i\hbar\frac{\partial}{\partial t}\psi(x, t) = \left(-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + \frac{1}{2}Kx^2\right)\psi(x, t), \tag{3.1}$$

and the initial state is the displaced ground-state wave function,

$$\psi(x, 0) = \frac{\alpha^{1/2}}{\pi^{1/4}}e^{-\frac{1}{2}\alpha^2(x-A)^2}, \tag{3.2}$$

where $\alpha^4 = mK/\hbar^2$, $\omega = \sqrt{K/m}$, and A is the displacement from the origin of the ground-state wave function. The closed expression for the time-evolved wave function is

$$\begin{aligned}\psi_{\text{exact}} = \frac{\alpha^{1/2}}{\pi^{1/4}}\exp&\left[-\frac{1}{2}(\xi - \xi_0\cos\omega t)^2\right.\\ &\left.-i\left(\frac{1}{2}\omega t + \xi\xi_0\sin\omega t - \frac{1}{4}\xi_0^2\sin 2\omega t\right)\right],\end{aligned} \tag{3.3}$$

where $\xi = \alpha x$ and $\xi_0 = \alpha A$.

We set $\hbar = m = 1$, $\omega = 0.2$, and $A = 10$ and choose the computational space such that $x \in [x_0, x_J] = [-40, 40]$. The period of oscillation is then $T = 10\pi$. We allow the coherent state to oscillate for 11 periods before comparing the numerical solution to the exact one by calculating the error e_2 using Eq. (2.43).

In Fig. 2, we plot e_2 as a function of r for a select set of M values. In all cases, $\Delta x = 0.2$. We choose Δt as shown in Table III.

Remarkably, when r ranges from 1 to 30 and M from 0 to 10, we note 30 orders of magnitude decrease in the error e_2. Even at the top end of the ranges when $M = 10$ and $r = 25, \dots, 30$, the error e_2 still shrinks five orders of magnitude. Worth mentioning is the fact that the conditions that are employed in many elementary calculations, such as

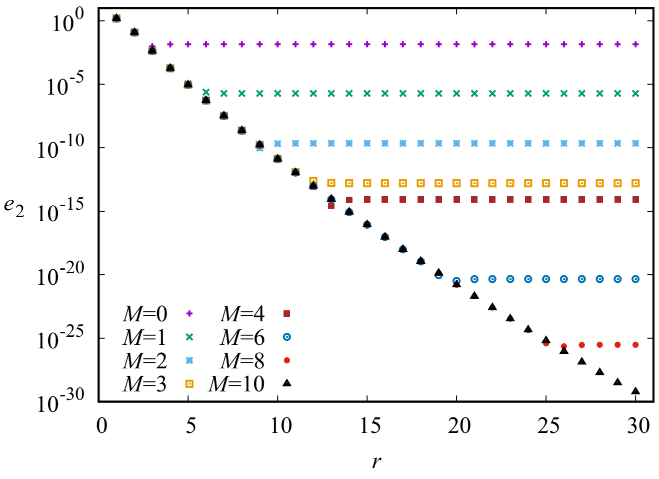

FIG. 2. The error as a function of r for a select set of values of M given in Table III. The parameters are $\hbar = m = 1, \omega = 0.2, A = 10$. The error is determined when $t = 110\pi$. In all cases, $\Delta x = 0.2$.

in Ref. [6], are represented by the single point in the upper left-hand corner of the graph when $M = 0$ and $r = 1$.

In order to test the efficacy, we aim for an error approximately equal to, but less than, 10^{-8} and determine the CPU time[1] for various values of r and M. In Table IV, we list the CPU times when M is fixed at 10, $\Delta t = \pi/55$, and the number of space steps J is chosen to result in an error close to the preset limit. Note that a minimum CPU time occurs when r is approximately 11.

The efficiency as a function of M is a little less clear because of the rapid increase in efficiency when M is relatively small. Setting $r = 10$, $\Delta x = 0.4$ and an error around 6×10^{-6}, we adjust the time-step size to obtain the target error along with the resulting CPU time. The results are tabulated in Table V. We note that the CPU time reduces drastically for $M = 1$ compared with the traditional $M = 0$ calculation.

For values of M greater than two, the error is robust. By this is meant that the error remains the same for smaller Δt; the values listed correspond to the largest Δt that results in a stable solution. The CPU time is an estimate and goes down to approximately 1 s compared with 1000 s for $M = 0$. Instability of the algorithm is evident when, during the time stepping, e_2 becomes large and the calculated wave function is no longer normalized to unity.

To evaluate the method with the current state of the art, in Table VI we compare CPU time to the Chebyshev method which is claimed "to outperform all other methods for problems with time-independent Hamiltonians" [17], and two other methods by van Dijk and Toyama (VT) [1] and by Shao and Wang (SW) [18]. The latter two are generalized Crank-Nicolson with central differencing and high-order-compact differencing, respectively, of arbitrary order.

TABLE IV. Computation time (CPU) as a function of r when the number of time steps is adjusted to obtain an error close to, but smaller than, the preset limit of 10^{-8}. Parameters are $\omega = 0.2, A = 10, t_{\text{final}} = 110\pi$.

M	r	J	Δx	Δt	e_2	CPU (s)
10	30	152	0.52632	$\pi/55$	8.695×10^{-9}	7.095
10	25	161	0.49689	$\pi/55$	9.387×10^{-9}	6.477
10	20	176	0.45454	$\pi/55$	9.903×10^{-9}	5.905
10	15	206	0.38835	$\pi/55$	9.231×10^{-9}	5.445
10	13	227	0.35242	$\pi/55$	9.351×10^{-9}	5.241
10	11	260	0.30769	$\pi/55$	9.287×10^{-9}	5.240
10	10	284	0.28169	$\pi/55$	9.482×10^{-9}	5.303
10	9	317	0.25237	$\pi/55$	9.576×10^{-9}	5.624
10	8	364	0.21978	$\pi/55$	9.841×10^{-9}	5.734

Clearly, the latter two methods become much less efficient as the spatial dimension increases. From the entries in the table, it is evident that the Chebyshev method, which uses fast-Fourier transforms to calculate the second-order spatial derivative of the wave function and in which the wave function is determined only at the end of the total time interval, is remarkably efficient. However, if one needs the wave function at various times to investigate the time behavior of observables, the Chebyshev method requires more time, although the precision does not change. The Chebyshev method with 440 equal time intervals requires a CPU time of 0.488 s, and with 4180 intervals, a CPU time of 1.650 s. Although the relative magnitudes of these CPU times are not precise, they are typical, approximate indications. A comparison of the methods to determine time-dependent wave functions is model and application dependent. With that caveat, we mention the recent comparison of the CPU times incurred when computing the time-dependent quantum states of bilayer graphene in a magnetic field using either the Chebyshev method or an improved split-operator method [19]. Qualitatively, the efficacy of the two methods is comparable, although if expectation values at small time intervals are required, the Chebyshev method loses efficiency and the split-operator method can be preferable.

Split-operator methods for linear Schrödinger equations with unbounded potentials, such as the one used in this section, have been investigated in Ref. [20]. With two-dimensional systems, they compare the error as a function of temporal step size using split-operator approximations of the

TABLE V. Computation time as functions of M and the corresponding Δt. Parameters are $\omega = 0.2, A = 10, x \in [-40, 40], t_{\text{final}} = 110\pi$.

M	Δt	J	r	e_2	CPU (s)
0	$\pi/100,000$	200	10	5.773×10^{-6}	499.7
0	$\pi/200,000$	200	10	6.054×10^{-6}	992.8
1	$\pi/1,000$	200	10	6.197×10^{-6}	9.714
2	$\pi/150$	200	10	6.159×10^{-6}	2.504
4	$\pi/40$	200	10	6.161×10^{-6}	0.904
10	$\pi/22$	200	10	6.161×10^{-6}	1.513
15	$\pi/16$	200	10	6.161×10^{-6}	1.577

[1] The CPU time is dependent on the computer used and is relevant for comparisons only if the calculations are done on the same computer. The CPU model used for the calculations of this paper is the Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz.

TABLE VI. Comparison with other methods.

	Sine (expans.)	Chebyshev	VT	SW
r, M	16, 4		16,5	6,5
J	228	127	255	255
$e_2 \times 10^{10}$	0.831	0.223	0.174	0.116
CPU(s)	1.650	0.193	2.298	0.565
N	4180	1	440	440

order of 2, 4, and 6. Higher-order split-level operator methods have significant complexity [21]. Table III displays a similar relationship, i.e., higher order accommodates larger step size, but with no corresponding increased complexity.

B. Wave-packet scattering

For further comparison of a known exact solution with the numerical one, we consider the analytic solution of the free potential [$V(x) = 0$], i.e., the Hermite-Gaussian traveling wave function [22,23], and allow it to scatter from a finite potential. The free wave function is

$$\Psi_n(x,t) = \frac{N_n(\alpha)e^{in\theta}}{\sqrt{1+i\alpha^2\gamma}} H_n(\xi)\exp(-\xi^2/2) \times \exp\left\{i\left[\frac{\alpha^4\gamma(x-A)^2 + 2k_0(x-A) - k_0^2\gamma}{2(1+\alpha^4\gamma^2)}\right]\right\}, \tag{3.4}$$

where

$$\gamma = \hbar(t-t_0)/m, \quad \theta = -\arctan(\alpha^2\gamma),$$
$$\xi = \frac{\alpha[(x-A)-k_0\gamma]}{\sqrt{1+\alpha^4\gamma^2}}, \quad N_n(\alpha) = \left(\frac{\alpha}{\sqrt{\pi}2^n n!}\right)^{1/2}. \tag{3.5}$$

The integer $n = 0, 1, \ldots$ labels a wave function that at $t = t_0$, is the nth energy state of the harmonic oscillator centered at $x = A$ with an oscillator constant $K = \hbar^2\alpha^4/m$. The initial time $t_{\rm init}$ can have any positive or negative value; t increases starting from $t_{\rm init}$. The quantity k_0 is the wave number since the expectation value of the momentum is $\langle p\rangle_n = \hbar k_0$.

As an example of the propagation of the Hermite-Gaussian wave function, we take parameters $k_0 = 3, \alpha = 0.3, A = 0$, and $t_0 = 15$. Initially, at $t = 0$, the center of the wave packet is at $x = -45$ (assuming $\hbar = m = 1$), and at $t = 15$ it has moved to $x = 0$, and subsequently at $t = 30$ it is at $x = 45$, as shown in Fig. 3. For the numerical calculation $x \in [-200, 200]$, $\Delta x = 0.25$, and $\Delta t = 0.05$.

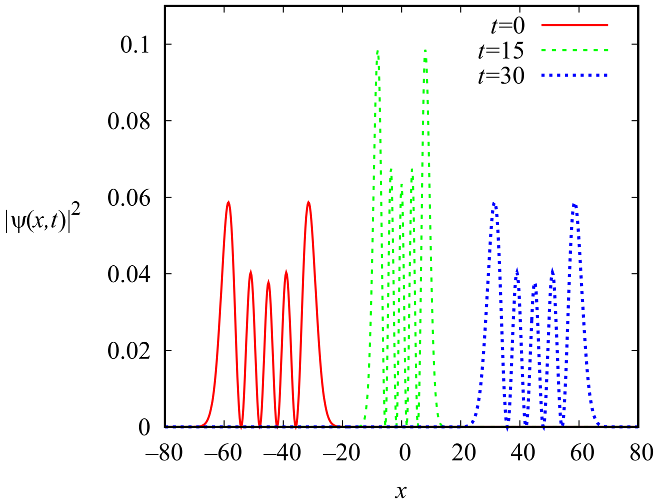

FIG. 3. The numerical results for the square of the modulus of the free Hermite-Gaussian wave function at the times indicated. Note that $k_0 = 3, \alpha = 0.3, A = 0, x \in [-200, 200], \Delta x = 0.25$, and $\Delta t = 0.05$.

For this calculation, we take $M = 15$, $r = 30$, resulting in an error of the final wave function $e_2 = 2.3 \times 10^{-6}$. Note that the minimum spatial uncertainty of the wave packet (width) occurs at $x = 0$, but this point could be chosen elsewhere by adjusting t_0.

To explore the scattering of the Hermite-Gaussian wave function, we consider the Pöschl-Teller potential [24,25],

$$V(x) = V_1 \text{sech}^2(\beta x), \tag{3.6}$$

which yields transmission and reflection coefficients from a time-independent analysis,

$$T_{\rm stat}^{\rm (PT)}(k) = \frac{q^2}{1+q^2}, \quad R_{\rm stat}^{\rm (PT)}(k) = \frac{1}{1+q^2}, \tag{3.7}$$

where

$$q = \sinh(\pi k/\beta)\text{sech}\left(\frac{\pi}{2}\sqrt{\frac{4V_1}{\beta^2} - 1}\right). \tag{3.8}$$

Numerically, the scattering coefficients are determined using the asymptotic time-dependent wave functions after they have exited the scattering, or potential, region. Hence the transmission and reflection coefficients are

$$T(k) = \int_b^\infty |\psi(x,t)|^2\, dx, \quad R(k) = \int_{-\infty}^{-b} |\psi(x,t)|^2\, dx, \tag{3.9}$$

where $b > 0$ is larger than the range of the potential centered at the origin. The wave number k is taken to be the expectation value k_0 of the Hermite-Gaussian wave function.

Figure 4 displays the analytically and numerically calculated transmission and reflection coefficients. The parameters that determine the graphs are $V_1 = 13, \beta = 1$, $\alpha = 0.05, A = 0, t_0 = 150/k, x \in [-300, 300], \Delta x = 0.375$, $\Delta t = 0.05, t_{\rm init} = 0$, and $t_{\rm final} = 300/k$.

For the numerical computation, $M = 15$ and $r = 30$. There is good agreement between the time-independent scattering coefficients and the numerically determined ones. The progression of the wave packet in time is shown in Fig. 5 for the case when $k_0 = 3.6$.

The nodal character of the transmitted and reflected packets is similar to that of the incident packet. The relation of the shape of the scattered packet compared to that of the incident packet is discussed in Ref. [23].

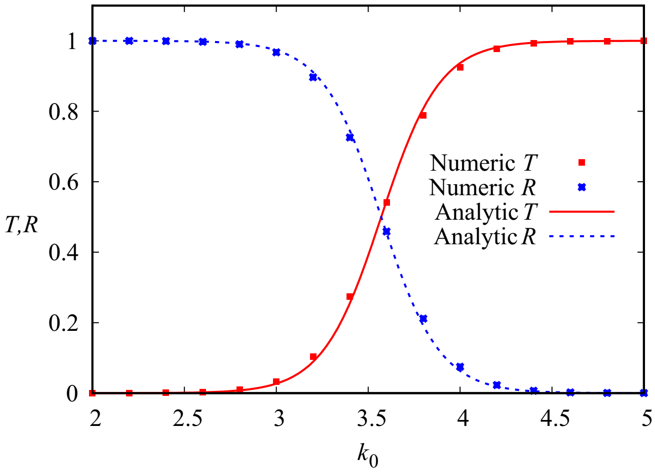

FIG. 4. The transmission and reflection coefficients calculated numerically using Eq. (3.9) and analytically using Eq. (3.7) with $V_1 = 13$, $\beta = 1$, $\alpha = 0.05$, $A = 0$, $t_0 = 150/k$, $x \in [-300, 300]$, $\Delta x = 0.375$, $\Delta t = 0.05$, $t_{\text{init}} = 0$, $k = k_0$, and $t_{\text{final}} = 300/k$.

C. Two-dimensional model

The potential of a simple solvable two-dimensional model that has been employed to test numerical methods [12,26,27] is

$$V(x, y) = -(3 - 2\tanh^2 x - 2\tanh^2 y), \qquad (3.10)$$

which has an exact solution,

$$\psi(x, y, t) = \tfrac{1}{2} i e^{it}\,\text{sech}(x)\text{sech}(y). \qquad (3.11)$$

Here we assume $\hbar = 2m = 1$. With a domain of spatial integration of $[-40, 40]^2$ and of the time integration of [0,1], we take the time-step size for different values of M as shown in Table VII.

In all cases, $J = 400$ and $\Delta x = \Delta y = 0.2$, but for $M > 1$, larger Δt gives convergent results. As in the one-dimensional applications, the instability of the integration is indicated by a drastic concurrent increase of the error and the normalization while stepping through the time intervals. Thus one can determine the stability of the numerical solution by monitoring the normalization. It is extremely unlikely that a runaway error occurs while the normalization remains unity.

TABLE VII. The number of spatial grid points (J) and the total number of time intervals (N) for different values of M in Fig. 6.

M	J	N	Δt	M	J	N	Δt
0	400	500	$0.003\dot{3}$	10	400	50	0.0200
1	400	150	$0.006\dot{6}$	15	400	50	0.0200
5	400	100	0.0100				

The patterns of the dots representing errors in Fig. 6 show a consistent behavior of a sharp decline for smaller values of r, followed by a linear portion with negative slope, which in turn is followed by a slow nonlinear decline. With the exception of the $M = 0$ case, the linear portion starts at smaller values of r for larger values of M. By using smaller time increments, we achieve a universal curve which asymptotically matches the curves of the dots shown (see $M = 10$ and 15 cases). The salient conclusion is that significant improvement of accuracy is possible when $M > 0$. The rogue point on the graph corresponding to $M = 1$ and $r = 28$ is an instance of the onset of instability: both e_2 and the normalization start to increase at some point in the course of time stepping and do so at an even larger rate for $r = 29$ and 30.

The figure also displays the CPU time as a function of r. For the cases shown when $M = 1$, one obtains the most efficient calculation, although slightly greater accuracy may be had with higher values of M and CPU time. Note that the CPU times for $M = 5$ and 10 with different Δt are nearly the same, but the $M = 10$ case gives greater accuracy.

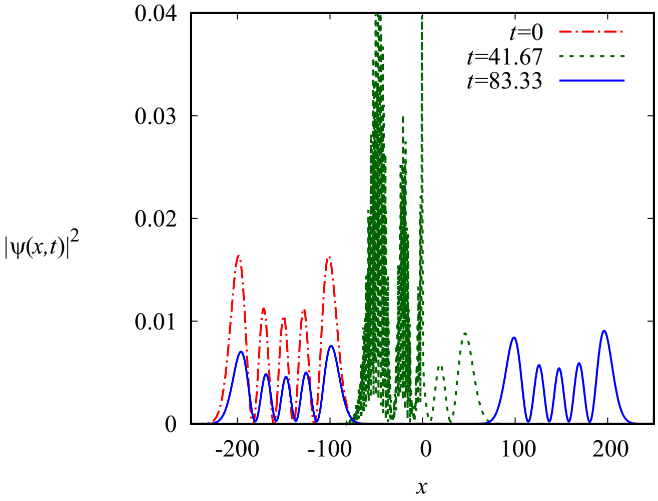

FIG. 5. The wave packet before, during, and after scattering from the Pöschl-teller potential. The parameters are those of Fig. 4 with $k_0 = 3.6$.

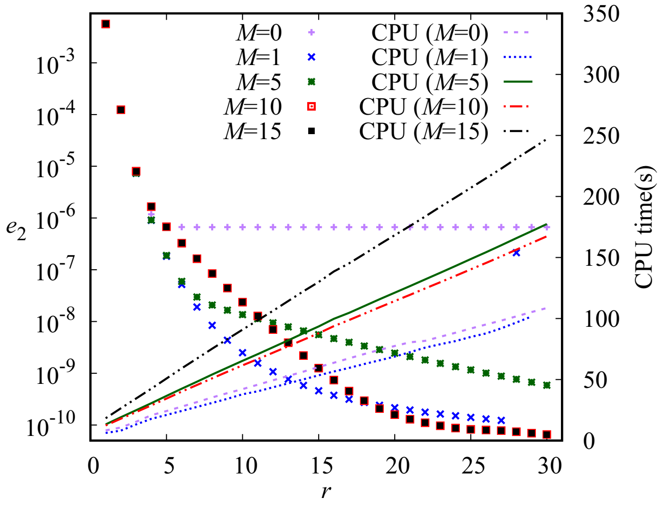

FIG. 6. The error and CPU time(s) as functions of r for different values of M for the potential of Eq. (3.10). The space increments are $\Delta x = \Delta y = 0.2$ and the time increment is as shown in Table VII.

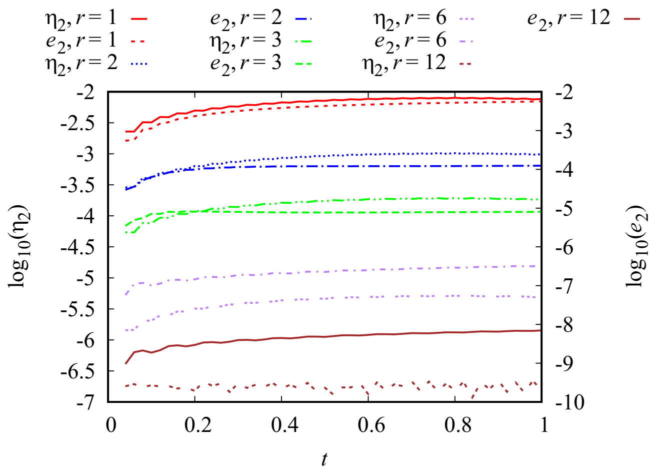

FIG. 7. The deviation from normalization and the error as a function of time for $M = 10$ and various values of r for the potential of Eq. (3.10). The space increments are $\Delta x = \Delta y = 0.2$ and the time increment is as shown in Table VII.

As a measure of the deviation from the exact normalization, we compute the quantity

$$\eta_2 = \sqrt{\left| 1 - \int_{-b}^{b} \int_{-b}^{b} dxdy\, |\psi(x,t)|^2 \right|} \tag{3.12}$$

with $b = 40$. In Fig. 7, η_2 and e_2 are plotted as functions of time for $M = 10$ and indicated values of r from 1 to 12. For the smaller r, the orders of magnitude of the error and the deviation from the norm are nearly the same, whereas for larger r, the error is smaller by one order of magnitude. In the cases shown, the normalization is preserved to within one order of magnitude of the error. Since the normalization is not rigorously conserved, it can be monitored as a check for runaway error when no exact solution is available [3].

IV. CONCLUSIONS AND COMMENTS

We have presented a method of numerically solving the time-dependent Schrödinger equation that is simple, explicit, stable, and flexible in terms of obtaining pertinent accuracy. The method allows calculations of higher order in time without additional and complicated analysis. In spite of the fact that the usual limitations of time-stepping methods in terms of the size of the time and spatial increments apply, the efficacy is comparable to those methods and may exceed it in multidimensional systems.

Advantages of explicit over implicit methods have been discussed in the literature; see, for example, Ref. [6]. Implicit methods require the inversion of matrices of dimension equal to the number of spatial grid points, whereas the explicit approach involves the product of a banded-diagonal matrix and a column vector. This is especially beneficial when dealing with higher spatial dimensions. An implicit method, such as Crank-Nicolson, is inherently unitary and normalization is preserved in spite of the error. Thus the magnitude of the wave function is correct, regardless of the (im)precision of the phase. We find, however, that although the normalization is not guaranteed in the explicit method, when it deviates it signals an unacceptable error.

We have shown that high-precision stable numerical solutions are possible with explicit methods by combining the stable algorithm for explicit solutions [6] with a factorized polynomial expansion of the propagator. Given the central differences approach to the spatial integration that we employ, we can characterize the precision of the solutions in terms of errors of $O[(\Delta x)^{2r}]$ and $O[(\Delta t)^{2M+3}]$ for arbitrary r and M. The stability of the algorithm and the precision of the solution have been tested with simple examples and indicate that the monitoring of the normalization is a way of ensuring accurate solutions. The proposed method is another among many others for consideration in practical applications. An interesting higher-order exponential split-operator method [28] also determines solutions to $O[(\Delta t)^{2M+1}]$. However, for $M \geqslant 3$, the calculations become prohibitive due to the number of exponential function evaluations.

It is evident from this work that calculations with large r and M are practical because of the simplicity of the algebraic operations; furthermore, higher values make a significant difference for the accuracy of the solutions. The algorithm requires two input states, say at $t = 0$ and $t = \Delta t$. For the examples of this paper, the exact solutions give the input. When there is no exact solution, the second time-stepped state may be obtained using an approach analogous to the one of this paper by factoring the exponential propagator to derive a two-level process. For the lowest-order term ($M = 0$), this algorithm is unstable, but it can be shown to be stable for higher M. We anticipate reporting on this work presently.

Alternative to the central-difference approach of the spatial integration is the Fourier spectral technique which employs fast-Fourier transforms. However, the time (or number of arithmetical operations) of this approach is proportional to $J \ln J$ as compared to rJ for central differencing. Gaspar *et al.* [29] have compared the two approaches for two-dimensional systems using the alternating-direction-implicit central-difference and the fast-Fourier-transform implementation for a specific example and have found that the former is about three times faster.

An alternative to Eq. (2.14) as an expression of the time propagation is

$$\psi(x, t + \Delta t) = -\psi(x, t - \Delta t) + 2\cos(H\Delta t/\hbar)\psi(x, t), \tag{4.1}$$

which is the starting point of Gray and Balint-Kurti [30] to generate the time evolution of the real part of a wave packet. This is useful to generate S-matrix elements which can be determined from the real part of the wave function. This cannot be done with an expansion of the cosine, as with the sine, since the expansion involves complex roots. Thus, for the purpose of obtaining S-matrix elements, the work of this paper calculates the imaginary part of the wave function superfluously and hence is less efficient and demands double the memory allocation. However, often the time dependence of the complex wave function is required to obtain time-dependent observable quantities.

In this analysis, we have assumed wave functions that vanish at the spatial boundary. There are approaches of including nonzero boundary conditions for higher-order methods, but they tend to depend on the order. For instance, the method of summation by parts and the simultaneous-approximation term [31,32] give a matrix representation of the second-order derivative which is modified for the first few rows and columns to account for the spatial boundary condition. It is not clear how to systematically implement this approach to arbitrary order.

Furthermore, the boundary effect when the wave packet exits the computational space is dealt with using absorbing or transparent boundary conditions. In its simplest form, the complex absorbing potential is employed [33] as a negative imaginary potential at the asymptotic regions that is added in the Hamiltonian so that the wave function is absorbed and vanishes (approximately) at the computational boundary. Alternatively, the transparent boundary condition allows the wave function to exit the computational space without reflection. This approach is discussed by Moyer [34] and a frequently used method was introduced by Basakov and Popov [35] and later applied to two- and three-dimensional Cartesian systems [36,37]. Since both approaches depend on the values of the wave function near or at the grid boundary, their applicability and limitations can be tested using the wave functions obtained with the explicit method discussed in this paper [38].

[1] W. van Dijk and F. M. Toyama, Accurate numerical solutions of the time-dependent Schrödinger equation, Phys. Rev. E **75**, 036707 (2007).

[2] H. Tal-Ezer and R. Kosloff, An accurate and efficient scheme for propagating the time dependent Schrödinger equation, J. Chem. Phys. **81**, 3967 (1984).

[3] C. Leforestier, R. H. Bisseling, C. Cerjan, M. D. Feit, R. Friesner, A. Guldberg, A. Hammerich, G. Jolicard, W. Karrlein, H.-D. Meyer, N. Lipkin, O. Roncero, and R. Kosloff, A comparison of different propagation schemes for the time dependent Schrödinger equation, J. Comput. Phys. **94**, 59 (1991).

[4] W. van Dijk, J. Brown, and K. Spyksma, Efficiency and accuracy of numerical solutions to the time-dependent Schrödinger equation, Phys. Rev. E **84**, 056703 (2011).

[5] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran 77: The Art of Scientific Computing*, 2nd ed. (Cambridge University Press, Cambridge, 1992).

[6] A. Askar and A. S. Cakmak, Explicit integration method for the time-dependent Schrödinger equation for collision problems, J. Chem. Phys. **68**, 2794 (1978).

[7] R. J. Rubin, Comment on the explicit integration method for the time-dependent Schrödinger equation, J. Chem. Phys. **70**, 4811 (1979).

[8] H. J. Harmuth, On the solution of the Schrödinger and the Klein-Gordon equations by digital computers, J. Math. Phys. **36**, 269 (1957).

[9] Z. Wang and H. Shao, A new kind of discretization scheme for solving a two-dimensional time-independent Schrödinger equation, Comput. Phys. Commun. **180**, 842 (2009).

[10] D. R. Lynch, *Numerical Partial Differential Equations for Environmental Scientists and Engineers: A First Practical Course I* (Springer Science + Business Media, New York, 2005).

[11] M. A. Ajaib, Numerical methods and causality in physics, arXiv:1302.5601v1 (2013).

[12] Z. F. Tian and P. X. Yu, High-order compact ADI (HOC-ADI) method for solving unsteady 2d Schrödinger equation, Comput. Phys. Commun. **181**, 861 (2010).

[13] K. Kumari and D. A. Donzis, A generalized von Neumann analysis for multi-level schemes: Stability and spectral accuracy, J. Comput. Phys. **424**, 109868 (2021).

[14] D. Kosloff and R. Kosloff, A Fourier method solution for the time dependent Schrödinger equation as a tool in molecular dynamics, J. Comput. Phys. **52**, 35 (1983).

[15] M. Schulte, Numerical solution of the Schrödinger equation on unbounded domains, Ph.D. thesis, Westfälischen Wilhelms Universität, Münster, 2007.

[16] L. I. Schiff, *Quantum Mechanics*, 3rd ed., International Series in Pure and Applied Physics (McGraw-Hill, New York, 1968).

[17] I. Schaefer, H. Tal-Ezer, and R. Kosloff, Semi-global approach for propagation of the time-dependent Schrödinger equation for time-dependent and nonlinear problems, J. Comput. Phys. **343**, 368 (2017).

[18] H. Shao and Z. Wang, Numerical solutions of the time-dependent Schrödinger equation: Reduction of the error due to space discretization, Phys. Rev. E **79**, 056705 (2009).

[19] P. A. Maksym and H. Aoki, Fast split operator method for computation of time dependent quantum states of bilayer graphene in a magnetic field, Physica **112**, 66 (2019).

[20] C. Neuhauser and M. Thalhammer, On the convergence of splitting methods for linear evolutionary Schrödinger equations involving an unbounded potential, BIT Numer. Math. **49**, 199 (2009).

[21] S. Blanes and P. Moan, Fourth- and sixth-order commutator-free Magnus integrators for linear and non-linear dynamical systems, Appl. Num. Math. **56**, 1519 (2006).

[22] W. van Dijk, F. M. Toyama, S. J. Prins, and K. Spyksma, Analytic time-dependent solutions of the one-dimensional Schrödinger equation, Am. J. Phys. **82**, 955 (2014).

[23] W. van Dijk, D. W. L. Sprung, and Y. Castonguay-Page, Tunnelling of Hermite-Gaussian wave packets, Phys. Scr. **95**, 065223 (2020).

[24] L. D. Landau and E. M. Lifshitz, *Quantum Mechanics: Nonrelativistic Theory* (Pergamon Press, London, 1958).

[25] S. Flügge, *Practical Quantum Mechanics* (Springer-Verlag, Berlin, 1974).

[26] Y. Xu and L. Zhang, Alternating direction implicit method for solving two-dimensional cubic nonlinear Schrödinger equation, Comput. Phys. Commun. **183**, 1082 (2012).

[27] R. Čiegis, A. MirinaviČius, and M. Radziunas, Comparison of split step solvers for multidimensional Schrödinger problems, Comp. Methods Appl. Math. **13**, 237 (2013).

[28] A. D. Bandrauk and H. Shen, Exponential split operator methods for solving coupled time-dependent Schrödinger equations, J. Chem. Phys. **99**, 1185 (1993).

[29] F. J. Gaspar, C. Rodrigo, R. Čiegis, and A. Mirinavičius, Comparison of solvers for 2d Schrödinger problems, Intl. J. Numer. Anal. Model. **11**, 131 (2014).

[30] S. K. Gray and G. G. Balint-Kurti, Quantum dynamics with real wave packets, including application to three-dimensional ($j = 0$) $D + H_2 \rightarrow HD + H$ reactive scattering, J. Chem. Phys. **108**, 950 (1998).

[31] K. Mattsson and J. Nordström, Summation by parts operators for finite difference approximations of second derivatives, J. Comput. Phys. **199**, 503 (2004).

[32] A. Nissen, G. Kreiss, and M. Gerritsen, High order stable finite difference methods for the Schrödinger equation, J. Sci. Comput. **55**, 173 (2013).

[33] J. G. Muga, J. P. Palao, B. Navarro, and I. L. Egusquiza, Complex absorbing potentials, Phys. Rep. **395**, 357 (2004).

[34] C. A. Moyer, Numerov extension of transparent boundary conditions for the Schrödinger equation in one dimension, Am. J. Phys. **72**, 351 (2004).

[35] V. Baskakov and A. Popov, Implementation of transparent boundaries for numerical solution of the Schrödinger equation, Wave Motion **14**, 123 (1991).

[36] R. M. Feshchenko and A. V. Popov, Exact transparent boundary condition for the parabolic equation in a rectangular computational domain, J. Opt. Soc. Am. A **28**, 373 (2011).

[37] R. M. Feshchenko and A. V. Popov, Exact transparent boundary condition for the three-dimensional Schrödinger equation in a rectangular cuboid computational domain, Phys. Rev. E **88**, 053308 (2013).

[38] A. Nissan, Absorbing boundary techniques for the time-dependent Schrödinger equation, Master's thesis, Uppsala University, 2010.