Computers in Physics

A fast explicit algorithm for the timedependent Schrödinger equation

P. B. Visscher

Citation: Computers in Physics **5**, 596 (1991); doi: 10.1063/1.168415
View online: http://dx.doi.org/10.1063/1.168415
View Table of Contents: http://scitation.aip.org/content/aip/journal/cip/5/6?ver=pdfcov
Published by the AIP Publishing

A fast explicit algorithm for the time-dependent Schrödinger equation

P. B. Visscher
Department of Physics and Astronomy, University of Alabama, Tuscaloosa, Alabama 35487-0324

(Received 9 November 1990; accepted 1 May 1991)

An explicit algorithm for the time-stepping solution of the Schrödinger equation is described, which is second-order accurate in time. It is a staggered-time algorithm, in which the real and imaginary parts of the wave function are defined at alternate times. The method combines the speed and simplicity of explicit methods with the accuracy and stability of second-order implicit methods. Because of this simplicity and speed, the algorithm is well suited for pedagogical applications on personal computers, as well as for computation-intensive research applications.

INTRODUCTION

Methods for solving partial differential equations fall into two general categories: explicit [such as the forward-time centered-space algorithm given by Eq. (4) below] and implicit (either fully implicit or Crank–Nicholson) methods. Explicit methods are much simpler, but, in general, have the disadvantage that the error is of order Δt, so that one needs an unreasonably small Δt to achieve enough accuracy even for qualitative pedagogical applications. In the case of the Schrödinger equation the situation is even worse—this explicit method is always unstable.[1] Implicit methods can be made second-order accurate in Δt (Crank–Nicholson methods) and are stable. However, implicit methods are slow and complicated because a system of linear equations must be solved at each time step. The presently proposed staggered-time algorithm has the advantages of both with the disadvantages of neither.

I. BACKGROUND

The Schrödinger equation is

$$i\frac{d\psi}{dt} = H\psi \tag{1}$$

in units in which $\hbar = 1$, where the Hamiltonian is

$$H = -\nabla^2/2m + V. \tag{2}$$

In all of the algorithms we will discuss here, we discretize the Schrödinger equation on a one-dimensional grid of N points separated by Δr, and approximate the Laplacian ∇^2 by the three-point ("centered-space"[2]) formula

$$\nabla^2\psi(r,t) = [\psi(r+\Delta r,t) - 2\psi(r,t) + \psi(r-\Delta r,t)]/\Delta r^2. \tag{3}$$

The approach is easily generalized to two or three dimensions. Either periodic or hard-wall ($\psi = 0$) boundary conditions can be easily implemented.

In the forward-time centered-space algorithm,[2] the time derivative is approximated by a forward difference, giving

$$\psi(t+\Delta t) = \psi(t) + iH\Delta t\psi(t), \tag{4}$$

where we have suppressed the position index r, i.e., $\psi(t)$ is a vector of N complex components, and H is a complex $N\times N$ matrix. As can be seen by Taylor expanding $\psi(t+\Delta t)$, this has an error at each time step of order Δt^2, so that after $T/\Delta t$ steps the error is of order $T\Delta t$. In addition, this algorithm is unstable for any choice of Δr and Δt.[1]

The Crank–Nicholson method improves on this by thinking of $\psi(t+\Delta t) - \psi(t)$ as a centered difference, so that the ψ on the right-hand side of Eq. (1) is evaluated at $t + \frac{1}{2}\Delta t$ and approximated by an average:

$$\psi(t+\Delta t) - \psi(t) = +iH\Delta t\,\tfrac{1}{2}[\psi(t+\Delta t) + \psi(t)]. \tag{5}$$

By Taylor expanding about $t + \frac{1}{2}\Delta t$, it can be seen that this involves a smaller overall error of order $T\Delta t^2$. This algorithm also has the advantage of conserving probability; this follows from the fact that the discrete time-evolution operator is unitary.[1] However, it is an implicit algorithm [$\psi(t+\Delta t)$ appears on both sides of the equation] so that solution of an $N\times N$ system of linear equations is required at each time step. To do this straightforwardly requires $O(N^3)$ steps; there are specialized techniques for tridiagonal systems[1] that can reduce the number of steps to $O(N)$, but it is still at least a factor of 3 slower than an explicit method. This added complexity also makes the algorithm much less useful in a pedagogical context.

II. ALGORITHM

The present algorithm is motivated by writing the Schrödinger equation in terms of the real and imaginary parts R and I of the wave function:

$$\frac{dR}{dt}(t) = HI, \tag{6a}$$

$$\frac{dI}{dt}(t) = -HR. \tag{6b}$$

These equations have a structure similar to those for particle positions and velocities q and v in a trajectory calcula-

tion (the equation for dR/dt has only I on the right-hand side, just as dq/dt has only v). Thus it is natural to define R and I at staggered times, as is commonly done in trajectory calculations.[3] We will define R at times $0,\Delta t,2\Delta t,\ldots$, and I at $0.5\Delta t,1.5\Delta t,\ldots$. The natural discretization of Eqs. (6) is therefore

$$R(t+\tfrac{1}{2}\Delta t) = R(t-\tfrac{1}{2}\Delta t) + \Delta t H I(t), \tag{7a}$$

$$I(t+\tfrac{1}{2}\Delta t) = I(t-\tfrac{1}{2}\Delta t) - \Delta t H R(t). \tag{7b}$$

Given "initial" values $R(x,0)$ and $I(x,\frac{1}{2}\Delta t)$, Eqs. (7) explicitly determine the evolution of the system. Like the implicit unitary algorithm, this staggered-time algorithm can easily be seen to be second-order accurate in Δt.

The proper definition of probability density ($P = R^2 + I^2$, in the continuum limit) is not obvious in this system, since R and I are not defined at the same time. The choice

$$P(x,t) = R(x,t)^2 + I(x,t+\tfrac{1}{2}\Delta t)I(x,t-\tfrac{1}{2}\Delta t) \tag{8a}$$

at integer $t/\Delta t$ and

$$P(x,t) = R(x,t+\tfrac{1}{2}\Delta t)R(x,t-\tfrac{1}{2}\Delta t) + I(x,t)^2 \tag{8b}$$

at half-integer $t/\Delta t$ can easily be shown to conserve probability [the discrete time derivative $\Sigma_x P(x,t+\frac{1}{2}\Delta t) - \Sigma_x P(x,t)$ vanishes identically] and is therefore the most reasonable choice.

The stability of the new algorithm is analyzed in the Appendix. The conclusion is that stability is not a limitation: the algorithm is stable for any choices of dr and dt that are reasonable on other grounds.

In Fig. 1 we show an example of the implementation of this algorithm on an IBM-PC, for a square-well scattering problem in which the reflection coefficient is approximately $\frac{1}{2}$.

Figure 1 is a screen dump from a Turbo Pascal program for the IBM PC which implements the algorithm described in this paper. It offers the user a menu of potentials and wave functions, which are stored on files. The program and scenario files may be downloaded via BITNET from the Phys-L physics file list by sending the message GET SCHRODZ EXE to LISTSERV AT UWF. (In the VM operating system, the proper syntax is TELL LISTSERV AT UWF GET SCHRODZ EXE.) The total time required on a 20 MHz 30386 for the calculation and display shown is 7 s; even on a 6 MHz AT it is only 20 s. Thus the new algorithm brings the time down to the order of magnitude of a typical student attention span.

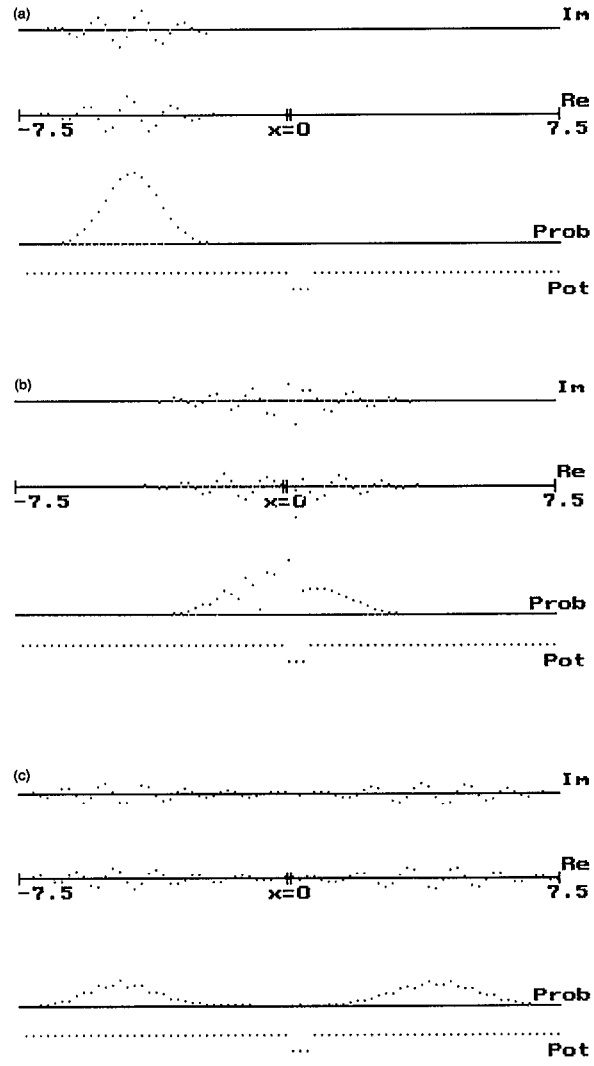

FIG. 1. Wave packet of center wavelength 6 Δr ($\Delta r = 0.02$, $\Delta t = 0.2$), incident on a square well of depth 29 and width 3 Δr. (a) Initial condition. From top, graphs are I, R, probability P, and potential V. (b) Wave function during the scattering process, showing interference of the incident and backscattered waves. (c) Wave functions after scattering, showing reflected and transmitted waves.

III. CONCLUSION

In summary, we have seen that the new staggered-time algorithm has the advantages of both explicit and implicit algorithms: like explicit ones, it is very fast, and like the implicit unitary (Crank–Nicholson) algorithm it is stable, second-order accurate, and conserves probability exactly. Its speed and simplicity enable the practical use of computer simulation as a pedagogical tool in quantum mechanics.

ACKNOWLEDGMENTS

The author would like to thank Harvey Gould and Jan Tobochnik for encouraging him to write up this work for publication.

APPENDIX

An important criterion for evaluating algorithms of the sort discussed in this paper is stability. There is no easy way to treat the case of variable potential V, so we will assume V can be regarded as a constant locally, to establish a sort of "local stability." As is usually done, we look at the evolution of a wave function with sinusoidal position dependence $\psi = e^{ikx}$, which is an eigenvector of H with eigenvalue

$$h = 2\sin^2(\tfrac{1}{2}k\Delta r)/m\Delta r^2 + V. \tag{A1}$$

To analyze stability, we denote the eigenvalue of $H\,dt$ by A:

$$A \equiv h\,dt \tag{A2}$$

and write Eq. (7a) in matrix form:

$$\begin{pmatrix} R(t+\frac{1}{2}\Delta t) \\ I(t) \end{pmatrix} = \begin{pmatrix} 1 & A \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R(t-\frac{1}{2}\Delta t) \\ I(t) \end{pmatrix}. \tag{A3a}$$

This cannot be regarded as the evolution matrix of the system, because the column vectors on the left and right are not of the same form (in the column vector on the right, I is at a later time than R, whereas on the left the reverse is true.) But if we rewrite Eqs. (7b) similarly:

$$\begin{pmatrix} R(t-\frac{1}{2}dt) \\ I(t) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -A & 1 \end{pmatrix} \begin{pmatrix} R(t-\frac{1}{2}\Delta t) \\ I(t-\Delta t) \end{pmatrix} \tag{A3b}$$

and substitute Eq. (A3b) into Eq. (A3a), we obtain the evolution matrix as the product of the above matrices:

$$\begin{pmatrix} R(t+\frac{1}{2}\Delta t) \\ I(t) \end{pmatrix} = \begin{pmatrix} 1-A^2 & A \\ -A & 1 \end{pmatrix} \begin{pmatrix} R(t-\frac{1}{2}\Delta t) \\ I(t-\Delta t) \end{pmatrix}. \tag{A4}$$

The eigenvalues of this matrix are

$$1 - \tfrac{1}{2}A^2 \pm A\,[A^2/4 - 1]^{1/2}. \tag{A5}$$

If $|A| > 2$, it is easy to see that at least one eigenvalue is less than -1, so the system is unstable for this wave vector. If $|A| \leqslant 2$, there are two complex-conjugate eigenvalues of modulus 1, and the system is stable. To get a condition for overall stability (at all wave vectors) note that h [Eq. (A1)] has its maximum value of $2/m\,\Delta r^2 + V$ for $k = \pi/\Delta r$, and its minimum value of V for $k = 0$. Thus the most unstable wave is a constant or a corrugation wave that alternates sign from site to site, and overall stability is guaranteed if

$$\frac{-2}{\Delta t} \leqslant V \leqslant \frac{2}{\Delta t} - \frac{2}{m\,\Delta r^2} \quad \text{(stability condition)}. \tag{A6}$$

For any Δr and V, this can be achieved by making Δt small enough.

It should be noted that the unitary implicit method is unconditionally stable for any choices of Δr, V, and Δt.[1] Its evolution matrix can be seen [Eq. (5)] to be $(1 + \frac{1}{2}iH\,\Delta t)^{-1}(1 - \frac{1}{2}iH\,\Delta t)$, whose eigenvalues all have unit magnitude for *all* k, unlike the staggered-time algorithm which has unstable eigenvalues if the eigenvalue A of $H\,\Delta t$ exceeds 2. This sounds like an advantage of the unitary implicit method, but, in fact, is not in any practical sense. The exact evolution matrix is $\exp(iH\,\Delta t)$, and by expanding both matrices in powers of $H\,\Delta t$ we can see that the fractional error made (per time step) is of order $(H\,\Delta t)^3$, i.e., A^3. Thus if we make Δt large enough to induce instability in our new algorithm (that is, $A > 2$ for the shortest-wavelength wave), the fractional error in the unitary implicit algorithm is of order 1 *per time step*! Clearly, no information is then contained in these short waves, and we would be foolish not to increase Δr, which would make either algorithm stable. (It might be argued that if there are small and unimportant regions where V, and hence H, is large, one might be willing to sacrifice accuracy there for stability. However, this could be achieved equally well by truncating V at its stability limit.)

REFERENCES

1. S. Koonin, *Computational Physics* (Benjamin–Cummings, Menlo Park, CA, 1986), pp. 161–177.
2. W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in Pascal* (Cambridge U.P., Cambridge, 1988).
3. H. Gould and J. Tobochnik, *Computer Simulation Methods* (Addison-Wesley, New York, 1988), Part 1, p. 120.