JOURNAL OF COMPUTATIONAL PHYSICS 97, 337–351 (1991)

Accuracy of Product-Formula Algorithms

C. B. VREUGDENHIL

Institute of Meteorology and Oceanography, University of Utrecht, P.O. Box 80.000, NL-3508 TA Utrecht, The Netherlands

Received November 3, 1989; revised July 20, 1990

To obtain accurate solutions to the time dependent Schrödinger equation, a "product formula" algorithm has been proposed in literature which is both explicit and unconditionally stable. In this paper the accuracy of this algorithm is considered for a few basic problems in mathematical physics, viz. the convection equation and the diffusion equation, in addition to the Schrödinger equation. From a theoretical analysis, which is confirmed by numerical experiments, it is concluded that the product-formula method can indeed produce accurate solutions, but only for small time steps so that the unconditional stability is not of very much use. A comparison is also made with standard finite difference methods, such as leap-frog and Crank–Nicholson.

1. INTRODUCTION

In order to obtain time-accurate solutions of the multi-dimensional Schrödinger equation, an algorithm using product formulas (PF) has been put forward by De Raedt [1]. It is a splitting method, the idea of which is, of course, not new [4, 6]. The splitting is, however, not based on dimensions or processes with different time scales, but on a division of the grid into overlapping subsets of two grid-points each, for which the semi-discretized equations can be solved exactly with respect to time. The resulting algorithm is efficient, very well vectorizable and can be extended to multi-dimensional cases relatively easily. This paper has the purpose to see whether the idea is useful more generally in computational physics. Therefore, the method is described in this paper for three schematic cases: the Schrödinger, convection (simple-wave), and diffusion equations. The notation is given in standard mathematical form in contrast with the theoretical-physics notation of De Raedt's paper.

2. DESCRIPTION FOR SCHRÖDINGER EQUATION

The model equation is:

$$\frac{\partial c}{\partial t} - iK\frac{\partial^2 c}{\partial x^2} = 0. \tag{1}$$

0021-9991/91 $3.00

Semi-discretizing this on a grid with size Δx, using central differences in space, a set of ordinary differential equations results:

$$\frac{d\mathbf{c}}{dt} + A\mathbf{c} = 0, \tag{2}$$

where A is a tridiagonal matrix

$$A = \begin{bmatrix} a_2 & a_3 & 0 & 0 & \cdots & a_1 \\ a_1 & a_2 & a_3 & 0 & \cdots & 0 \\ 0 & a_1 & a_2 & a_3 & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_3 & 0 & 0 & 0 & \cdots & a_1 a_2 \end{bmatrix}$$

with $a_1 \Delta t = a_3 \Delta t = -\frac{1}{2}\lambda i$, $a_2 \Delta t = \lambda i$, and $\lambda = 2K \Delta t / \Delta x^2$. Some problems in theoretical physics have this discrete form intrinsically. For simplicity, we consider periodic boundary conditions, such that $c_{j+m} = c_j$. The formal solution of Eq. (2) over one time step Δt is

$$\mathbf{c}_{n+1} = \exp(-A\,\Delta t)\mathbf{c}_n. \tag{3}$$

As shown by Strang [7], if $A = A_1 + A_2$ this can be split as follows

$$\mathbf{c}_{n+1} = \exp(-A_1\,\Delta t)\exp(-A_2\,\Delta t)\,\mathbf{c}_n. \tag{4}$$

The PF splitting is based on overlapping pairs of grid points:

$$A_1 = \begin{bmatrix} \frac{1}{2}a_2 & a_3 & 0 & 0 & \cdot \\ a_1 & \frac{1}{2}a_2 & 0 & 0 & \cdot \\ 0 & 0 & \frac{1}{2}a_2 & a_3 & \cdot \\ 0 & 0 & a_1 & \frac{1}{2}a_2 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

and A_2 similarly with the blocks shifted over one grid interval. This is a first-order approximation in time unless the operators $A_{1,2}$ commute [1, 4, 5, 7], which is not the case here. In this way, the system has been split into small 2×2 subsystems which can be solved exactly. It is shown in the Appendix that the exact solution of one such system amounts to multiplication by the matrix

$$B = e^{-i\lambda/2} \begin{pmatrix} \cos\frac{1}{2}\lambda & i\sin\frac{1}{2}\lambda \\ i\sin\frac{1}{2}\lambda & \cos\frac{1}{2}\lambda \end{pmatrix} \tag{5}$$

which is constant for linear problems. It is easily checked that the eigenvalues of B are unity in absolute value, so that the method is unconditionally stable. Yet the algorithm is effectively explicit due to the analytic evaluation of the matrix B.

However, it may be argued that the method is implicit in principle as it requires the solution of 2×2 systems of equations for the field variables at the new time level. That the matrices corresponding to these systems are inverted analytically does not change the principle. Similar single-grid-interval schemes are known under the names of Preissmann or Wendroff [8] for hyperbolic equations and for diffusion-like problems a comparable idea is given in [3].

3. Accuracy for Wave Propagation: Semi-Discretization

The accuracy is analyzed here in terms of harmonic wave propagation, which can be interpreted physically [8]. The solution of the differential equation with periodic boundary conditions is

$$c(x, t) = \exp\{ik(x - Kkt)\}. \tag{6}$$

The corresponding exact solution of the semi-discrete system (2) is

$$c_j(t) = \exp\{ij\xi - 2iKt(1 - \cos \xi)/\Delta x^2\}, \tag{7}$$

where $\xi = k\,\Delta x = 2\pi/m$ ($m =$ number of grid points per wave length). Comparison of (6) and (7) shows that there is a relative phase shift or relative propagation speed

$$c_r = 2(1 - \cos \xi)/\xi^2 \tag{8}$$

which should be close to unity. There is no amplitude error. The expression (8) is illustrated in Fig. 1 which shows that c_r approaches unity with second-order accuracy for small ξ as expected.

4. Accuracy of Time Discretization

The discrete solution defined by Eq. (4) has the form

$$\mathbf{c}_n = G^n \mathbf{c}_0, \tag{9}$$

where $n = t/\Delta t$ is the number of time steps and G is the amplification matrix for a complete time step. For one wave period of the semi-discrete solution $n = 1/\lambda(1 - \cos \xi)$. Due to the splitting, the eigenvectors of G can be expected to have some alternating structure:

$$u_{2j} = \alpha^2 u_{2j-2} = \alpha^{2j} u_0; \qquad u_{2j+1} = \alpha^2 u_{2j-1} = \alpha^{2j} u_1$$

with constants α and u_1 to be determined and normalized such that $u_0 = 1$ for simplicity. The periodic boundary condition requires that $\alpha^m = 1$ or $\alpha_j = \exp(2\pi ij/m)$,

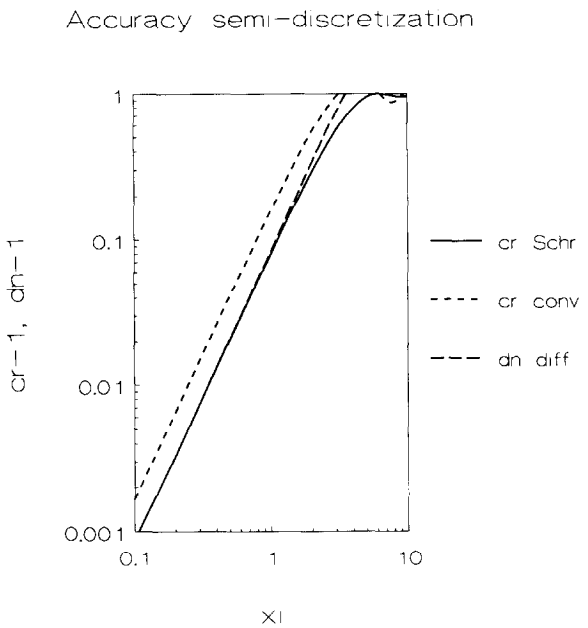

FIG. 1. Accuracy of semi-discretization for the Schrödinger, convection, and diffusion equations as a function of $\xi = 2\pi/m$ (m = number of points per wave length).

$j = 1, m$. The initial condition allows only the value $j = 1$. There are two corresponding eigenvalues $\rho_{1,2}$ with eigenvectors $\mathbf{u}^{(1,2)}$ which have been determined numerically. The numerical solution takes the form

$$\mathbf{c}_n = C_1 \rho_1^n \mathbf{u}^{(1)} + C_2 \rho_2^n \mathbf{u}^{(2)} \tag{10}$$

with coefficients $C_{1,2}$ determined by the initial condition. One of these terms corresponds to the exact solution and the other is parasitic. Then even- and odd-numbered values separately behave as harmonic functions which can be compared with the semi-discrete solution (7) to identify the effect of time discretization in terms of wave damping and phase shift. There is no physical damping, so a numerical damping factor can be defined as

$$d_n = |\rho|^n. \tag{11}$$

The argument of ρ gives rise to a phase shift, or a relative speed of propagation:

$$c_r = -\arg(\rho)/\lambda(1 - \cos \xi). \tag{12}$$

Both quantities should be close to unity to have any accuracy. For the "physical" wave component, the behaviour of c_r and $u_1^{(1)}$ is shown numerically in Fig. 2. It is remarkable that $1 - c_r$ depends quadratically on Δt, even though the method as a

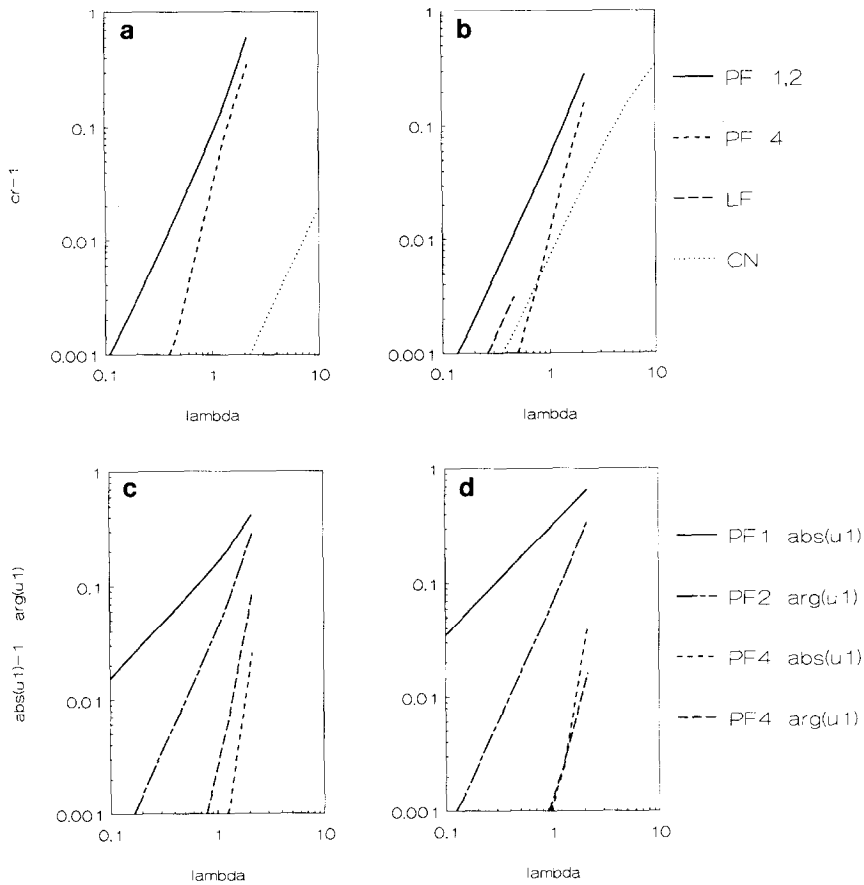

FIG. 2. Accuracy of time discretization for Schrödinger equation, (a), (b) relative wave speed for $m = 20$ and 8 points, respectively, per wave length; (c), (d) error in eigenvector generating spurious mode for $m = 20$ and 8, respectively. PF = product formula with order indicated; LF = leap-frog; CN = Crank–Nicolson. In (a), the line for LF is below the figure margin.

whole is only first-order accurate. There is no wave damping. It is found that $u_1^{(1)}$ is real, so only its absolute value is shown.

The physical component will have a small oscillation around the exact initial condition. This oscillation will be compensated by the spurious component with strength

$$C_2 = (1 - u_1^{(1)})/(u_1^{(2)} - u_1^{(1)})$$

which will then persist in the solution but behave in a completely unphysical way.

5. Higher Order Variants

It is well known [4, 7] that a second-order variant can be constructed by a symmetric formula:

$$\mathbf{c}_{n+1} = \exp(-\tfrac{1}{2}A_1\,\Delta t)\exp(-A_2\,\Delta t)\exp(-\tfrac{1}{2}A_1\,\Delta t)\,\mathbf{c}_n. \tag{13}$$

The half time-step operations can be combined except at those instants of time where one wants to have output. Thus, the amount of work is hardly greater than for the first-order method.

The accuracy analysis in terms of wave propagation can be performed along the same lines as above. It can be shown that the eigenvectors have the same structure as before and, due to the periodic boundary conditions, we find the same values for α as above. The values of ρ' and $u_1^{(1)}$ have been determined numerically. The damping and wave-speed characteristics are the same as those for the first-order method (Fig. 2). The value of $u_1^{(1)}$ is now complex with unit absolute value. Its phase angle, shown in Fig. 2c, behaves with second-order accuracy in time.

De Raedt, being interested in long-time integrations, developed a variant of fourth-order accuracy in time, which is perhaps less standard. The idea is to include a correction step in the second-order method, such that the second-order error is cancelled:

$$\begin{aligned}\mathbf{c}_{n+1} = {} & \exp(-\tfrac{1}{2}A_1\,\Delta t)\exp(-\tfrac{1}{2}A_2\,\Delta t)\exp(\Delta t^3 K) \\ & \times \exp(-\tfrac{1}{2}A_2\,\Delta t)\exp(-\tfrac{1}{2}A_1\,\Delta t)\,\mathbf{c}_n\end{aligned} \tag{14}$$

with $K = [A_1 + 2A_2, [A_1, A_2]]/24$ in which $[A_1, A_2] = A_1A_2 - A_2A_1$ by definition. Some details are given in the Appendix.

The numerical properties are determined in the same way as above. The results are included in Fig. 2. The factor $u_1^{(1)}$ is now complex and clearly shows the fourth-order dependence on λ as does the relative wave speed.

If the resolution gets worse, i.e., the number of points per wave length m gets smaller, the time-accuracy of the PF methods is not very much affected (Figs. 2a and 2b). However, the spatial phase error does, of course, get larger, so any advantage of time accuracy is useful only if spatial errors are not important for some reason (e.g., because the problem is intrinsically discrete).

6. Numerical Experiments

Some numerical experiments were performed to check the theoretical conclusions. One single harmonic wave was considered with periodic boundary conditions. The results are shown for $m = 20$ points per wave length in Fig. 3 in comparison with the exact semi-discrete solution. The spatial discretization gives a phase error of about 1%.

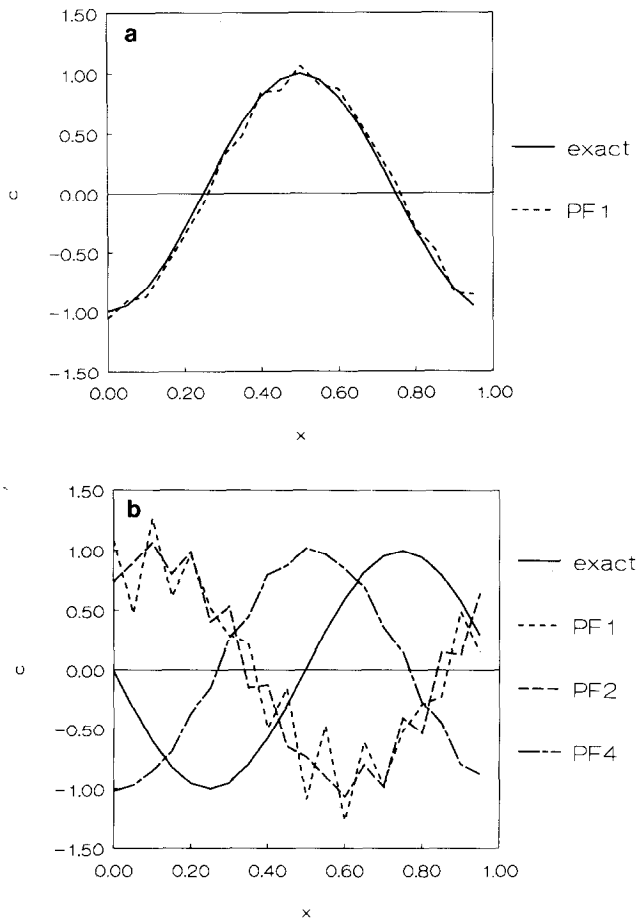

FIG. 3. Numerical experiments with first-order PF method for the Schrödinger equation for $\lambda = 0.5$ (a) or 2 (b) compared with exact semi-discrete solution.

Considering the wave speed according to Fig. 2a, we expect $c_r = 1.02$ for PF1 and PF2 at $\lambda = 0.5$ and 0.998 for PF4. At $\lambda = 2$, the values are 1.6 and 0.65, respectively. In Figs. 3a and b this is indeed observed.

The observation of the spurious component is a little more involved as it interferes with the oscillating part of the main wave and both oscillations propagate at completely different speeds. In the worst case, the two reinforce one another to the double amplitude and Fig. 3 refers to the times approximately at which this occurs, i.e., at 0.5 wave period in Fig. 3a and 0.8 in 3b. The amplitudes expected from the theory for PF1 are 0.035 at $\lambda = 0.5$ and 0.18 at $\lambda = 2$ and this agrees with Fig. 3. For the PF2 and PF4 methods, smaller values are observed which have not been checked numerically, but the general behaviour is in agreement with the theory.

However, even for the fourth-order method the spurious oscillations are non-negligible at this value of λ. The conclusion is that solutions with any reasonable accuracy can only be obtained for small λ.

7. Comparison with Finite-Difference Methods

A very simple explicit finite-difference method for solving the Schrödinger equation (1) is the leap-frog method:

$$c_j^{n+1} - c_j^{n-1} = i\lambda(c_{j+1}^n - 2c_j^n + c_{j-1}^n), \tag{15}$$

where $c_j^n = c(j\,\Delta x, n\,\Delta t)$. The spatial discretization is identical to that in Eq. (2) so the same numerical error is involved. Solutions of the form $c_j^n = \rho^n \exp(ij\xi)$ result with

$$\rho_{1,2} = -iz \pm \{1 - z^2\}^{1/2},$$

where $z = \lambda(1 - \cos\xi)$. It is easily checked that the requirement for stability is $\lambda \leqslant \frac{1}{2}$. Of the two possible roots, one is spurious. The latter is set up by the second initial condition required by Eq. (15). It has the same wave length as the physical wave, but it oscillates in time. This may introduce a very unpleasant time-splitting behaviour, which can, however, be controlled by introducing some smoothing in time. For some applications, this may not be acceptable (e.g., [2]). Note that the type of spurious solution is different in behaviour from the one in the PF method.

For the leap-frog method there is no wave damping ($|\rho| = 1$). The accuracy of the time-discretization is expressed by the relative wave speed for the physical component in comparison with the semi-discrete solution:

$$c_r = \operatorname{atan}\{z/\sqrt{1 - z^2}\}/z \tag{16}$$

The error is so small that it does not show up in Fig. 2a; it does, however for smaller resolution ($m = 8$) in Fig. 2b.

Another standard finite-difference method that can be used for comparison is the Crank–Nicolson method:

$$c_j^{n+1} - c_j^n = i\lambda/4(c_{j+1}^{n+1} - 2c_j^{n+1} + c_{j-1}^{n+1} + c_{j+1}^n - 2c_j^n + c_{j-1}^n). \tag{17}$$

This method is unconditionally stable but requires the solution of a tridiagonal system of equations. In multi-dimensional cases, a splitting method will be used (alternating direction). The method is of second-order accuracy in space and time, does not produce any wave damping nor spurious solutions, and its relative wave speed in comparison with the semi-discrete solution reads:

$$c_r = 2\,\operatorname{atan}(\tfrac{1}{2}z)/z \tag{18}$$

(see Fig. 2). Both finite-difference methods compare favourably with the PF methods for not-too-coarse grids ($m = 20$). However, they are much more sensitive to coarser grids than PF.

In one dimension with m grid points, the PF1 and PF2 methods require $8m$ multiplying operations per time step; PF4 even $32m$. For Crank–Nicolson, the operation count is $6m$ and for leap-frog, $2m$; so both will be more efficient than the PF methods even if the accuracies are the same. Figure 2 shows that only PF4 will eventually be more efficient than the others if a very high accuracy is sought.

8. Convection Equation

As a second example of great practical interest we consider the convection or simple-wave equation

$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} = 0. \tag{19}$$

The procedure for the PF algorithm is the same as above. The coefficients of the semi-discretised equation are

$$a_1\,\Delta t = -\tfrac{1}{2}\sigma, \qquad a_2 = 0, \qquad a_3\,\Delta t = \tfrac{1}{2}\sigma,$$

where $\sigma = u\,\Delta t/\Delta x$ is the Courant number. This gives

$$B = \begin{pmatrix} \cos\tfrac{1}{2}\sigma & -\sin\tfrac{1}{2}\sigma \\ \sin\tfrac{1}{2}\sigma & \cos\tfrac{1}{2}\sigma \end{pmatrix}. \tag{20}$$

It is a standard exercise to show that the relative phase error for the semi-discrete solution is

$$c_r = \sin\xi/\xi \tag{21}$$

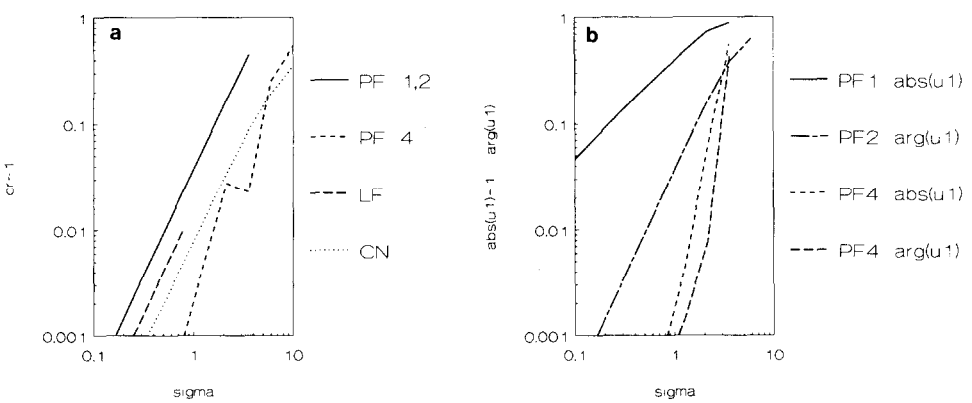

FIG. 4. As Fig. 2, for convection equation, $m = 20$.

(cf. Fig. 1). The number of time steps to cover one semi-discrete wave period is $n = 2\pi/\sigma \sin \xi$. Analysis of the time accuracy for the PF methods shows that there is no wave damping. The relative speed of propagation is given in Fig. 4, together with the values for the finite-difference methods, which turn out to be identical to Eqs. (16) and (18) if z is replaced by $z = \sigma \sin \xi$. The leap-frog method is stable if $|\sigma| \leqslant 1$. The "wiggle" in the line for PF4 is caused by a change of sign of $c_r - 1$, which is only coarsely represented in the figure. Coarse-grid values are not shown, but the behaviour is similar to the Schrödinger case. Some numerical experiments are shown in Fig. 5.

Again, there is no difference in numerical wave speed between the first- and second-order PF methods. Theoretically, relative wave speeds for PF1 or PF2, and PF4 are 0.99 and 1.0002 for $\sigma = 0.5$ and 0.82 and 1.025 for $\sigma = 2$. These values can be recognized in Fig. 5.

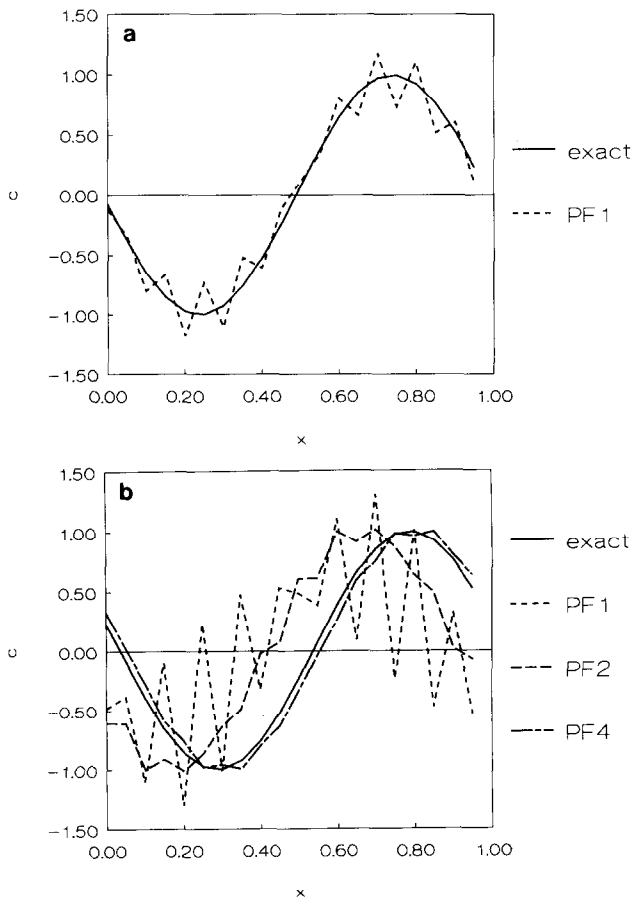

FIG. 5. Numerical experiments with the PF method (orders 1, 2, and 4) for the convection equation with Courant number $\sigma = 0.5$ (a) or 2 (b).

The spurious components are again shown at the worst instant of time. For PF1 the theoretical amplitudes are 0.14 for $\sigma = 0.5$ and 0.2 for $\sigma = 2$, which is in agreement with the figures, where the doubled values are seen. Figure 5b shows the better accuracy of PF2. The fourth-order method is considerably more accurate, at least for moderate values of σ. In this case, it is also more accurate than the LF and CN methods.

9. Diffusion Equation

Finally, the same procedure is applied to the standard diffusion equation

$$\frac{\partial c}{\partial t} - D\frac{\partial^2 c}{\partial x^2} = 0. \tag{22}$$

Semi-discretising gives

$$a_1\,\Delta t = a_3\,\Delta t = -\tfrac{1}{2}\lambda, \qquad a_2\,\Delta t = \lambda,$$

where $\lambda = 2D\,\Delta t/\Delta x^2$ is the diffusion parameter. This gives

$$B = e^{-\lambda/2}\begin{pmatrix} \cosh\tfrac{1}{2}\lambda & \sinh\tfrac{1}{2}\lambda \\ \sinh\tfrac{1}{2}\lambda & \cosh\tfrac{1}{2}\lambda \end{pmatrix}. \tag{23}$$

The semi-discrete solution shows only an error in the damping rate. After one relaxation time $t = (k^2 D)^{-1}$, the ratio between the amplitudes of the semi-discrete and continuous solutions is

$$d = \exp\{1 - 2(1 - \cos\xi)/\xi^2\} \tag{24}$$

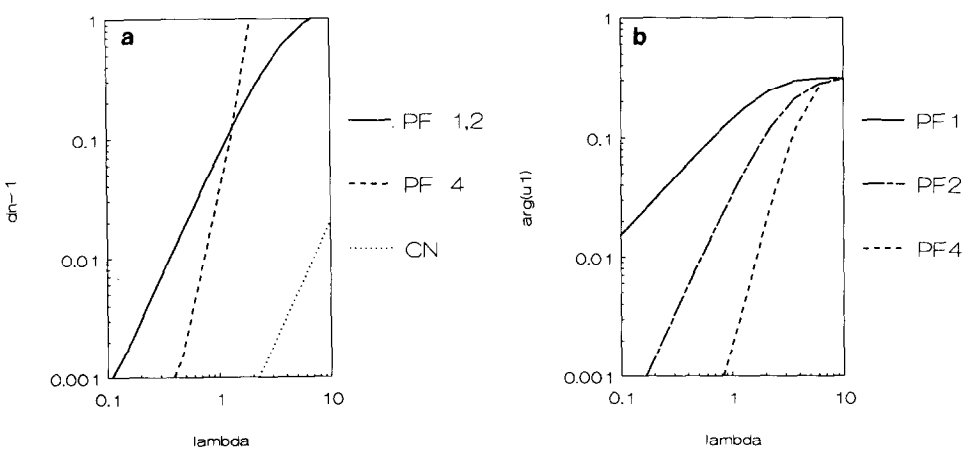

FIG. 6. As Fig. 2 for diffusion, $m = 20$. Wave-propagation now characterized by wave-damping factor d_n.

as shown in Fig. 1. The number of time steps is chosen to cover one relaxation time for the semi-discrete case $n = 1/\lambda(1 - \cos \xi)$.

The PF methods and the standard finite-difference methods have no phase errors either. A measure for the numerical damping due to time discretization is

$$d_n = e\,|\rho|^n. \tag{25}$$

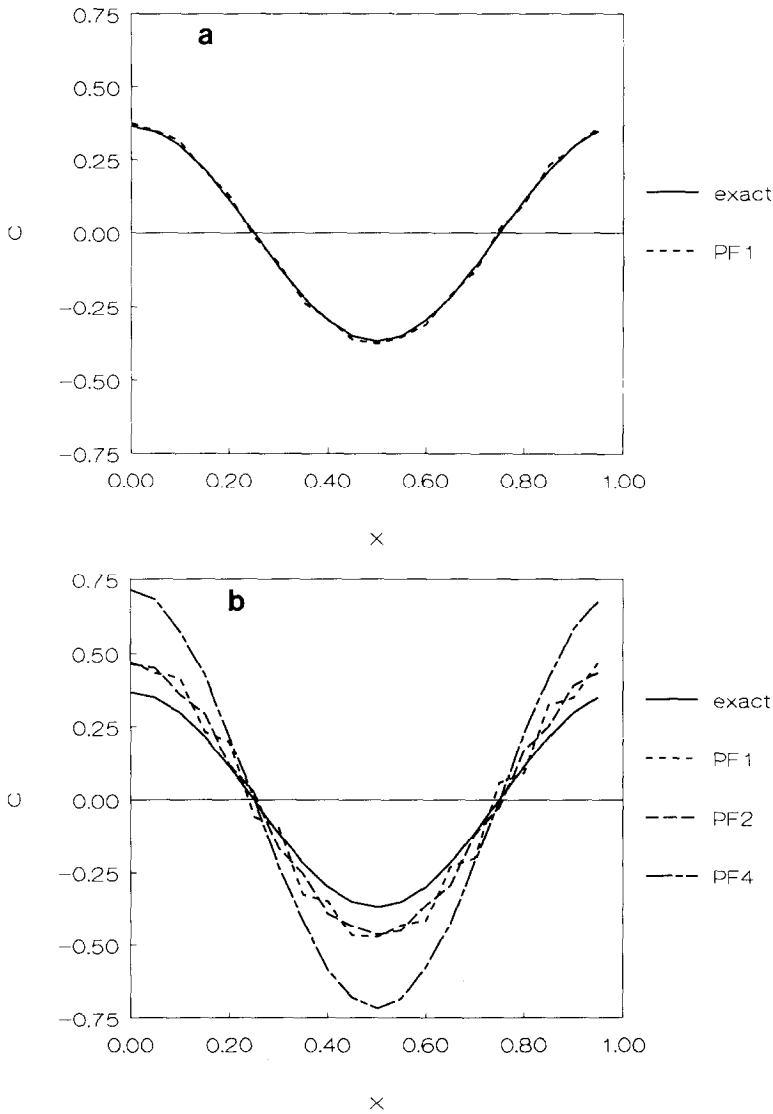

FIG. 7. Numerical experiments with the PF method (orders 1, 2, and 4) for the diffusion equation with diffusion parameter $\lambda = 0.5$ (a) and 2 (b).

The leap-frog method cannot be used in this case as it is unconditionally unstable. For the Crank–Nicholson method we find a damping factor

$$d_n = \left| \frac{1 - \frac{1}{2}z}{1 + \frac{1}{2}z} \right|^{1/z} \tag{26}$$

with $z = \lambda(1 - \cos \xi)$. The various relations are illustrated in Fig. 6. The behaviour is comparable to the previous cases. It is to be noted that the fourth-order correction matrices C_i (cf. Appendix) in this case have real eigenvalues, one of which is greater than unity (though very slightly), which means that PF4 is actually unstable for the diffusion equation. This does not show up for moderate values of λ, but a very large growth factor was indeed observed in a numerical experiment for $\lambda = 5$.

Some numerical experiments are shown in Fig. 7. For $\lambda = 0.5$ accurate solutions are found, although the spurious component is still visible. For $\lambda = 2$, the amplitude error is quite apparent; theoretical values of $d_n = 1.28$ for PF1 or PF2, and 2.3 for PF4 are confirmed. The spurious components do not propagate, so the amplitude-doubling effect does not occur; they have, however, a different damping rate and therefore get more serious as time progresses. Only PF4 is acceptable in this respect.

10. Conclusions

The product-formula algorithm is equivalent to standard finite difference methods as far as spatial differencing is concerned. The potential advantage of unconditional stability combined with an effectively explicit algorithm turns out to be unexploitable, as large numerical errors will be involved.

Spurious oscillations may occur in all PF methods; they are reasonably small only for moderate values of σ or λ. These oscillations are particularly important in the PF1 case. The fourth-order PF algorithm may be used to obtain a high-accuracy time integration; it will eventually be superior to standard (second-order) finite-difference methods.

An attractive feature of the PF methods is that their time-dicretization errors are only slightly affected by the spatial resolution, so that good temporal accuracy can be obtained in cases where a modest spatial resolution is sufficient.

Appendix

The construction of the matrix B for the product-formula algorithm is illustrated here for the Schrödinger case. Consider one pair of grid points from Eq. (4); the matrix is

$$A_1 \, \Delta t = \Delta t \begin{pmatrix} \frac{1}{2}a_2 & a_3 \\ a_1 & \frac{1}{2}a_2 \end{pmatrix}.$$

It has eigenvalues $\mu_{1,2}$ with corresponding left and right eigenvectors $\mathbf{w}_{1,2}$ and $\mathbf{v}_{1,2}$ such that

$$\mathbf{w}_j^{\mathrm{T}} A_1 \, \Delta t = \mu_j \mathbf{w}_j^{\mathrm{T}} \qquad \text{and} \qquad A_1 \, \Delta t \, \mathbf{v}_j = \mu_i \mathbf{v}_j$$

and normalization

$$\mathbf{w}_i^{\mathrm{T}} \cdot \mathbf{v}_j = \delta_{ij}.$$

For some initial vector $\mathbf{c}$, we can write

$$\mathbf{c} = \sum C_j \mathbf{v}_j \qquad \text{with} \quad C_j = \mathbf{w}_j^{\mathrm{T}} \cdot \mathbf{c}.$$

Then

$$\exp(-A_1 \, \Delta t)\mathbf{c} = \sum C_j \exp(-A_1 \, \Delta t)\mathbf{v}_j = \sum C_j \exp(-\mu_j)\mathbf{v}_j = B\mathbf{c}$$

with

$$B = \mathbf{v}_1 \mathbf{w}_1^{\mathrm{T}} e^{-\mu_1} + \mathbf{v}_2 \mathbf{w}_2^{\mathrm{T}} e^{-\mu_2}.$$

For the present case, we find

$$\begin{aligned} &\mu_1 = 0, && \mathbf{v}_1^{\mathrm{T}} = (1, 1), && \mathbf{w}_1^{\mathrm{T}} = \tfrac{1}{2}(1, 1) \\ &\mu_2 = \lambda i, && \mathbf{v}_2^{\mathrm{T}} = (1, -1), && \mathbf{w}_2^{\mathrm{T}} = \tfrac{1}{2}(1, -1) \end{aligned}$$

which gives the matrix B mentioned in Eq. (5).

The fourth-order correction shown in Eq. (14) involves the matrix

$$K = a_1^2/12 \begin{bmatrix} 0 & -2a_1 & 0 & -a_3 & 0 & 0 & 0 & 0 & \cdot \\ -2a_3 & 0 & a_1 & 0 & 2a_3 & 0 & 0 & 0 & \cdot \\ 0 & a_3 & 0 & -2a_1 & 0 & -a_3 & 0 & 0 & \cdot \\ -a_1 & 0 & -2a_3 & 0 & a_1 & 0 & 2a_3 & 0 & \cdot \\ 0 & 2a_1 & 0 & a_3 & 0 & -2a_1 & 0 & -a_3 & \cdot \\ & & \cdot & & \cdot & & \cdot & & \cdot \end{bmatrix}.$$

This can be split into four matrices involving pairs of grid points, only one of which is shown as an example,

$$K_1 = a_1^2/12 \begin{bmatrix} 0 & 0 & 0 & -a_3 & 0 & 0 & 0 & 0 & \cdot \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdot \\ 0 & 0 & 0 & 0 & 0 & -a_3 & 0 & 0 & \cdot \\ -a_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdot \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -a_3 & \cdot \\ 0 & 0 & -a_1 & 0 & 0 & 0 & 0 & 0 & \cdot \end{bmatrix}.$$

In the same way as for the basic algorithm, we can write

$$\exp(\Delta t^3 K_j) = C_j$$

which consists of 2×2 matrices for pairs of grid points. A complete time-step for the fourth-order method then reads

$$\mathbf{c}_{n+1} = B_E B_O C_1 C_2 C_3 C_4 B_O B_E \mathbf{c}_n$$

in which E and O evidently indicate even and odd pairs of points. There is now no possibility of combining sub-steps as in the second-order case, so that the total amount of work is abour four times greater than for the first- or second-order methods.

ACKNOWLEDGMENTS

This work was performed at the Netherlands Energy Research Foundation ECN (Petten, The Netherlands). Discussions with Prof. dr. H. de Raedt and P. de Vries from the University of Amsterdam and G. P. Leendertse from ECN have contributed significantly to this paper.

REFERENCES

1. H. DE RAEDT, *Comput. Phys. Rep.* 7, No. 1, 1 (1987).
2. V. M. DWYER AND D. L. WEAIRE, *Phil. Mag. B* **53**, No. 1, L35 (1986).,
3. D. J. EVANS AND A. R. ABDULLAH, in *Proc. 3d Int. Conf. Numerical Methods in Thermal Problems, Seattle* (Pineridge Press, Swansea, 1983), p. 330.
4. A. R. GOURLAY, in *The State of the Art in Numerical Analysis*, edited by D. Jacobs (Academic Press, New York, 1977), p. 757.
5. R. J. LEVEQUE AND J. OLIGER, *Math. Comput.* **40**, 162, 469 (1983).
6. G. I. MARCHUK, *Methods of Numerical Mathematics* (Springer-Verlag, New York, 1975).
7. G. STRANG, *SIAM J. Num. Anal.* **5**, 506 (1968).
8. C. B. VREUGDENHIL, *Computational Hydraulics* (Springer-Verlag, Berlin, 1989).