JOURNAL OF COMPUTATIONAL PHYSICS 102, 398–406 (1992)

An Adaptive Algorithm for Spectral Computations on Unbounded Domains

A. CLOOT AND J. A. C. WEIDEMAN

Department of Applied Mathematics, University of the Orange Free State, Bloemfontein, 9300, South Africa

Received March 23, 1990; revised May 9, 1991

We consider spectral methods for solving differential equations on unbounded domains. In particular, we consider a spectral rational function method and a method based on domain truncation combined with Fourier series. Both methods contain a free parameter which determines the accuracy of the spectral method. When solving evolution equations the optimal parameter may change with time resulting in a significant loss of accuracy if the parameter is kept fixed. We develop an automatic algorithm which adapts the parameter at regular time levels to maintain optimum accuracy. In numerical tests we found that the proposed method is robust and efficient. © 1992 Academic Press, Inc.

1. INTRODUCTION

Spectral methods for solving differential equations have become very popular in the last decade or two. (For a general introduction to these methods, see for example the monographs by Gottlieb and Orszag [1] or Canuto *et al.* [2].) The methods are based on global interpolants, in contrast to the methods of finite differences or finite elements. For periodic problems the global interpolant is a Fourier series and, when Dirichlet or Neumann conditions are enforced, a Chebyshev, Legendre, or some other polynomial bases are used.

For problems on unbounded domains the question of which basis to use seems not quite settled yet. The competitors are Hermite polynomials (Funaro and Kavian [3]), sine functions (Stenger [4]), rational functions (Boyd [5]), or simply truncating the domain and then using Fourier or Chebyshev expansions, possibly combined with a mapping (Grosch and Orszag [6]).

In this paper we shall consider two of these approaches, namely, rational functions and domain truncation combined with Fourier series. The reason for selecting these two approaches is that they can both be implemented efficiently with the fast Fourier transform (FFT).

In both the domain truncation and rational function approaches there is a free parameter L that needs to be fixed. In the domain truncation approach it is the size of the truncated computational domain, and in the rational function method it is a stretching variable. Obviously the optimum choice for such a parameter would be the one that maximizes the accuracy of the spectral method.

Some theory exists for estimating these optimal parameters, see, for example, Weideman and Cloot [7] or Boyd [8]. However, these formulas are sometimes difficult to use and obviously require some knowledge of the function we are approximating. When solving evolution equations, for instance, the properties of the function are usually changing with time. Even if the optimal L corresponding to the initial condition can be found theoretically, it may soon become non-optimal, resulting in a significant decrease in accuracy. In this paper we discuss an automatic algorithm for estimating the optimal parameter.

The outline of the paper is as follows. In Section 2 we give a brief overview of Fourier differentiation on periodic domains. In Section 3, we extend these ideas to unbounded domains, introducing both the domain truncation and rational function approaches. We describe the updating algorithm in Section 4, and in Section 5 we apply it to two test problems. Finally, in Section 6 we investigate numerically whether the increase in accuracy justifies the extra cost of implementing the algorithm.

2. FOURIER DIFFERENTIATION ON A PERIODIC GRID

The Fourier pseudospectral differentiation process can be described as follows: Let $v(s)$ be a 2π-periodic function, and let $v_j = v(s_j)$ denote the values of the function sampled on the uniform grid $s_j = 2\pi j/N$, $j = -N/2, ..., N/2 - 1$. The discrete Fourier series of the data v_j is defined as

$$v_j = \sum_{k=-N/2}^{N/2-1} c_k e^{iks_j}, \tag{1}$$

0021-9991/92 $5.00

where the coefficients c_k are given by

$$c_k = \frac{1}{N} \sum_{j=-N/2}^{N/2-1} v_j e^{-iks_j}. \tag{2}$$

Once the coefficients c_k have been computed, the approximate derivative $v_j' \approx v'(s_j)$ can be computed through

$$v_j' = \sum_{k=-N/2+1}^{N/2-1} ikc_k e^{-iks_j}, \tag{3}$$

a process requiring two FFTs, i.e., $O(N \log N)$ operations. For smooth functions this differencing process is very accurate. We quote the following results from Tadmor [9]. For functions $v(z)$ analytic in the strip

$$-\eta < \mathrm{Im}(z) < \eta,$$

we have

$$\max_{|j| \leq N/2} |v_j' - v'(s_j)| \leq C_N \exp(-\tfrac{1}{2}\eta N), \tag{4}$$

where C_N is a slowly varying function of N. The error therefore decays essentially exponentially as $N \to \infty$, and the decay rate is determined by the position of the nearest singularity to the real axis. For non-analytic functions the convergence rate is no longer exponential, but of the form N^{-s} with s bounded only by the regularity of the solution.

Note that in the case of an analytic function, the asymptotic rate of decay of the Fourier coefficients of the function $v(s)$ is

$$c_k = O(\exp(-\eta\,|k|)), \tag{5}$$

as $|k| \to \infty$, see for example Dieudonné [10, p. 280]. This means that the accuracy of the Fourier differentiation process is of the same order of magnitude as the last Fourier coefficient retained in the expansion. We shall exploit this fact in our adaptive algorithms, by optimizing the accuracy of the differentiation through minimizing the magnitude of the highest Fourier coefficient.

3. FOURIER DIFFERENTIATION ON AN INFINITE GRID

In this paper we are only concerned with Fourier differentiation on the infinite interval. We shall consider two approaches to this problem namely (1) domain truncation, and (2) rational functions.

Domain Truncation

We truncate $x \in (-\infty, \infty)$ to $x \in [-L, L]$. Then the latter interval is scaled linearly to $s = [-\pi, \pi]$ through

$$s = \pi x / L \tag{6}$$

and Fourier differentiation can be applied as described in Section 2. Obviously this introduces artificial periodicity, and hence a discontinuity at $x = \pm L$. The Fourier differentiation process is therefore not as accurate as with analytic periodic functions (see Weideman and Cloot [7], for a discussion).

Rational Functions

An alternative to the above domain truncation approach is the use of rational functions. We introduce the mapping

$$x = L \tan(s/2), \qquad s \in [-\pi, \pi]; \tag{7}$$

and let $v(s) = u(x)$. Then the Fourier series representation of $v(s)$, given by

$$\begin{aligned} v(s) &= \sum_{k=-\infty}^{\infty} c_k e^{iks} \\ &= a_0/2 + \sum_{k=1}^{\infty} a_k \cos(ks) + b_k \sin(ks), \end{aligned} \tag{8}$$

becomes

$$u(x) = a_0/2 + \sum_{k=1}^{\infty} a_k C_k(x) + b_k S_k(x), \tag{9}$$

where $C_k(x) = \cos(2k \tan^{-1}(x/L))$ and $S_k(x) = \sin(2k \tan^{-1}(x/L))$ turn out to be rational functions of x (see Boyd [5]).

We use the points $x_j = L \tan(s_j/2)$, where $s_j = 2\pi j/N$, $j = -N/2, \ldots, N/2 - 1$, as collocation points. The derivative of the data $u_j = u(x_j)$ can be computed through the chain rule:

$$\left.\frac{du}{dx}\right|_{x=x_j} = (2/L)\cos^2(s_j/2) \left.\frac{dv}{ds}\right|_{s=s_j}. \tag{10}$$

The derivative dv/ds is computed by the Fourier-differencing process described by Eq. (2) and (3), with $v_j \equiv u_j$. In both the domain truncation and the rational function approaches we have to fix the free parameter L, for a given N. Obviously the optimal choice would be the one that maximizes the accuracy of the differentiation process (10). The demonstration of the existing of such an optimal value is made in [7], where it is shown that the optimal

choice, $L = L_{\text{opt}}$, can be determined through an asymptotic balance between the resolution and boundary errors arising during the computation.

All theoretical estimates of the optimal parameters rely, however, on a priori knowledge of the properties of the function to be differentiated. Obviously this is not always available, and in the next section we shall describe an automatic procedure for estimating the parameter, given only the values of the function to be differentiated.

4. THE ALGORITHM

We start by noting that both the differentiation procedures described in the previous section can be expressed as

$$\left.\frac{du}{dx}\right|_{x=x_j} = g'(x_j)\left.\frac{dv}{ds}\right|_{s=s_j}, \tag{11}$$

where $u(x) = v(s)$, and $s = g(x)$ is either given by the linear mapping (6) (domain truncation), or the tangent mapping (7) (rational functions). Since the differentiation $g'(x)$ is done analytically, the error is committed in the Fourier differentiation dv/ds. We therefore aim to select the parameter L in (6) and (7) so that this can be done as accurately as possible.

Our strategy is based on the following assumption. Fourier differentiation of $v(s)$ will be accurate if the interpolation of $v(s)$ by a Fourier series is good. This will be the case if the Fourier coefficients decay rapidly. We therefore aim to select L in order to minimize the magnitdue of the highest Fourier coefficient retained in the expansion

$$\Gamma = |c_{-N/2}| = \frac{1}{N}\left|\sum_{j=-N/2}^{N/2-1}(-1)^j v_j\right|,$$

where $v_j = v(s_j) = u(x_j)$. To verify that the above strategy is indeed practical, we made the following test. We compared the error

$$E = \max_{|j|\le N/2} |u'(x_j) - u_j'|, \tag{12}$$

where u_j' is the numerically computed derivative (11), with the magnitude of Γ, for a test function $u(x) = \text{sech } x$'s. The results, for both domain truncation and rational function, reproduced on Figs. 1a–b show clearly that the minimum value of Γ and the minimum value of the error (12) occur at approximately the same value of L.

Thus, at first sight, the monitoring of the optimal value

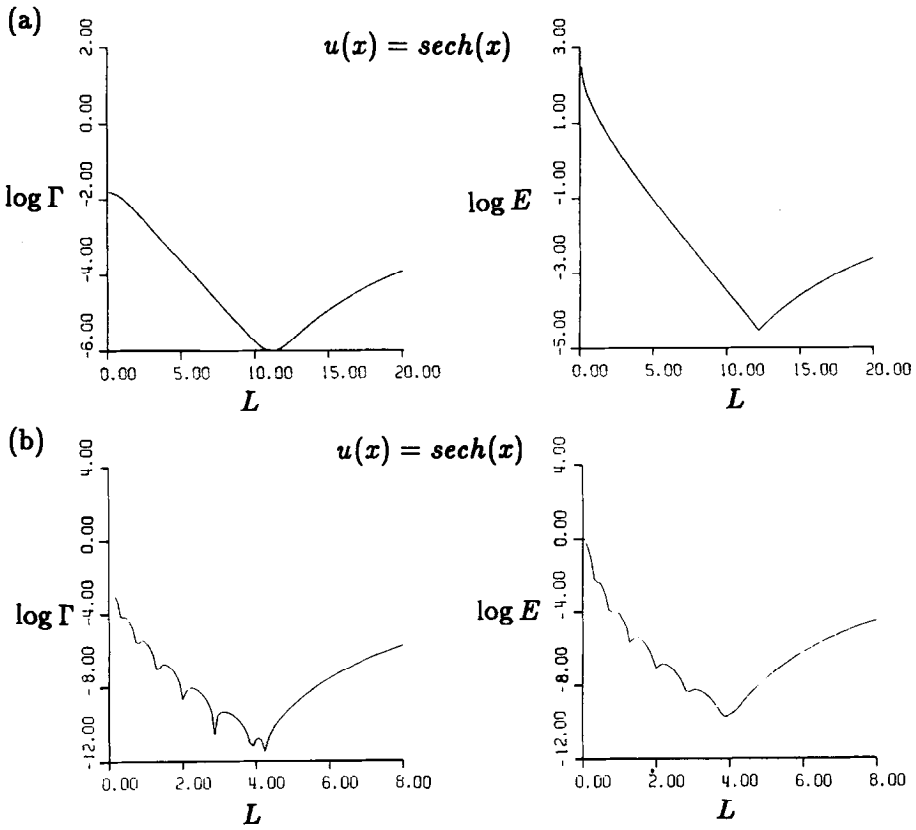

FIG. 1. Comparison of the error $\log_{10} E$ (as defined by (12)) and the magnitude of the highest Fourier coefficient $\log_{10} \Gamma$, for different values of L when either, domain truncation (a) or rational functions (b) are used; $N = 64$.

L_{opt} through an algorithm based on the satisfaction of the additional condition

$$\partial c_{-N/2}/\partial L = \sum_{j=-N/2}^{N/2-1} (-1)^j j u'(x_j) = 0$$

seems to be relevant.

Numerical tests of this procedure reveal that it is working well when dealing with the domain truncation method. However, it cannot be extended to the case of rational functions. The problem is that in this case the graph of $c_{-N/2}$ as a function of L has many local minima (compare Fig. 1), and convergence to a non-optimal L is easily obtained. We therefore disregard this approach in favor of a cruder, but more robust, algorithm. We simply compute $\Gamma = |c_{-N/2}|$ for different values of L, and the optimum L is then obtained through a least squares quadratic fit.

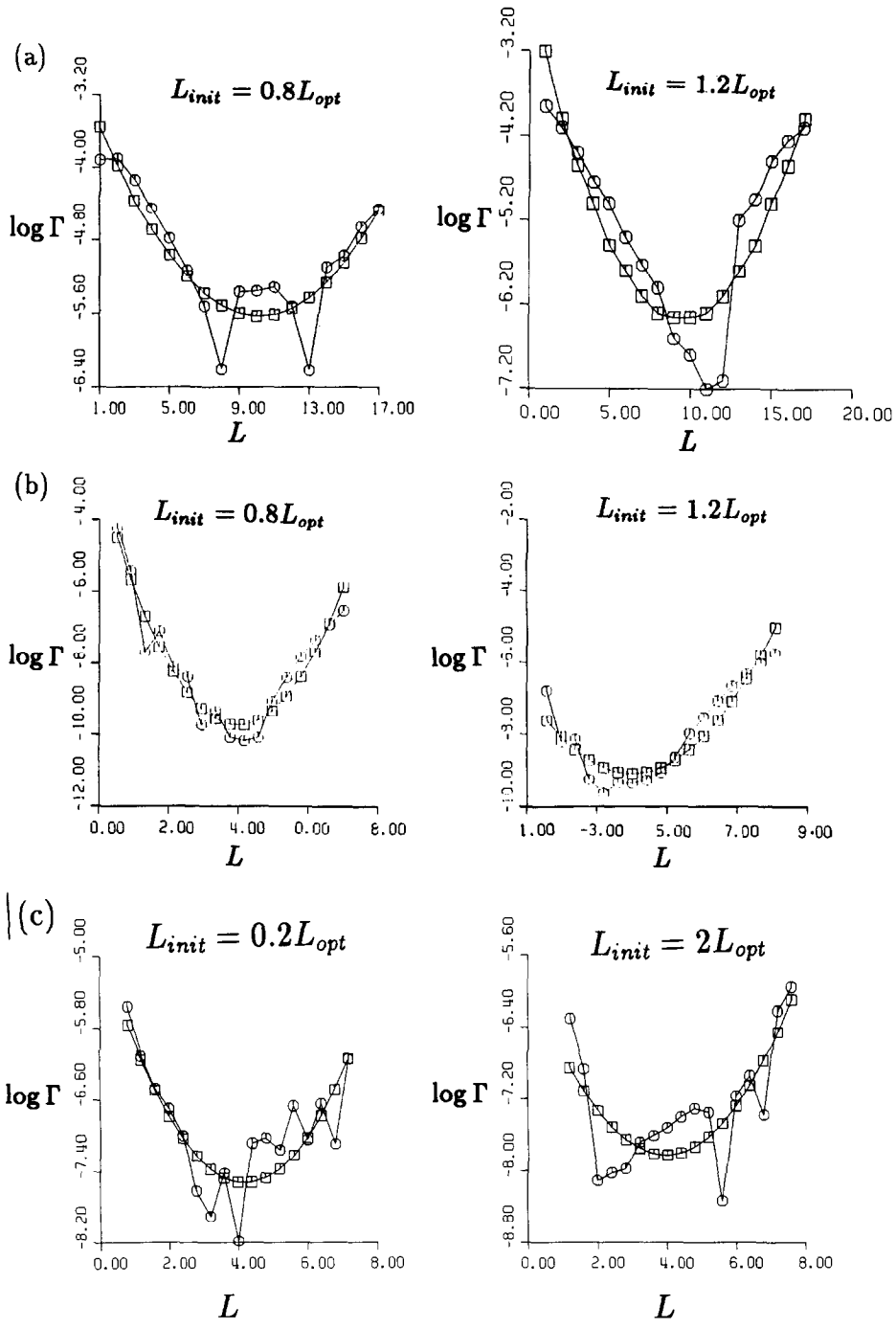

FIG. 2. The values of $\log_{10} \Gamma$ as computed in the algorithm for different values of L (circles), together with the least squares fit $\log_{10} \Gamma = aL^2 + bL + c$ (squares) when using (a) domain truncation; (b), (c) rational functions; $N = 64$.

ALGORITHM. Given the discrete function u_j, $j=-N/2, ..., N/2-1$, this algorithm finds an estimate for the optimal parameter L. It is assumed that the initial data u_j correspond to a grid with mapping parameter $L=L_{\text{init}}$.

Step 1. Choose a value of L and compute $x_j = Ls_j/\pi$ (or $x_j = L\tan(s_j/2)$), where $s_j = 2\pi j/N$, and $v_j = v(s_j) = u(x_j)$, $j=-N/2, ..., N/2-1$, from the initial data u_j, by means of Fourier interpolation.

Step 2. Compute

$$\Gamma = |c_{-N/2}| = \frac{1}{N}\left|\sum_{j=-N/2}^{N/2-1}(-1)^j v_j\right|. \tag{13}$$

Step 3. Repeat steps 1 and 2 for different values of L.

Step 4. Fit the data to the statistical model

$$\log \Gamma = aL^2 + bL + c \tag{14}$$

and determine the minimum of the parabola. This provides the approximation to L_{opt}.

To test this algorithm, we made the following comparisons. We initially supplied the values of a test function $u(x) = \operatorname{sech} x$ on a $N=64$ grid corresponding to a value $L=L_{\text{init}}$, with L_{init} chosen to be non-optimal. It was then left to the algorithm to try and find the optimal L.

In a first set of experiments we have chosen L_{init} to be 20% less than the optimal L (L_{opt} was determined from Figs. 1a, b), and for the other 20% more. The results are shown in Figs. 2a (domain truncation) and 2b (rational functions). The circles denote the values of Γ as calculated by (13), and the squares denote the quadratic fit (14). In all cases L ranged through seventeen equidistributed values in Step 3 of the algorithm.

These figures should be compared with Figs. 1a and b, respectively. In particular, the curves denoted by the circles in Figs. 2a and b are supposed to be approximations of the curves $\log \Gamma$ vs. L in Figs. 1a and b. These approximations are good, except in the case of domain truncation. Obviously, this problem arises when one is searching for function values at gridpoints *outside* the range $x \in [-L_{\text{init}}, L_{\text{init}}]$, where *extrapolation* is required. If we assume a decay $u(x) \to 0$, as $x \to \infty$, this can be achieved by setting simply $u \equiv 0$ for values of x outside $[-L_{\text{init}}, L_{\text{init}}]$. This introduces large errors in the interpolation process, however, unless the decay of $u(x)$ is very rapid. We therefore found that the algorithm is not very effective when used in conjunction with domain truncation. Of course, the same problem does not arise with rational functions, since the entire interval $x \in (\infty, \infty)$ is mapped to $s \in [-\pi, \pi]$ through $x = L\tan(s/2)$. In this latter case, the quadratic fit manages to find a reasonable approximation for the optimal value of L even we, deliberately, choose L_{init} to be a poor approximation of L_{opt}. This assertion is supported in Fig. 2c showing the fitting obtained when considering an initial L that is either 80% less or 100% more than the optimal value.

Note one could use more sophisticated minimization algorithms but we found that the simple quadratic fit works well in practice. As mentioned earlier, the function usually contains a number of local minima and the quadratic fit serves to smooth the curve. At any rate, one is only interested in obtaining an estimate for L to within say 10%, not 1%.

5. IMPLEMENTATION FOR EVOLUTION EQUATIONS

We consider evolution equations of the form

$$u_t = F(x, u, u_x, u_{xx}), \qquad -\infty < x < \infty.$$

By the usual method of lines we approximate this equation by the semi-discrete system

$$\frac{du_j}{dt} = F(x_j, u_j, u_j', u_j''), \quad j=-N/2, ..., N/2-1, \tag{15}$$

where u_j' and u_j'' are computed by the differentiation processes described in Section 3, and $u_j(t) \approx u(x_j, t)$. The standard Runge–Kutta fourth-order method was used to integrate this system of ordinary differential equations in time.

We implemented the algorithm of the previous section as follows. Since the initial optimal value of the parameter corresponding to the initial condition is not always known accurately, we use the algorithm to find a relevant initial approximate value of L_{opt}. Then the time integration is carried out for a pre-assigned number of time steps, say M, at which time the function values are sent to algorithm and a new approximation to L_{opt} is computed. The function values on the new grid are then computed through Fourier interpolation and the time integration continues for M further time steps, and so on.

To test the proposed scheme we chose two problems with known analytical solutions.

PROBLEM 1. The nonlinear Schrödinger equation

$$u_t = iu_{xx} + 8i|u|^2 u, \qquad -\infty < x < \infty,$$

where $i^2 = -1$. With the initial condition

$$u(x, 0) = \operatorname{sech} x$$

the exact solution is a "breather" soliton

$$u(x, t) = e^{it}\left\{\frac{4 + 3(e^{i8t} - 1)\,\mathrm{sech}^2 x}{4 - 3\sin^2(4t)\,\mathrm{sech}^4 x}\right\}\mathrm{sech}\, x. \qquad (16)$$

PROBLEM 2. The outflow problem

$$u_t = -xu_x, \qquad -\infty < x < \infty.$$

With the initial condition

$$u(x, 0) = g(x),$$

the exact solution is

$$u(x, t) = g(xe^{-t}).$$

We start with Problem 1, using rational functions, $N = 64$, and a time step of $\Delta t = 0.001$. In Fig. 3a we show the analytical solution, together with the error

$$E = \max_{|j| \le N/2} |u_j(t) - u(x_j, t)|, \qquad (17)$$

as a function of time, for the updating and fixed L algorithms. For the fixed L algorithm, we selected the optimal value corresponding to the initial condition. In the updating algorithm we considered the optimal parameter value corresponding to the initial condition can only be roughly guessed and we provided $L_{\text{init}}(t = 0) = 2L_{\text{opt}}(t = 0)$. Further, we performed an updating of L at every $M = 20$ steps. In Step 3 of the algorithm, we let L range through nine values, distributed uniformly around the current estimate of the optimal parameter.

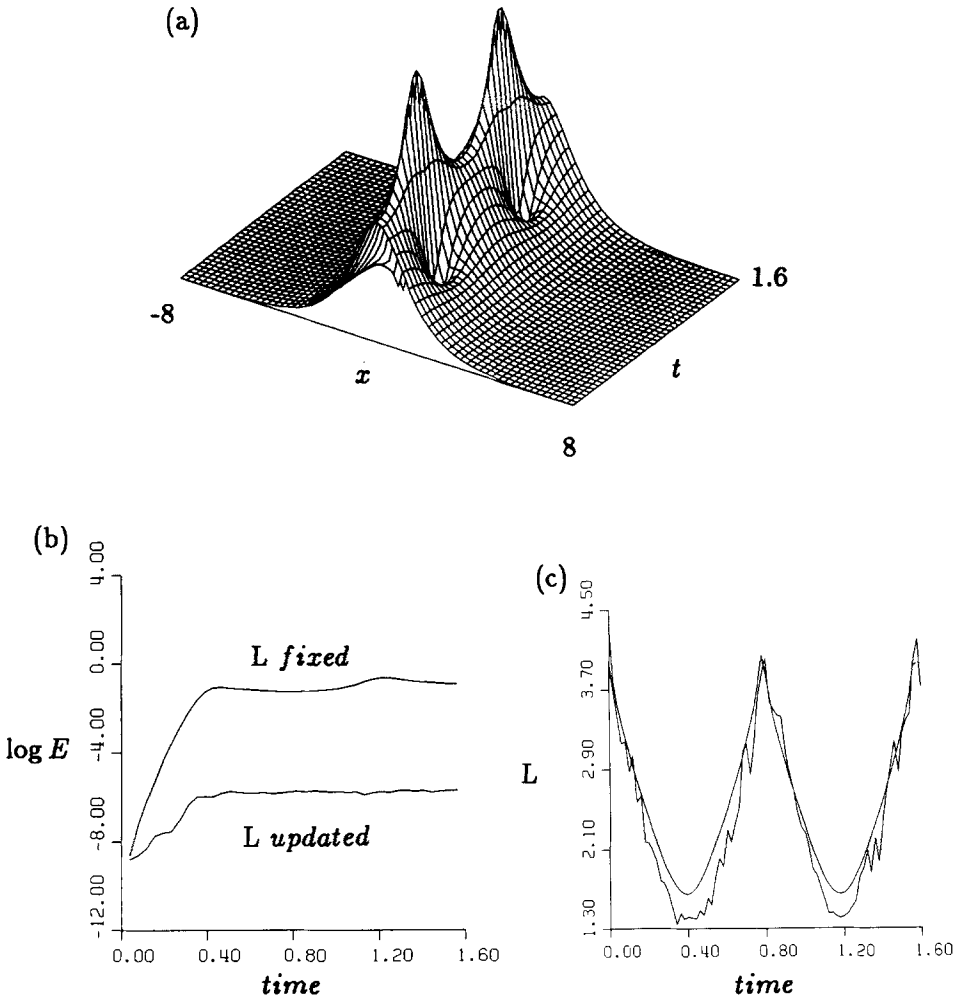

FIG. 3. (a) Theoretical solution of Problem 1 (the modulus is shown); (b) error in the numerical solutions of this problem; and (c) the time evolution of the parameter L_{opt}. The smooth line denotes the theoretical estimate (18), and the jagged line is the estimate of L_{opt} as computed by the updating algorithm. The parameters are $M = 20$, $N = 64$, and $\Delta t = 0.001$.

Note in Fig. 3b that the updating algorithm maintains an accuracy of five to six digits throughout the computation, improving by many orders of magnitude on the fixed L computation.

In Fig. 3c we show the evolution of the parameter L_{opt} as computed by the updating algorithm (jagged line) and compare it with its optimal expression

$$L_{opt}(t) = \{\tfrac{1}{2}N \arccos^2 [(\tfrac{3}{4}\sin^2 4t)^{1/4}]\}^{1/3}, \qquad (18)$$

obtained through theoretical consideration (see Weideman and Cloot [7]).

As observed in Fig. 3c, the updating algorithm is capable of tracking the optimal value of L for a considerable period of time. Moreover, the algorithm appears to be robust and self-correcting.

For this problem domain, truncation failed. The updating algorithm could not improve significantly on the fixed L algorithm, and both methods gave errors much larger than when we used rational functions. This failure of the updating algorithm is obviously due to the problems associated with Fourier extrapolation, as was discussed in the previous section.

Applying the algorithm to Problem 2 with similar initial condition, but starting with the initial rough guess $L_{init}(t=0) = 0.2L_{opt}(t=0)$, yielded the results shown in Figs. 4a–c. Once again, the updating algorithm, used in conjunction with rational functions, improves by orders of magnitude on the fixed L algorithm. In this case, the updating algorithm shows its ability to follow the time evolution of L_{opt} even when a fast exponential variation occur. For comparisons purpose, we represent on Fig. 4c this time evolution as prescribed by the algorithm together with its analytical equivalent

$$L_{opt}(t) = \tfrac{1}{2}(\pi^2 N)^{1/3} e^t. \qquad (19)$$

We have now established that the updating algorithm does indeed lead to a significant increase in accuracy, at least for the two test problems considered. However, the repeated Fourier interpolations in the algorithm are not inexpensive. Is the increase in accuracy sufficient to justify the added cost? We attempt to answer this in the next section.

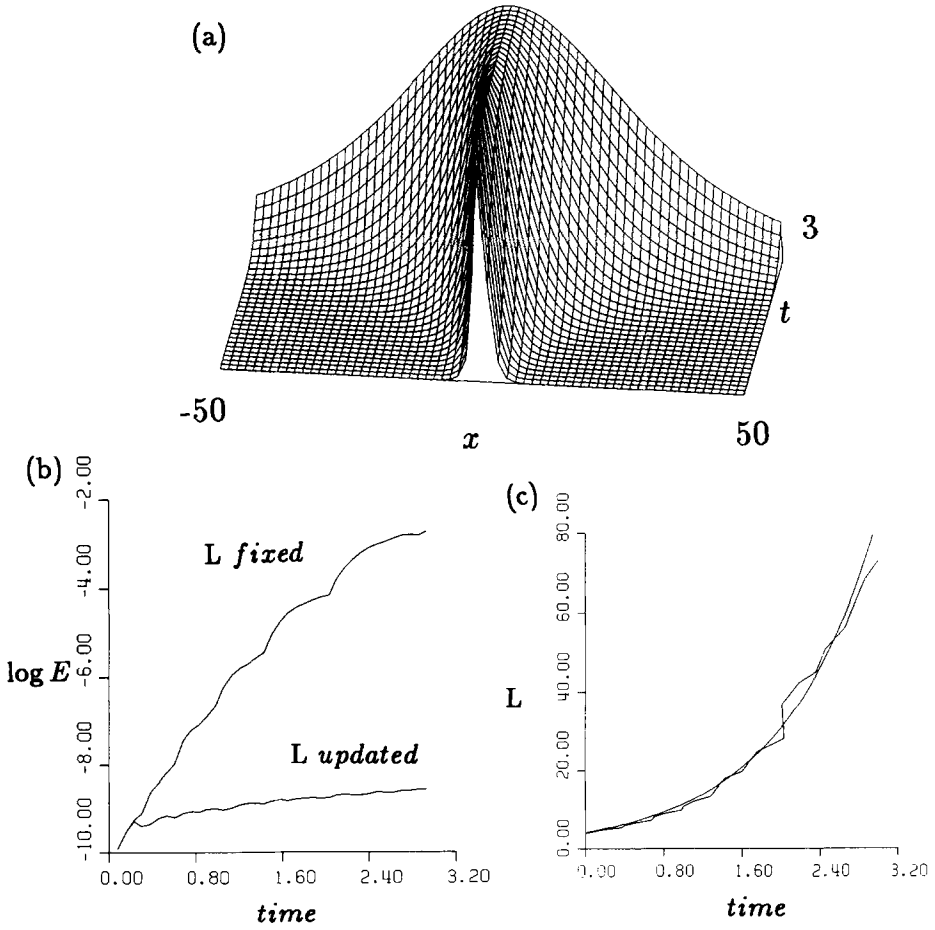

FIG. 4. Same as Fig. 3, but we consider Problem 2. The parameters are $M = 10$, $N = 64$, and $\Delta t = 0.01$.

TABLE I

Error E in the Numerical Solution of Problem 1 at Time $t = 0.4$, as well as the CPU Time c

	N	
M	32	64
10	$E = 3.310^{-4}$ $c = 0.203$	$E = 8.510^{-7}$ $c = 0.587$
20	$E = 2.910^{-4}$ $c = 0.159$	$E = 1.110^{-6}$ $c = 0.406$
40	$E = 2.910^{-4}$ $c = 0.130$	$E > 10^{-6}$
100	$E = 5.410^{-4}$ $c = 0.116$	$E > 10^{-6}$

Note. The updating algorithm is used. The time step was $\Delta t = 0.001$.

TABLE III

Error E in the Numerical Solution of Problem 2 (Initial Condition $u(x, 0) = \operatorname{sech} x$), at Time $t = 3$, as well as the CPU Time c

	N	
M	32	64
15	$E = 1.110^{-5}$ $c = 0.209$	$E = 3.110^{-9}$ $c = 0.645$
30	$E = 4.010^{-5}$ $c = 0.177$	$E = 6.910^{-8}$ $c = 0.435$
45	$E > 10^{-4}$	$E = 1.510^{-7}$ $c = 0.387$

Note. The updating algorithm is used. The time step was $\Delta t = 0.01$.

6. EFFICIENCY

To see whether the extra cost incurred by the updating procedure is indeed worth the effort, we made the following tests. We only use rational functions since we have already established that the updating algorithm is not very successful in the domain truncation approach.

We start with Problem 1 and in Table I we list the error

$$E = \max_{j \le N/2} |u_j(t) - u(x_j, t)|$$

computed at $t = 0.4$ (this corresponds roughly to the time that the breather reaches its first peak; see Fig. 3). We also list the normalized cpu time; the cpu time corresponding to the integration with $N = 256$ and a fixed L being chosen as cpu unit. For comparison, we list in Table II the results for the fixed L computation, where we have selected the optimal L corresponding to the initial condition. In all experiments the same time step $\Delta t = 0.001$ in the time integration scheme was used. (For this value of Δt we obtained convergence as far as the time integration is concerned, for all cases listed in Tables I and II.) We used nine values of L in Step 3 of the algorithm, as described before.

A comparison of Tables I and II reveals that the updating algorithm reaches the same accuracy as the fixed L computation using roughly four times fewer points in space. This amounts to reaching the same accuracy in approximately one-half to one-third of the computational time, provided the number of steps between updating is not unreasonably small. We conclude that the updating algorithm is indeed efficient.

We also give the corresponding results for Problem 2, with initial condition $u(x, 0) = \operatorname{sech} x$, in Tables III and IV. Again we observe that the updating algorithm can reach an accuracy comparable with that of the fixed L algorithm using significantly less cpu time.

Note that for Problem 2 the parameter needs to be updated more regularly than in Problem 1. This is, of course, due to the fact that for Problem 2 the optimal L grows exponentially away from its initial value. In Problem 1 the optimal L does not wander far off.

TABLE II

Same as Table I but L Is Kept Fixed at the Optimal Value Corresponding to the Initial Condition

N	
128	256
$E = 6.210^{-4}$ $c = 0.464$	$E = 9.310^{-7}$ $c = 1.0$

TABLE IV

Same as Table III, but L Is Kept Fixed at the Optimal Value Corresponding to the Initial Condition

N	
128	256
$E = 6.410^{-5}$ $c = 0.484$	$E = 8.610^{-7}$ $c = 1.0$

7. CONCLUSIONS

We have presented an updating algorithm for spectral computations of evolution equations on unbounded domains. Through numerical tests we have ascertained that the method is robust and efficient. It beats the methods which do not employ updating by roughly a factor of four, regarding the number of gridpoints required to reach a given accuracy, and by approximately a factor of two to three if computational time is the criterion.

We make no claims that the method will be efficient for all problems. In particular, the problems we considered have smooth solutions; we doubt if the present algorithm will be effective if discontinuities develop. But if this is the case the usefulness of spectral methods, as compared to finite differences, becomes questionable anyway.

Moreover, the problems we considered remained localized. For instance, had the solution of the nonlinear Schrödinger equation broken up into solitons travelling at different speeds, the present method will not be as effective. In such a case it may, however, be possible to combine the present idea with a moving grid method. In fact, we implemented a method based on the mappings $x = a(t) + Ls/\pi$ or $x = a(t) + L\tan(s/2)$, where $a(t)$ represents the movement of the grid, and it is chosen to keep the solution as centered on the grid $s \in [-\pi, \pi]$ as possible. This idea worked well in the test cases we considered.

REFERENCES

1. D. Gottlieb, M. Y. Hussaini, and S. A. Orszag, *Theory and Application of Spectral Methods*, edited by R. G. Voigt *et al.* ed. (SIAM, Philadelphia, 1984).
2. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics* (Springer-Verlag, Berlin, 1988).
3. D. Furnaro and O. Kavian, "Approximation of some Diffusion Evolution Equations in Unbounded Domains by Hermite Functions," Prepublication 88/5, Université de Nancy (unpublished).
4. F. Stenger, *SIAM Rev.* **23**, 165 (1981).
5. J. P. Boyd, *J. Comput. Phys.* **69**, 112 (1987).
6. C. E. Grosh and S. A. Orszag, *J. Comput. Phys.* **25**, 273 (1985).
7. J. A. C. Weideman and A. Cloot, *Comput. Methods Appl. Mech. Eng.* **80**, 467 (1990).
8. J. P. Boyd, *J. Comput. Phys.* **45**, 43 (1982).
9. E. Tadmor, *SIAM J. Numer. Anal.* **23**, 1 (1986).
10. J. Dieudonne, *Infinitesimal Calculus* (Hougthon-Mifflin, Boston, 1971), p. 279.