JOURNAL OF COMPUTATIONAL PHYSICS 76, 103–130 (1988)

An Investigation into the Effect of Product Approximation in the Numerical Solution of the Cubic Nonlinear Schrödinger Equation

Y TOURIGNY AND J. LL. MORRIS

Department of Mathematical Sciences, University of Dundee, Scotland

Received December 29, 1986, revised April 24, 1987

We study the effect of product approximation on the Galerkin solutions of the one-dimensional cubic Schrodinger equation A Crank–Nicolson scheme is used to discretize in time The paper describes two numerical experiments· in the first, we examine the approximation obtained by the standard Galerkin method and discuss the possibility of enforcing discrete analogs of the conservation laws satisfied by the exact solution, in the second experiment, numerical results obtained by the product approximation version of the Galerkin method are compared and the effectiveness of the method for different combinations of test and trial functions is also investigated

1. INTRODUCTION

While the Galerkin method plays an important role in the theory of nonlinear evolution equations, it is seldom used in its original standard form as a computational procedure. Product approximation is a technique which consists of replacing the nonlinear term by its interpolant in the finite-dimensional space [1]. This leads to a simplified version of the Galerkin method which removes the need for numerical quadrature in the evaluation of the inner products. The Galerkin method with product approximation has been applied to a number of nonlinear problems [5] including the Korteweg–de Vries equation and the nonlinear Schrödinger equation [3, 4].

The purpose of our present study is to compare the Galerkin method and its modified version using product approximation, as applied to the one-dimensional cubic Schrödinger equation (henceforth CSE):

$$\begin{aligned} &i\frac{\partial u}{\partial t}+\frac{\partial^2 u}{\partial x^2}+|u|^2u=0 \qquad x\in\mathbb{R},\ t>0,\ i=\sqrt{-1} \\ &u(x,0)=f(x) \qquad x\in\mathbb{R}, \end{aligned} \tag{1.1}$$

where the solution u is complex-valued and f is some sufficiently smooth function which decreases exponentially as $|x|$ tends to infinity.

We have chosen this model equation for the following reasons:

0021-9991/88 $3 00

(1) The CSE is one of the few nonlinear equations for which an implementation of the standard Galerkin method is still manageable.

(2) Much is known, both computationally and theoretically about the equation. A convenient feature of the solution is that it satisfies an infinite number of conservation laws [8] including

$$E_1(t)=\int_{\mathbb{R}} |u(x,t)|^2\,dx=\text{const} \qquad \text{for all} \quad t\geqslant 0 \tag{1.2}$$

$$E_2(t)=\int_{\mathbb{R}} \left\{\frac{1}{2}\left|\frac{\partial u}{\partial x}(x,t)\right|^2-\frac{1}{4}|u(x,t)|^4\right\}dx=\text{const} \qquad \text{for all} \quad t\geqslant 0. \tag{1.3}$$

These provide us with a simple means of analysing the numerical results.

(3) The CSE is only one particular member of a wider family of nonlinear Schrödinger equations. These equations have found many applications in science and should be of general interest. A conclusion about the effectiveness of the numerical scheme for the CSE may lead to an efficient scheme for the solution in the general case.

In Section 2, we introduce the notation and derive the approximation methods. In Section 3, we state what and in what sense these methods are meant to approximate and how we propose to judge the quality of the approximation. In Section 4, we examine the numerical results obtained by the standard Galerkin method. We devote some time to the question (which, for the CSE, is of importance) as to whether or not the enforcement of discrete analogs of the conservation properties is advisable. Finally, in Section 5, we turn to the Galerkin method using product approximation. Numerical results are compared and we also consider different choices of test and trial functions.

A Galerkin method leads to a spatial discretization of the original equation and there remains to discretize in time. Naturally, our results will be affected by the way in which this is done. We have opted for a simple Crank–Nicolson scheme which will be used uniformly throughout. We study spatial convergence for a fixed time step.

2. The Galerkin Methods

We make the hypothesis that the solution of (1.1) has compact support on a bounded open interval $I=\,]a,b[$ during the time period $[0,T]$, where T is positive and finite. Under this assumption, (1.1) is equivalent to

$$\left.\begin{aligned} i\frac{\partial u}{\partial t}+\frac{\partial^2 u}{\partial x^2}+|u|^2u&=0 && x\in I,\ 0\leqslant t\leqslant T\\ u(x,0)&=f(x) && x\in I\\ u(a,t)=u(b,t)&=0 && 0\leqslant t\leqslant T;\end{aligned}\right. \tag{2.1}$$

(2.1) is the problem we are going to solve

We have to deal with complex-valued functions defined on I. $L^2(I) = L^2$ denotes the space of square integrable functions. $H^1(I) = H^1$ consists of the elements of L^2 which have a square integrable distributional derivative. $H_0^1(I) = H_0^1$ is the subspace of H^1 formed by those elements which vanish at the endpoints of I. We use the norms

$$\|u\|_{L^2} = \langle u, u\rangle^{1/2} \qquad \|u\|_{H^1} = \{\langle u, u\rangle + \langle u' + u'\rangle\}^{1/2},$$

where $\langle u, v\rangle = \int_I u\bar{v}\,dx$ is the inner product in L^2 and u' denotes the distributional derivative of u.

Let $\{S_n\}$ be a sequence of finite-dimensional subspaces of H_0^1 such that

$$\bigcup_{n=1}^{\infty} S_n \text{ is dense in } H_0^1.$$

We denote by v the dimension of S_n. If $\{\phi_j\}_{j=1}^{v}$ forms a basis of S_n, we write $S_n = [\phi_1, ..., \phi_v]$. The standard Galerkin method for (2.1) consists of defining a sequence $\{u_n\}$ where the general term $u_n\colon [0, T] \to S_n$ written as

$$u_n(t) = \sum_{j=1}^{v} \alpha_j(t)\,\phi_j, \qquad \alpha_j(t) \in C$$

satisfies

$$\begin{aligned} i\langle \dot{u}_n(t), \phi\rangle - \langle u_n'(t), \phi'\rangle + \langle |u_n(t)|^2\, u_n(t), \phi\rangle = 0 \qquad &\text{for all} \quad \phi \in S_n \\ u_n(0) = f_n. \qquad & \end{aligned} \tag{2.2}$$

The dot indicates differentiation with respect to time and f_n is the general term of a sequence converging to f in H_0^1.

Equation (2.2) constitutes a system of ordinary differential equations in time for the unknown coefficients $\alpha_j(t)$. We introduce the uniform time grid $0 = t_0 < t_1 < \cdots < t_M = T$ of gridsize $\Delta t = T/M$, where $M \in \mathbb{N}$.

If $u_n^m = \sum_{j=1}^{v} \alpha_j^m \phi_j$ denotes an approximation to $u_n(t_m)$, the Crank–Nicolson scheme for Eq. (2.2) is obtained by replacing $\dot{u}_n(t)$ by $(1/\Delta t)(u_n^{m+1} - u_n^m)$ and $u_n(t)$ by $\frac{1}{2}(u_n^{m+1} + u_n^m)$:

$$\begin{aligned} i\left\langle \frac{1}{\Delta t}(u_n^{m+1} - u_n^m), \phi\right\rangle - \left\langle \frac{1}{2}(u_n^{m+1} + u_n^m)', \phi'\right\rangle \\ + \left\langle \left|\frac{1}{2}(u_n^{m+1} + u_n^m)\right|^2 \frac{1}{2}(u_n^{m+1} + u_n^m), \phi\right\rangle = 0 \end{aligned}$$

for all $\phi \in S_n$, $m = 0, 1, ..., M-1$,

$$u_n^0 = f_n. \tag{2.3}$$

We briefly discuss the construction of the subspaces $S_n \subset H_0^1$. Introduce the uniform partition

$$\Delta_n: a = x_0 < x_1 < \cdots < x_{n+1} = b$$

and let $h = (b-a)/(n+1)$.

With this partition, we may define finite-dimensional spaces of *polynomial splines* and therefore give a precise meaning to the notions of *interpolation* and *interpolant* [7]. We shall need the following standard polynomial spline spaces:

(1) Piecewise linear functions. We use a basis with typical element

$$\phi_j(x) = \begin{cases} \dfrac{(x - x_{j-1})}{h}, & x_{j-1} \leqslant x \leqslant x_j, \\ \dfrac{(x_{j+1} - x)}{h}, & x_j \leqslant x \leqslant x_{j+1}, \\ 0, & \text{otherwise.} \end{cases}$$

(2) Cubic splines. We use a basis with typical element $B_j(x) = B((x - x_{j-2})/h)$, where

$$B(x) = \begin{cases} \frac{1}{6}(x+2)^3, & -2 \leqslant x \leqslant -1, \\ \frac{1}{6}(x+2)^3 - \frac{4}{6}(x+1)^3, & -1 \leqslant x \leqslant 0, \\ \frac{1}{6}(-x+2)^3 - \frac{4}{6}(-x+1)^3, & 0 \leqslant x \leqslant 1, \\ \frac{1}{6}(-x+2)^3, & 1 \leqslant x \leqslant 2, \\ 0, & \text{otherwise.} \end{cases}$$

(3) Hermite cubics. Typical basis functions are $R_j(x) = R((x - x_j)/h)$ and $T_j(x) = T((x - x_j)/h)$, where

$$R(x) = \begin{cases} (1+x)^2(1-2x), & -1 \leqslant x \leqslant 0, \\ (1-x)^2(1+2x), & 0 \leqslant x \leqslant 1, \\ 0, & \text{otherwise;} \end{cases}$$

$$T(x) = \begin{cases} x(x+1)^2, & -1 \leqslant x \leqslant 0, \\ x(x+1)^2, & 0 \leqslant x \leqslant 1, \\ 0, & \text{otherwise.} \end{cases}$$

Having chosen a particular spline space, S_n will consist of those elements which vanish at the endpoints of I.

Noting that $u_n^m = \sum_{j=1}^{\nu} \alpha_j^m \phi_j$, it is quite obvious that the implementation of a scheme like (2.3) involves tedious calculations, even when the basis functions ϕ_j vanish outside a small subinterval of I. The Galerkin method with product approximation consists of defining a sequence of functions such that the general term $u_n(t) = \sum_{j=1}^{\nu} \alpha_j(t)\,\phi_j$ satisfies

$$\begin{aligned} &i\langle \dot{u}_n(t), \phi\rangle - \langle u_n'(t), \phi'\rangle + \langle \gamma_n(t), \phi\rangle = 0 \qquad \text{for all} \quad \phi \in S_n \\ &\qquad\qquad u_n(0) = f_n, \end{aligned} \tag{2.4}$$

where $\gamma_n(t)$ is the interpolant of $|u_n(t)|^2 u_n(t)$ in S_n (hence a linear combination of the ϕ_j's). The time discretized version of (2.4) is

$$\begin{aligned} &i\left\langle \frac{1}{\Delta t}(u_n^{m+1} - u_n^m), \phi\right\rangle - \left\langle \frac{1}{2}(u_n^{m+1} + u_n^m)', \phi'\right\rangle + \langle \gamma_n^*, \phi\rangle = 0 \\ &\qquad\qquad \text{for all } \phi \in S_n,\ m = 0, ..., M-1, \\ &\qquad u_n^0 = f_n, \end{aligned} \tag{2.5}$$

where γ_n^* is the interpolant of $|\frac{1}{2}(u_n^{m+1} + u_n^m)|^2\, \frac{1}{2}(u_n^{m+1} + u_n^m)$.

3. Spatial Convergence

We study the convergence of the Galerkin methods (2.3), (2.5) for a fixed time step Δt as n tends to infinity. We make the assumption that there is only one set $\{u^0, ..., u^M\} \cup H_0^1$ such that

$$\begin{aligned} &i\left\langle \frac{1}{\Delta t}(u^{m+1} - u^m), \phi\right\rangle - \left\langle \frac{1}{2}(u^{m+1} + u^m)', \phi'\right\rangle \\ &\quad + \left\langle \left|\frac{1}{2}(u^{m+1} + u^m)\right|^2 \frac{1}{2}(u^{m+1} + u^m), \phi\right\rangle = 0 \\ &\qquad\qquad \text{for all} \quad \phi \in H_0^1,\ m = 0, ..., M-1, \\ &\qquad u^0 = f. \end{aligned} \tag{3.1}$$

We define

$$E_1^m = \int_I |u^m|^2\, dx \qquad \text{and} \qquad E_2^m = \int_I \{\tfrac{1}{2}|(u^m)'|^2 - \tfrac{1}{4}|u^m|^4\}\, dx.$$

The two quantities

$$E_1^{n,m} = \int_I |u_n^m|^2\, dx \qquad \text{and} \qquad E_2^{n,m} = \int_I \{\tfrac{1}{2}|(u_n^m)'|^2 - \tfrac{1}{4}|u_n^m|^4\}\, dx$$

are of fundamental importance, for their behaviour entirely determines the quality of the approximation. A proof of the following lemma can be found in [6].

LEMMA (Characterization of spatial convergence). *Let $\{u^0, ..., u^M\}$ be the solution of Eq.* (3.1) *and $\{u_n^0, ..., u_n^M\}$ the solution of Eq.* (2.3). *These two statements are equivalent*:

$$\lim_{n \to \infty} E_1^{n,m} = E_1^m \quad \text{and} \quad \lim_{n \to \infty} E_2^{n,m} = E_2^m, \qquad m = 0, 1, ..., M,$$

$$\lim_{n \to \infty} \|u_n^m - u^m\|_{H^1} = 0, \qquad m = 0, 1, ..., M.$$

Moreover, if $|E_2^{n,m}|$ is uniformly bounded in n and m, we have

$$\lim_{n \to \infty} \|u_n^m - u^m\|_{L^2} = 0, \qquad m = 0, ..., M. \tag{3.2}$$

It will come as a reassuring, though perhaps not unexpected fact, that this result remains valid for the Galerkin method with product approximation and piecewise linear functions as test and trial functions. In this case, however, (3.2) requires the additional hypothesis that also $E_1^{n,m}$ be uniformly bounded.

This brings to our attention the advantage of discrete analogs of the conservation laws (1.2), (1.3). For instance, the approximating sequence defined by the standard Galerkin method (2.3) is such that

$$E_1^{n,m+1} = E_1^{n,m}, \qquad m = 0, 1, ..., M-1, \tag{3.3}$$

$$E_2^{n,m+1} = E_2^{n,m} - \tfrac{1}{8} \int_I |u_n^{m+1} - u_n^m|^2 (|u_n^{m+1}|^2 - |u_n^m|^2)\, dx, \qquad m = 0, ..., M-1. \tag{3.4}$$

Equation (3.3) is achieved by setting $\phi = u_n^{m+1} + u_n^m$ in (2.3) and taking the imaginary part; (3.4) is achieved by setting $\phi = u_n^{m+1} - u_n^m$ in (2.3) and taking the real part.

We can ascribe the fact that the solution of (2.3) fails to satisfy the discrete analog of (1.3) to the use of the Crank–Nicolson scheme. It can easily be seen that the solution of (2.5) satisfies neither of the analogs of (1.2) and (1.3). This is inherent in the use of product approximation. However, as the above lemma indicates, the absence of discrete analogs of the conservation laws does not rule out spatial convergence.

Besides, the difficulty of satisfying discrete conservation laws like (3.3) in practice while dealing with nonlinear systems of algebraic equations has long been recognized [2]. As we shall demonstrate in our first numerical experiment, it is not always advisable to alter a numerical method for the sake of enforcing a conservation law.

Another point of interest is that we have no information about the validity of our lemma for the method (2.5) when the subspaces do not consist of piecewise linear functions. It will be the purpose of our second numerical experiment to investigate whether or not piecewise linear functions represent the most efficient choice.

4. First Numerical Experiment

In this section, we use the standard Galerkin method with piecewise linear functions for the numerical solution of the CSE. Let

$$\underline{u}_n(t) = \sum_{j=1}^{\nu} \underline{\alpha}_j(t)\,\phi_j, \qquad \text{where} \quad \underline{\alpha}_j(t) = \begin{bmatrix} \operatorname{Re} \alpha_j(t) \\ \operatorname{Im} \alpha_j(t) \end{bmatrix}$$

$$\underline{\gamma}(\underline{u}_n) = (\underline{u}_n^T \underline{u}_n)\, A\underline{u}_n, \qquad \text{where} \quad A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

With $(\underline{\alpha})_j = \underline{\alpha}_j$, (2.2) assumes the matrix form

$$M\dot{\underline{\alpha}} + S\underline{\alpha} + N(\underline{\alpha}) = 0. \tag{4.1}$$

In this expression, M and S are the obvious mass and stiffness matrices of order ν having matrices of order 2 as elements, and

$$(N(\underline{\alpha}))_j = \langle \underline{\gamma}(\underline{u}_n), \phi_j \rangle.$$

For the solution of (2.3), we adopt the following predictor–corrector pair

$$M\underline{\alpha}^* = M\underline{\alpha}^{\mathrm{m}} - \Delta t(S\underline{\alpha}^{\mathrm{m}} + N(\underline{\alpha}^{\mathrm{m}}))$$

$$\left(M + \frac{\Delta t}{2} S\right) \underline{\alpha}^{\mathrm{m}+1} = \left(M - \frac{\Delta t}{2} S\right) \underline{\alpha}^{\mathrm{m}} - \Delta t N\left(\frac{\underline{\alpha}^* + \underline{\alpha}^{\mathrm{m}}}{2}\right).$$

The initial vector $\underline{\alpha}^0$ is obtained readily from the interpolant of the initial condition $f(x)$.

For piecewise linear functions, M and S are clearly tridiagonal matrices of order $\nu = n = (b-a)/h - 1$. This scheme requires that M and $M + (\Delta t/2)\,S$ be factorized. This need be done only once and the LU decomposition can be kept in storage. At each time step, 2 backward–forward solves must be performed. For piecewise linear functions, M and S are clearly tridiagonal matrices of order n. Thus, each factorization requires $18n$ operations (an operation being either a division or a multiplication) while the forward–backward solution requires $12n$ operations. All in all, the algorithm involves $(36 + 24(T/\Delta t))\,n$ operations. For our numerical experiment, we choose as the initial condition

$$f(x) = \sqrt{2}\left\{e^{ic_1(x/2)} \operatorname{sech}\left(\frac{x}{\sqrt{2}}\right) + e^{ic_2((x-25)/2)} \operatorname{sech}\left(\frac{x-25}{\sqrt{2}}\right)\right\}$$

$$\text{with} \quad c_1 = 10, \qquad c_2 = \tfrac{1}{10}, \text{ and } i = \sqrt{-1}.$$

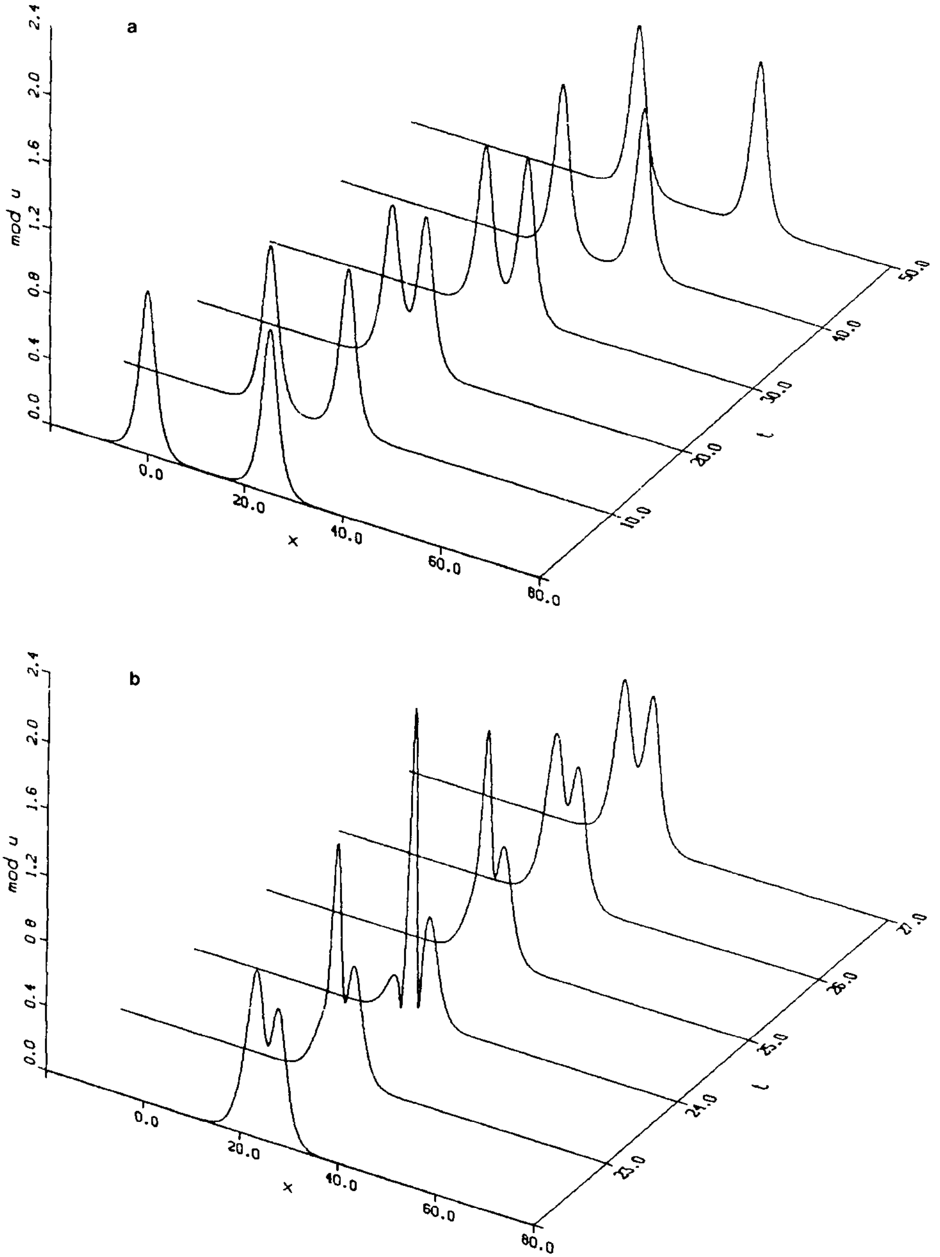

FIG 1 Standard Galerkin solution Predictor–corrector (1 iteration), $h = \frac{1}{5}$, $\tau = \frac{1}{10}$ (a) modulus from 0 to 50 s, (b) modulus from 22 to 27 s, (c) the two quantities in time

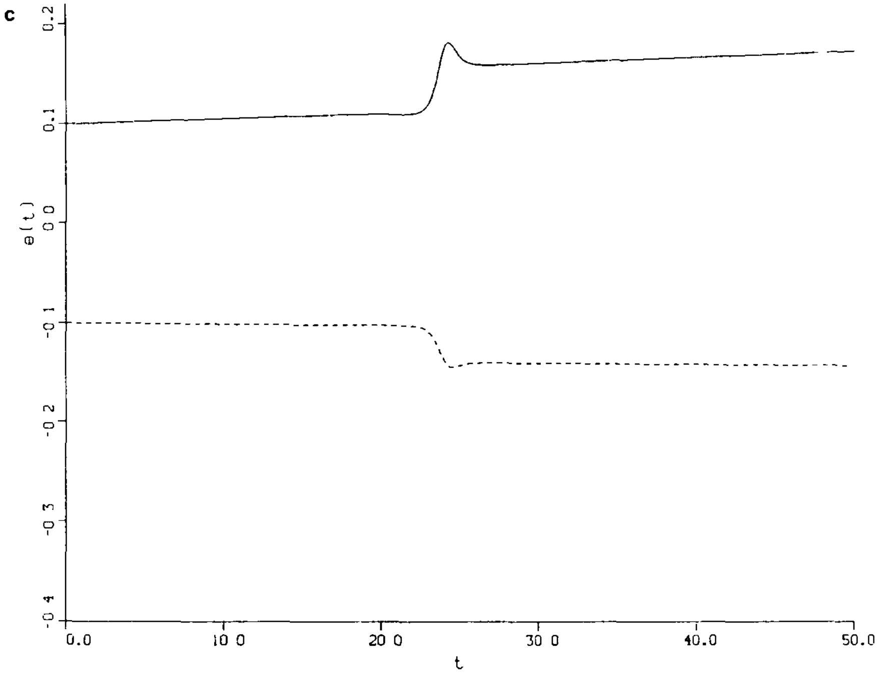

FIGURE 1—*Continued*

This initial condition has the following interpretation: two wave forms (or solitons) are separated by a distance of 25 units. As time progresses, the faster soliton (velocity c_1) eventually catches up with the slower one (velocity c_2) and, according to soliton theory [8], passes through it with only a phase shift resulting from the collision.

To allow sufficient room for the interaction to take place, we choose $]-20, 80[$ as the space interval and compute the solution for $0 \leqslant t \leqslant 50$. We implemented our scheme and ran the program with $h = \frac{1}{5}$ and $\Delta t = \frac{1}{10}$ using one iteration of the corrector. The results are depicted in Figs. 1a, b, and c. They are in qualitative agreement with the behaviour predicted by the theory. The two wave forms collide but recover their shapes afterwards (Fig. 1a) despite a strongly nonlinear interaction (Fig. 1b). The evolution of the two quantities $E_1^{n,m}$ and $E_2^{n,m}$ in time is given by Fig. 1c. The first quantity (full line) and the second quantity (broken line) have been shifted to $\frac{1}{10}$ and $\frac{-1}{10}$, respectively, for convenience. Interestingly enough, we observe the following behaviour: the first quantity grows when the second decays and conversely. The two quantities evolve linearly before and after the interaction but experience a jump during the interaction. Figure 2 (a, b, and c) depicts the results obtained with two iterations of the corrector. The additional iteration has a significant effect on the phase of the solution and also on the general behaviour of the two quantities.

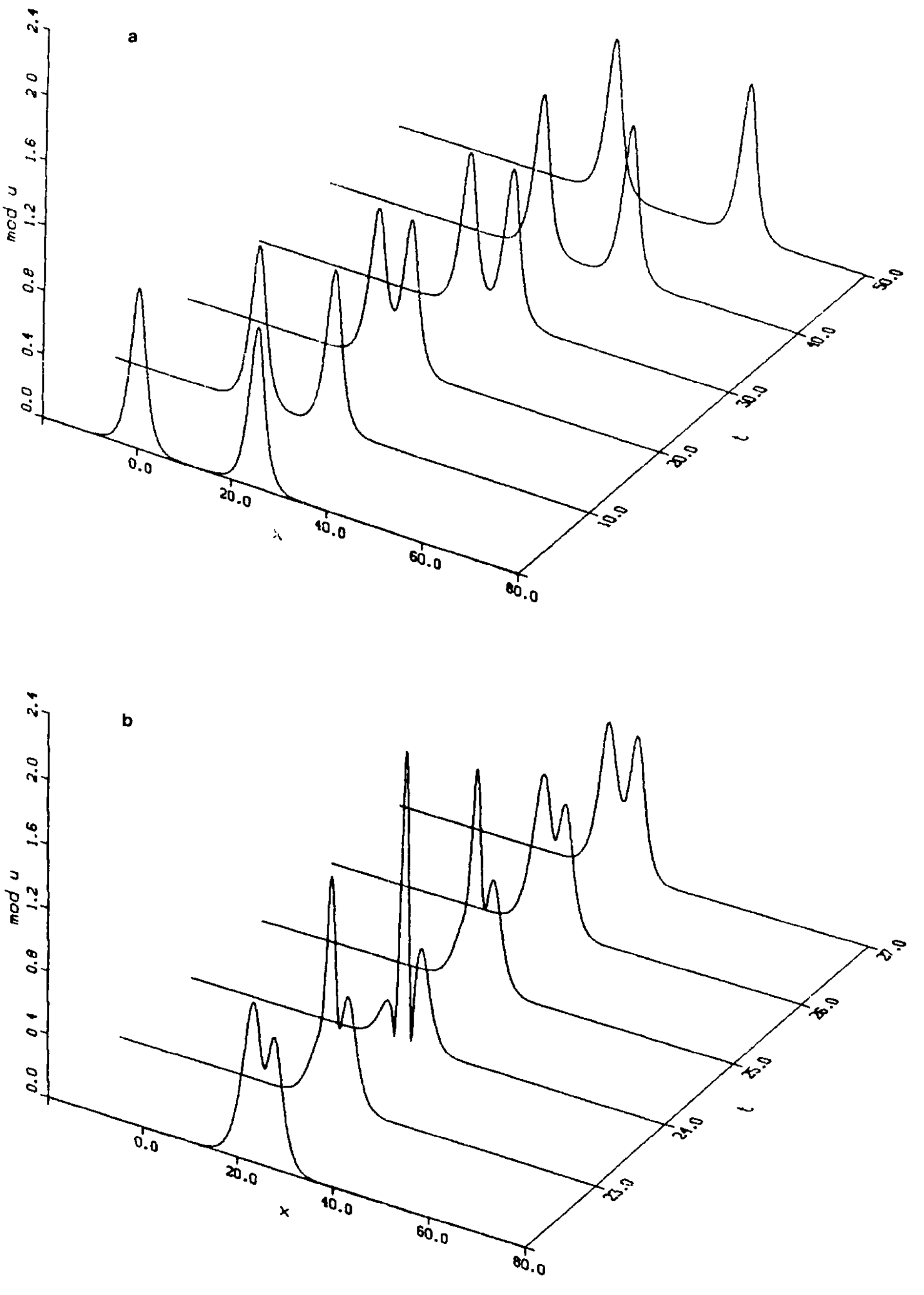

FIG 2 Standard Galerkin solution Predictor–corrector (2 iterations), $h=\frac{1}{5}$, $\tau=\frac{1}{10}$ (a) modulus from 0 to 50 s, (b) modulus from 22 to 27 s, (c) the two quantities in time.

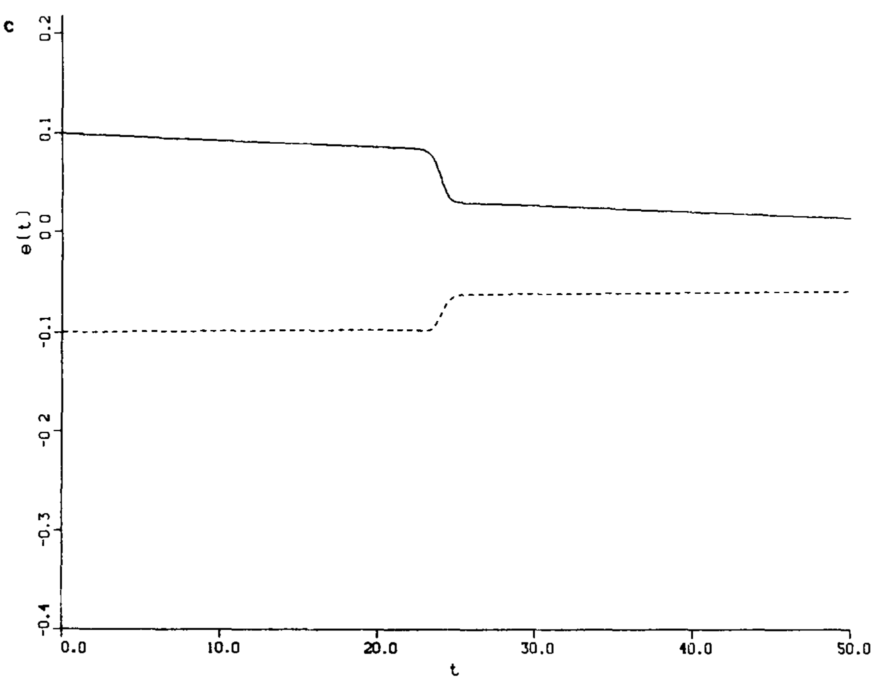

FIGURE 2—*Continued*

In view of the fact that the approximation scheme (2.3) is such that $E_1^{n,m+1} = E_1^{n,m}$, the fluctuations in the first quantity as shown in Figs. 1c and 2c point out the difficulty of reproducing the conservation laws in practice while solving nonlinear systems. This seems to have stimulated the search for methods which enforce conservation properties. For instance, Herbst *et al.* [4] implemented a method with a variable time step able to preserve the first quantity exactly at each time step. However, the method was found to be of little practical use because nothing prevents the time step from decreasing to zero or, even worse, assuming negative values.

In the remainder of this section, we present a method of Newton type which will preserve the first quantity exactly at each time step. We should therefore be in a position to decide whether such a special scheme presents some computational advantage.

Suppose we decide to solve (2.3) by means of Newton's iteration. The function to iterate is

$$F(\underline{\alpha}^{\mathrm{m}}, \underline{z}) = \left(M + \frac{\Delta t}{2} S\right) \underline{z} - \left(M - \frac{\Delta t}{2} S\right) \underline{\alpha}^{\mathrm{m}} + \Delta t N\left(\frac{\underline{\alpha}^{\mathrm{m}} + \underline{z}}{2}\right),$$

where $\underline{\alpha}^{\mathrm{m}}$ is kept fixed during the iteration process. Let $J(\underline{\alpha}^{\mathrm{m}}, \underline{z})$ be the jacobian

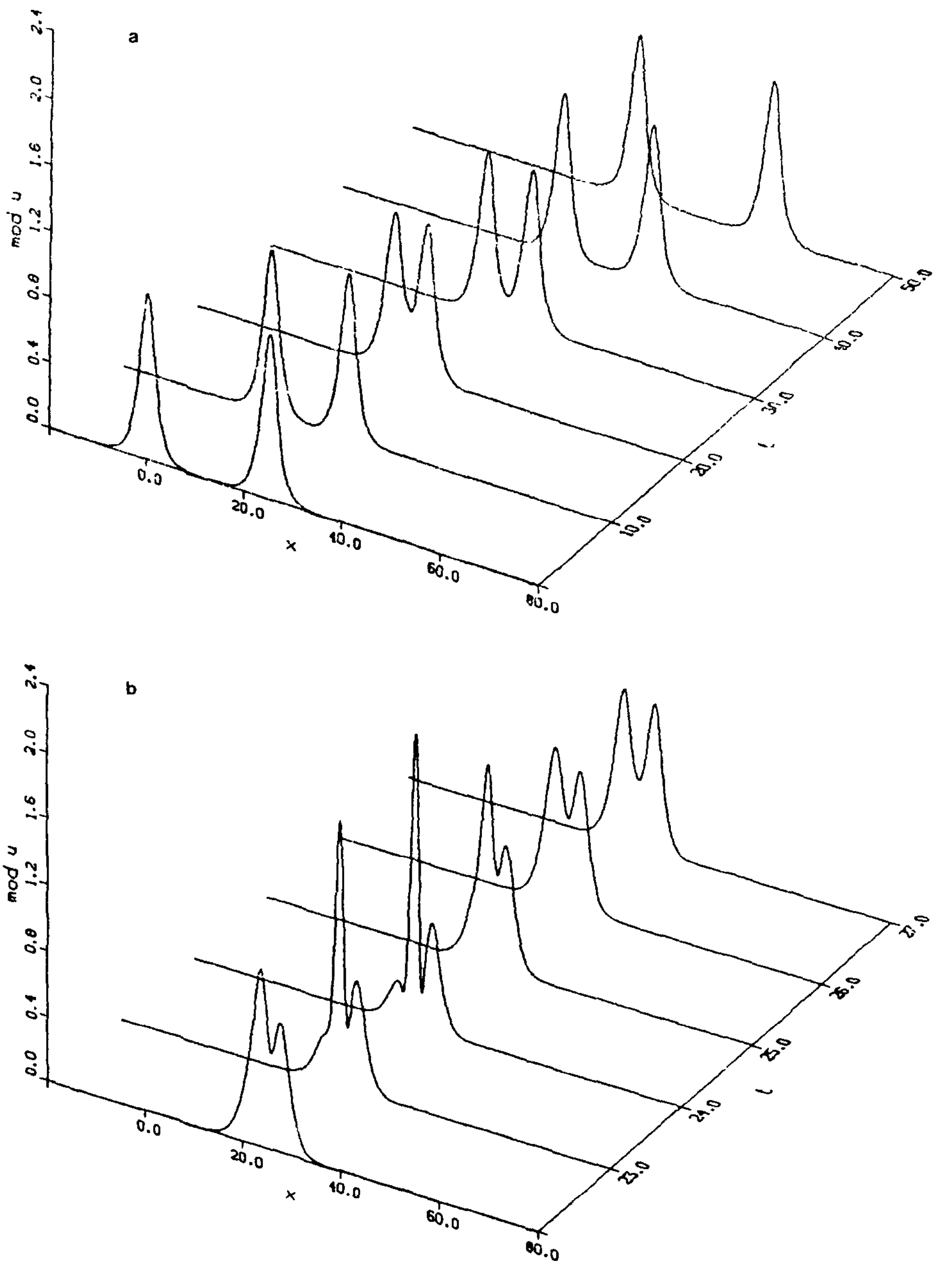

FIG 3 Standard Galerkin solution Modified Newton method, $h = \frac{1}{5}$, $\tau = \frac{1}{10}$. (a) modulus from 0 to 50 s; (b) modulus from 22 to 27 s, (c) the two quantities in time

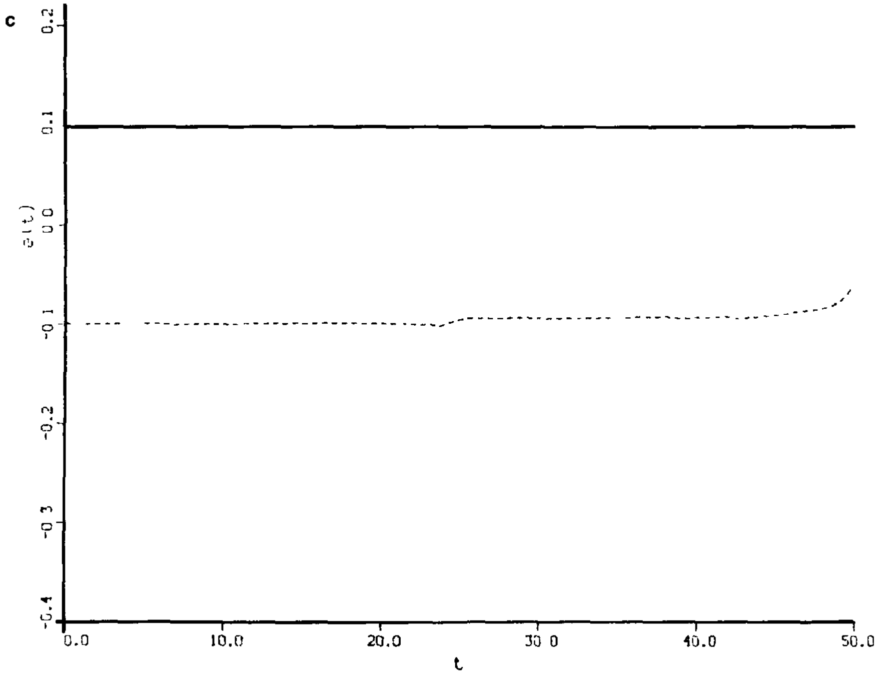

FIGURE 3—*Continued*

matrix of $F(\underline{\alpha}^{\mathrm{m}}, \underline{z})$. Choosing $\underline{\alpha}^{\mathrm{m}} = \underline{z}_0$ as a starting value, the iteration sequence is generated through

$$\underline{z}_{\mathrm{k}+1} = \underline{z}_{\mathrm{k}}(J(\underline{\alpha}^{\mathrm{m}}, \underline{z}_{\mathrm{k}}))^{-1} F(\underline{\alpha}^{\mathrm{m}}, \underline{z}_{\mathrm{k}}).$$

Noting that the first quantity $E_1^{n,m}$ takes the algebraic form

$$E_1^{n,m} = (\underline{\alpha}^{\mathrm{m}})^{\mathrm{T}} M \underline{\alpha}^{\mathrm{m}}$$

We will not have $\underline{z}_k^{\mathrm{T}} M \underline{z}_{\mathrm{k}} = \underline{z}_{k+1}^{\mathrm{T}} M \underline{z}_{\mathrm{k}+1}$ for each k, but only as $k \to \infty$. Therefore, we adopt a modified Newton method by introducing a matrix P and consider the sequence defined by

$$\underline{z}_{\mathrm{k}+1} - \underline{z}_{\mathrm{k}} = -P(J(\underline{\alpha}^{\mathrm{m}}, \underline{z}_{\mathrm{k}}))^{-1} F(\underline{\alpha}^{\mathrm{m}}, \underline{z}_{\mathrm{k}}).$$

Premultiply by M

$$M(\underline{z}_{\mathrm{k}+1} - \underline{z}_{\mathrm{k}}) = -MP(J(\underline{\alpha}^{\mathrm{m}}, \underline{z}_{\mathrm{k}}))^{-1} F(\underline{\alpha}^{\mathrm{m}}, \underline{z}_{\mathrm{k}}).$$

Then by $(\underline{z}_{\mathrm{k}+1} + \underline{z}_{\mathrm{k}})^{\mathrm{T}}$. Noting that M is symmetric, this yields

$$\underline{z}_{k+1}^{\mathrm{T}} M \underline{z}_{\mathrm{k}+1} - \underline{z}_k^{\mathrm{T}} M \underline{z}_{\mathrm{k}} = -(\underline{z}_{\mathrm{k}+1} + \underline{z}_{\mathrm{k}})^{\mathrm{T}} MP(J(\underline{\alpha}^{\mathrm{m}}, \underline{z}_{\mathrm{k}}))^{-1} F(\underline{\alpha}^{\mathrm{m}}, \underline{z}_{\mathrm{k}})$$

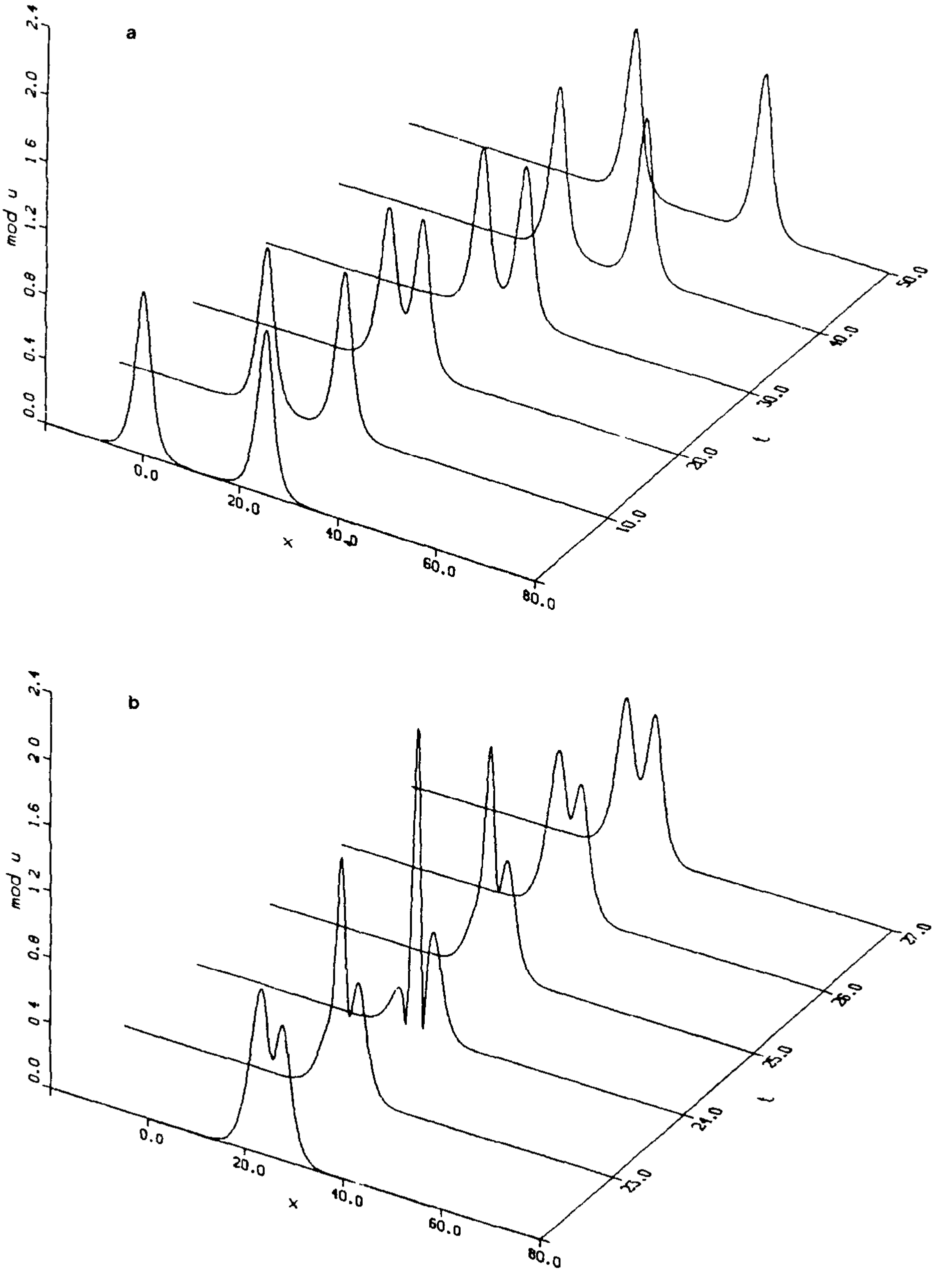

FIG 4. Standard Galerkin solution Newton Method, $h=\frac{1}{5}$, $\tau=\frac{1}{10}$ (a) modulus from 0 to 50 s, (b) modulus from 22 to 27 s, (c) the two quantities in time

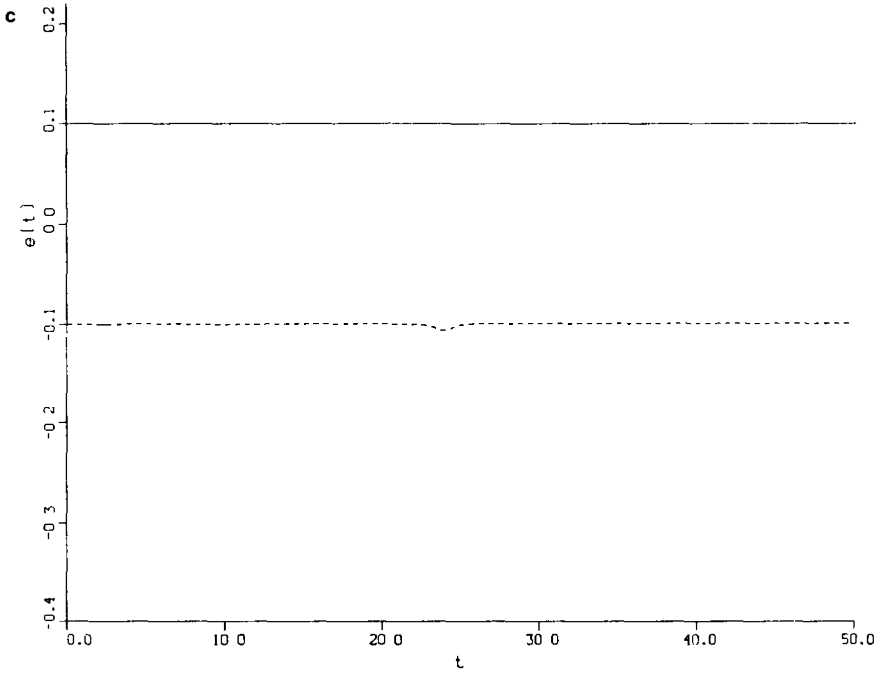

FIGURE 4—*Continued*

We require that the right-hand side be zero. Substituting for $(\underline{z}_{k+1}+\underline{z}_k)$ gives

$$[2\underline{z}_k - P(J(\underline{a}^m, \underline{z}_k))^{-1}F(\underline{a}^m, \underline{z}_k)]^T MP(J(\underline{a}^m, \underline{z}_k))^{-1}F(\underline{a}^m, \underline{z}_k) = 0$$

and this provides a condition on the matrix P. Some choices of P are trivial and we require that the entire expression vanish, not only part of it. For instance, assume $P = \operatorname{diag}(\pi)$, where π is a scalar. Then, letting

$$(J(\underline{a}^m, \underline{z}_k))^{-1}F(\underline{a}^m, \underline{z}_k) = \underline{\xi}_k$$

We see that

$$\pi = 2\,\frac{\underline{z}_k^T M \underline{\xi}_k}{\underline{\xi}_k^T M \underline{\xi}_k}.$$

Using the same initial condition, we ran the program with $h = \frac{1}{5}$ and $\Delta t = \frac{1}{10}$. The iteration process was to be terminated whenever

$$\|\underline{z}_{k+1} - \underline{z}_k\| < \varepsilon = 10^{-4}.$$

The results (Fig. 3a, b, and c) show that the first quantity was indeed exactly conserved. Typically, the first iteration was performed with $\pi \approx 1$ ($\pi = 1$ is the

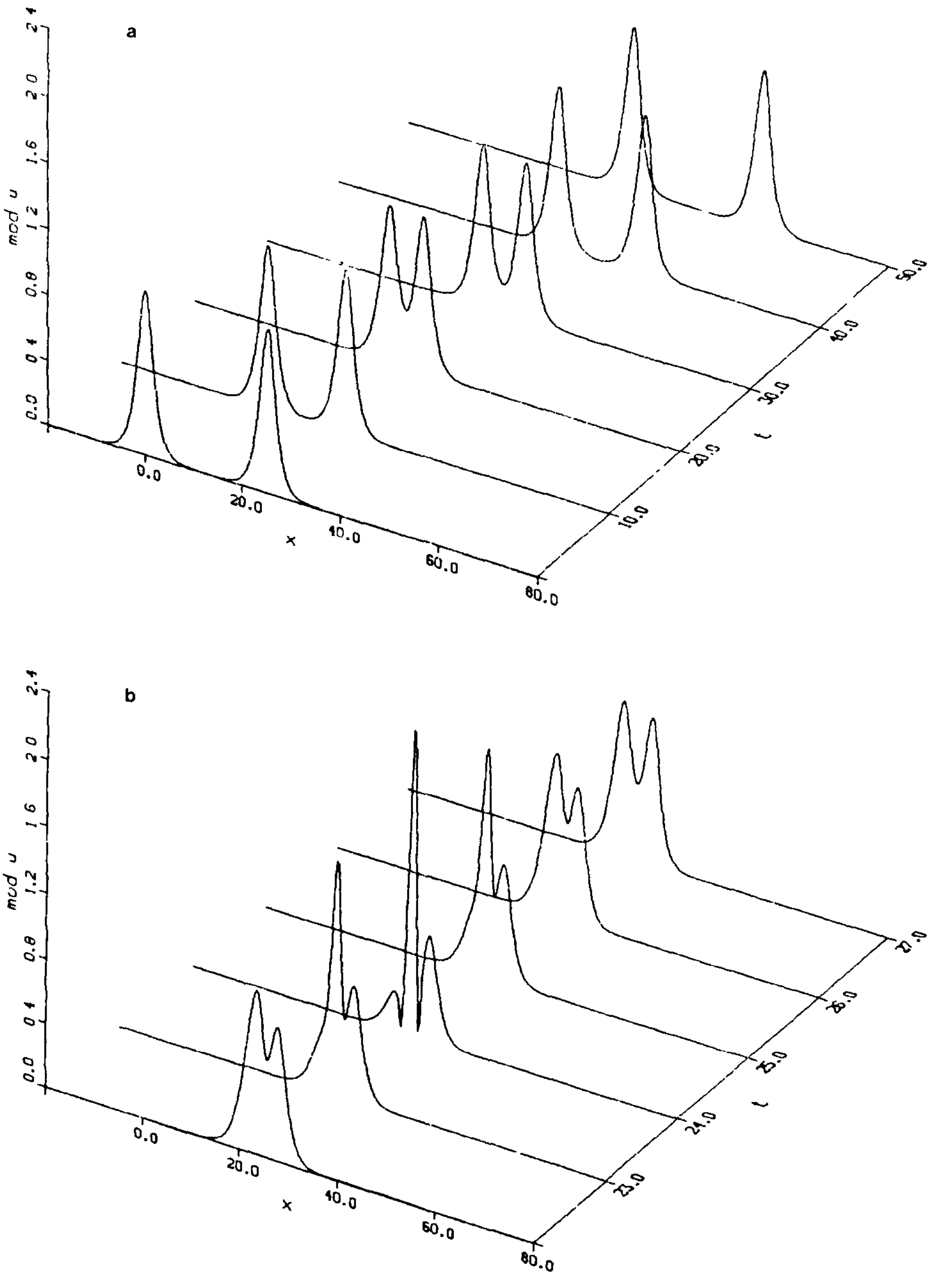

FIG. 5. Standard Galerkin solution Newton Method, $h=\frac{1}{5}$, $\tau=\frac{1}{16}$ (a) modulus from 0 to 50 s, (b) modulus from 22 to 27 s, (c) the two quantities in time

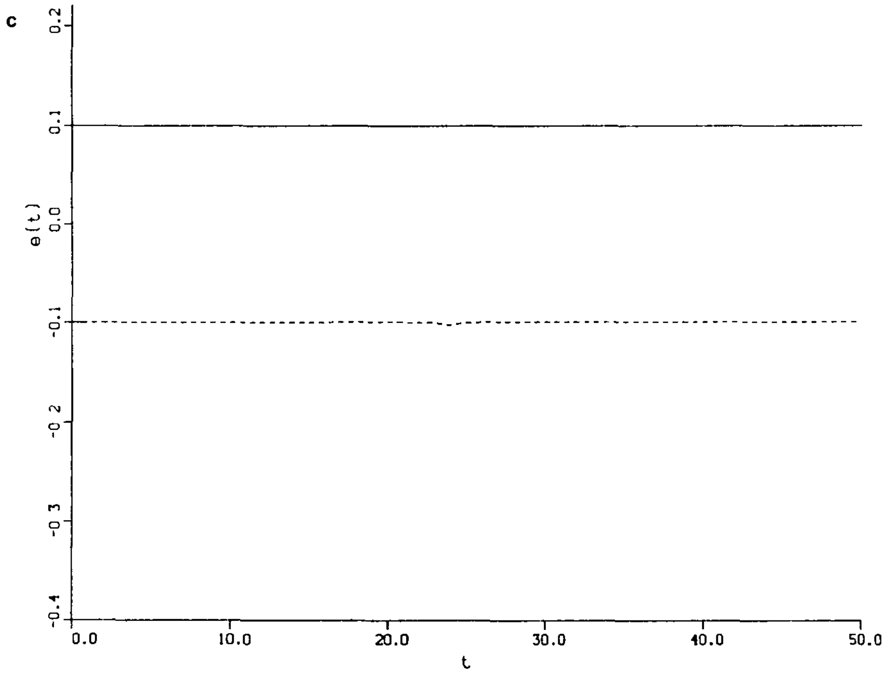

FIGURE 5—*Continued*

standard Newton method). After that, π would vary consistently between 10^{-2} and 10^{2}. During the interaction, up to 3 iterations were needed but, otherwise, the tolerance was met in two iterations. While the first quantity is exactly conserved, we notice that the second quantity fluctuates a little. These variations tend to grow in the last seconds (Fig. 3c).

For the sake of comparison, we also ran the standard Newton method (Fig. 4a, b, c). The two quantities are very well behaved (Fig. 4c: the first quantity is (very nearly) conserved at each time step and the second quantity does not experience a sudden growth in the last seconds. A reduction to $\Delta t = \frac{1}{16}$ indicates that the standard Newton method produces the best solution (Fig. 5a, b, c).

From these facts, we conclude that our modified Newton method presents little advantage upon the standard one. In fact, the bahaviour of the second quantity should serve as a warnıng: a method designed to preserve the first quantıty exactly may cause perturbations in the second quantıty. Our lemma of Section 3 shows that the two quantities are equally important, therefore such schemes could perform poorly.

5. Second Numerical Experiment

In this section, we examine the numerical performance of the modified version of the Galerkin method using product approximation. We use the same notation as before.

Product approximation consists of replacing $\underline{\gamma}$ by its interpolant. Thus, when piecewise linear functions are used, this interpolant is simply

$$\sum_{j=1}^{n} \underline{\gamma}(\underline{\alpha}_j(t))\,\phi_j$$

and (2.4) assumes the matrix form

$$M\dot{\underline{\alpha}} + S\underline{\alpha} + MF(\underline{\alpha}) = 0 \tag{5.1}$$

with $(F(\underline{\alpha}))_j = (\underline{\alpha}_j^{\mathrm{T}}\underline{\alpha}_j)A\underline{\alpha}_j$.

We note that the nonlinear term, in its simplicity, does not involve the tedious manipulations of the standard Galerkin method. We use the same predictor–corrector pair as before for the solution of (2.5). With one iteration of the corrector, the integration in time will again require $(36+24(T/\Delta t))\,n$ operations.

A pleasant feature of the product approximation version of the Galerkin method is that the use of other polynomial splines becomes attractive: (5.1) allows the programmer to implement the method for basis functions which have a larger support. It is natural to investigate whether more efficiency could be obtained through an alternative choice of test and trial functions. When the method uses Hermite cubics, the finite dimensional subspace is[1]

$$S_n = [R_1, T_1, ..., R_n, T_n].$$

The Galerkin solution takes the form

$$\underline{u}_n(t) = \sum_{j=1}^{n} (\underline{\alpha}_j(t)\,R_j + \underline{\beta}_j(t)\,T_j)$$

and product approximation will lead to the system of ordinary differential equations

$$M\dot{\underline{\xi}} + S\underline{\xi} + M\underline{\Psi}(\underline{\xi}) = 0,$$

where

$$(\underline{\xi})_j = \begin{bmatrix} \underline{\alpha}_j \\ \underline{\beta}_j \end{bmatrix} \quad \text{and} \quad (\underline{\Psi}(\underline{\xi}))_j = \begin{bmatrix} (\underline{\alpha}_j^{\mathrm{T}}\underline{\alpha}_j)\,A\underline{\alpha}_j \\ B(\underline{\alpha}_j)\,\underline{\beta}_j \end{bmatrix},$$

[1] It is understood that the basis functions at the extremities of the interval $]a, b[$ must be such as to satisfy the boundary condition It is clear how this is achieved from the definition of a typical basis function as given in Section 2

where B is the Jacobian matrix of $\underline{\gamma}$. M and S may be viewed as *scalar* matrices of order $4n$. They are band matrices with seven lower codiagonals and seven upper codiagonals. We find that the factorization of such a matrix requires $224n$ operations. The forward–backward solution is performed in $60n$ operations. Thus, the number of operations involved in the time integration process is

$$\left(448+120\frac{T}{\Delta t}\right)n=\left(448+120\frac{T}{\Delta t}\right)\left(\frac{b-a}{h}-1\right).$$

Let us now turn to cubic splines. Because the interpolation formula associated with this choice of basis functions involves the solution of a linear system of equations, the technique of product approximation cannot be used. However, this difficulty is avoided by using piecewise linear functions as basis functions and cubic splines as test functions. This is an instance of a Petrov–Galerkin method where test and trial functions do not belong to the same piecewise polynomial space. For the test functions, we wish to have a basis with only n elements. So we take[2]

$$T_n=[B_3, ..., B_{n+2}].$$

This Petrov–Galerkin method may be written

$$\sum_{j=1}^{n} \dot{\underline{\alpha}}_j(t)\langle\phi_j, B_k\rangle - A\sum_{j=1}^{n} \underline{\alpha}_j(t)\langle\phi_j', B_k'\rangle + \sum_{j=1}^{n} \underline{\gamma}(\underline{\alpha}_j(t))\langle\phi_j, B_k\rangle = 0$$

for $k=3, ..., n-2$. In matrix form

$$M\dot{\underline{\alpha}}+S\underline{\alpha}+MF(\underline{\alpha})=0,$$

TABLE I

Number of Operations and Efficiency Condition

Method	Number of operations	Efficiency condition
(l–l)	$\left(36+24\frac{T}{\Delta t}\right)\left(\frac{b-a}{h}-1\right)$	—
(h–h)	$\left(448+120\frac{T}{\Delta t}\right)\left(\frac{b-a}{h}-1\right)$	$h_h>5h_l$
(l–c)	$\left(100+40\frac{T}{\Delta t}\right)\left(\frac{b-a}{h}-1\right)$	$h_c>\frac{5}{3}h_l$

[2] There is a slightly modified version of our lemma of Section 3 for this Petrov–Galerkin method The set $\{u^0, \ , u^M\}$ satisfies (3 1) only for all ϕ in the closure of $\bigcup_{n=1} T_n$ in H_0^1 This is simply due to the fact that $\bigcup_{n=1}^{\infty} T_n$ is not dense in H_0^1 [6]

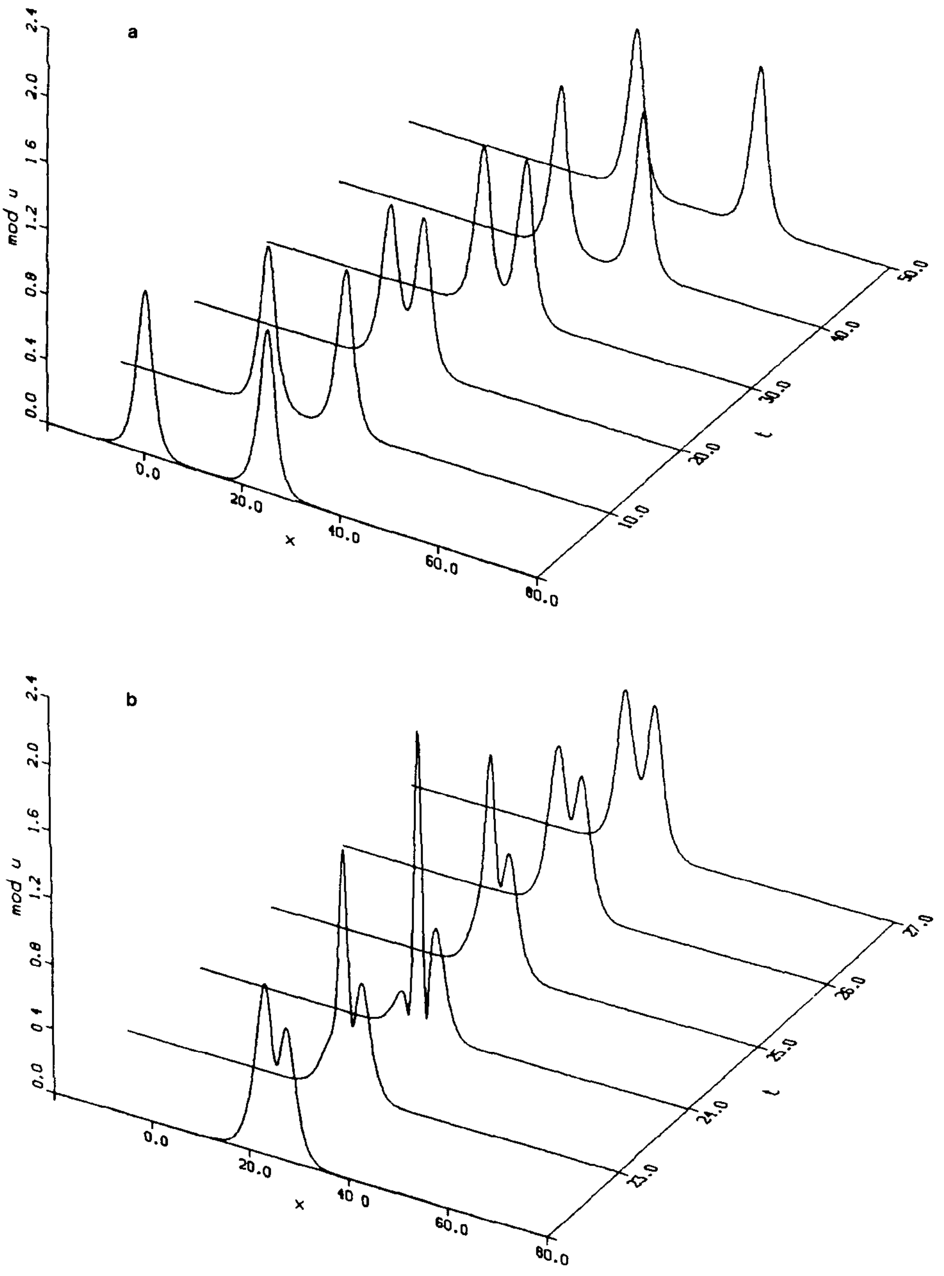

FIG. 6. Galerkin solution with product approximation Predictor–corrector (1 iteration), $h=\frac{1}{4}$, $\tau=\frac{1}{10}$ (a) modulus from 0 to 50 s; (b) modulus from 22 to 27 s, (c) the two quantities in time.

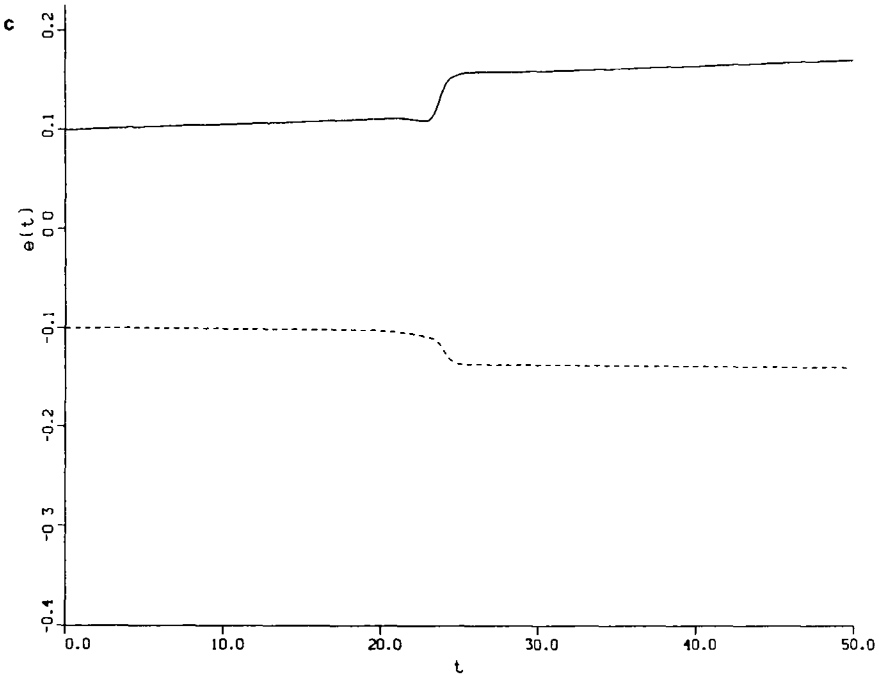

FIGURE 6—*Continued*

where M and S are now quindiagonal with blocks of order 2. The factorization requires $50n$ operations; the backward–forward solution requires $20n$ operations. The number of operations required for the time integration is therefore

$$\left(100+40\frac{T}{\Delta t}\right)\left(\frac{b-a}{h}-1\right).$$

To each particular choice of basis functions, we have attached a number of operations required for the integration in time. This number will provide a quan-

TABLE II

Results of the Experiment

Method	Efficiency condition	Optimal grid size	Efficiency
(l–l)	—	$\frac{1}{4}$	—
(h–h)	$h_h > \frac{5}{4}$	$\frac{1}{3}$	Worse
(l–c)	$h_c > \frac{5}{12}$	$\frac{1}{2}$	Better

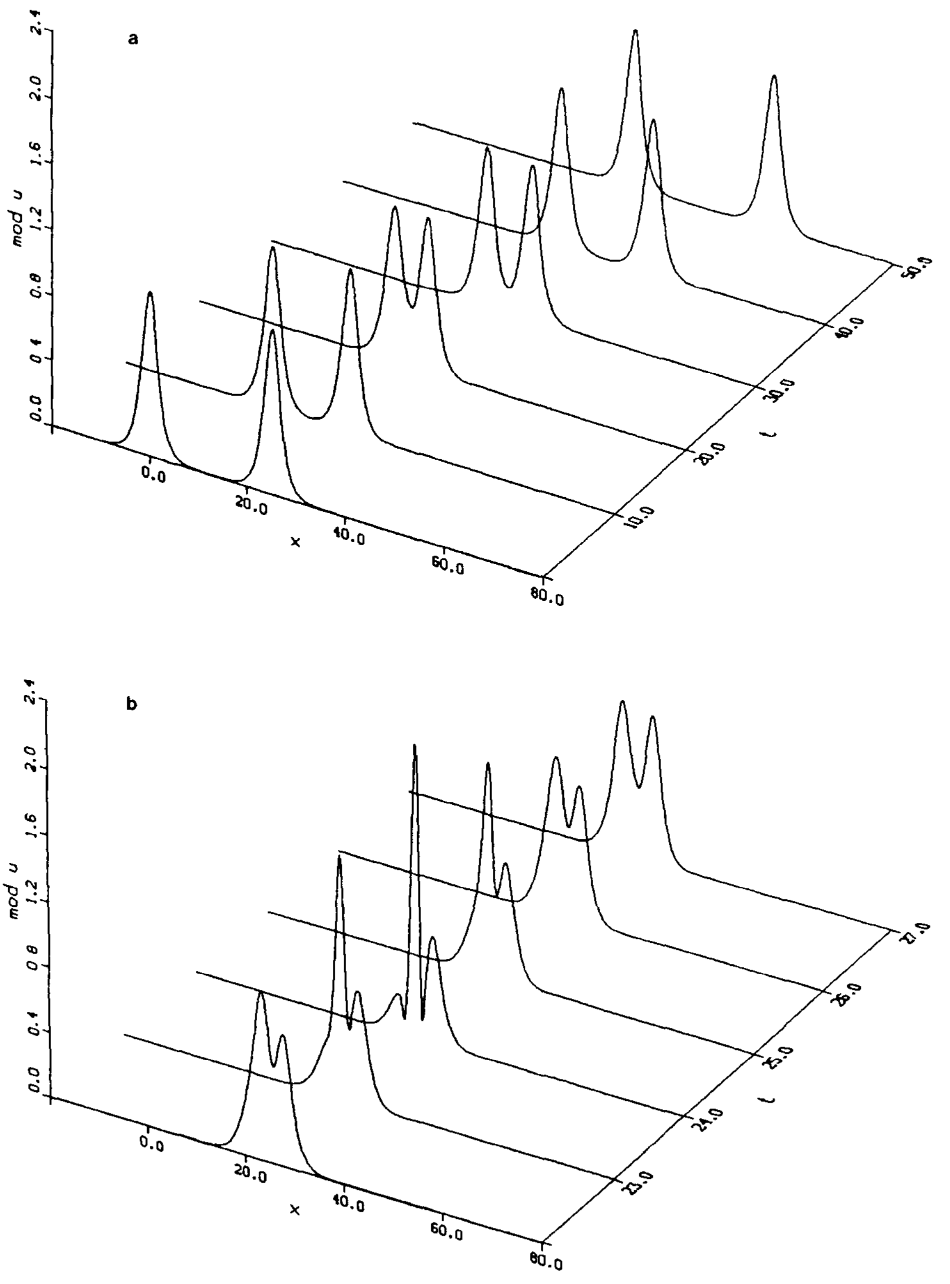

Fig 7. Galerkin solution with product approximation. Predictor–corrector (2 iterations), $h=\frac{1}{4}$, $\tau=\frac{1}{10}$: (a) modulus from 0 to 50 s, (b) modulus from 22 to 27 s; (c) the two quantities in time

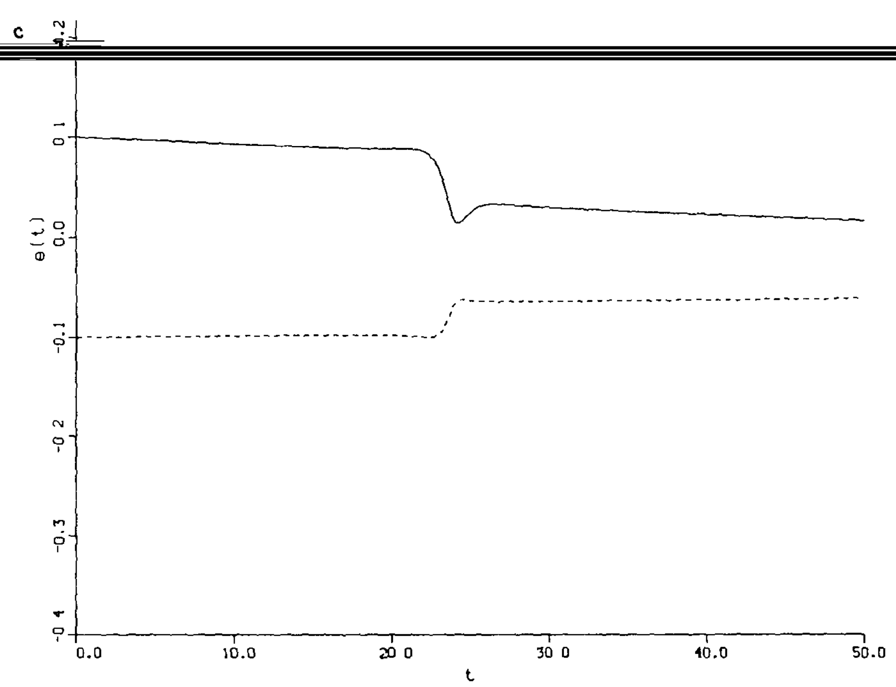

FIGURE 7—*Continued*

titative criterion upon which to base a comparison and judge the relative performance of the schemes. In order to avoid confusion, we make the following pseudo-definitions:

Given a fixed value of Δt, a method will be said to *converge in space* for $h = h(\Delta t)$ if a further reduction of h has no perceptible effect on the plots of the numerical solution. $h(\Delta t)$ is then said to be the *optimal grid size*. For fixed T, Δt, and interval $I =]a, b[$, the number of operations is a function of h only. We shall say that the *most efficient method* is the one for which convergence in space occurs for the least number of operations. Take, for instance, the Galerkin method with product approximation and piecewise linear functions as test and trial functions. For the sake of brevity, it will be denoted (1–1). The number of operations was found to be

$$\left(36 + 24\frac{T}{\Delta t}\right)\left(\frac{b-a}{h} - 1\right).$$

Now, take for example the (h–h) method (i.e., Hermite cubics as test and trial function). There, the number of operations involved was

$$\left(448 + 120\frac{T}{\Delta t}\right)\left(\frac{b-a}{h} - 1\right).$$

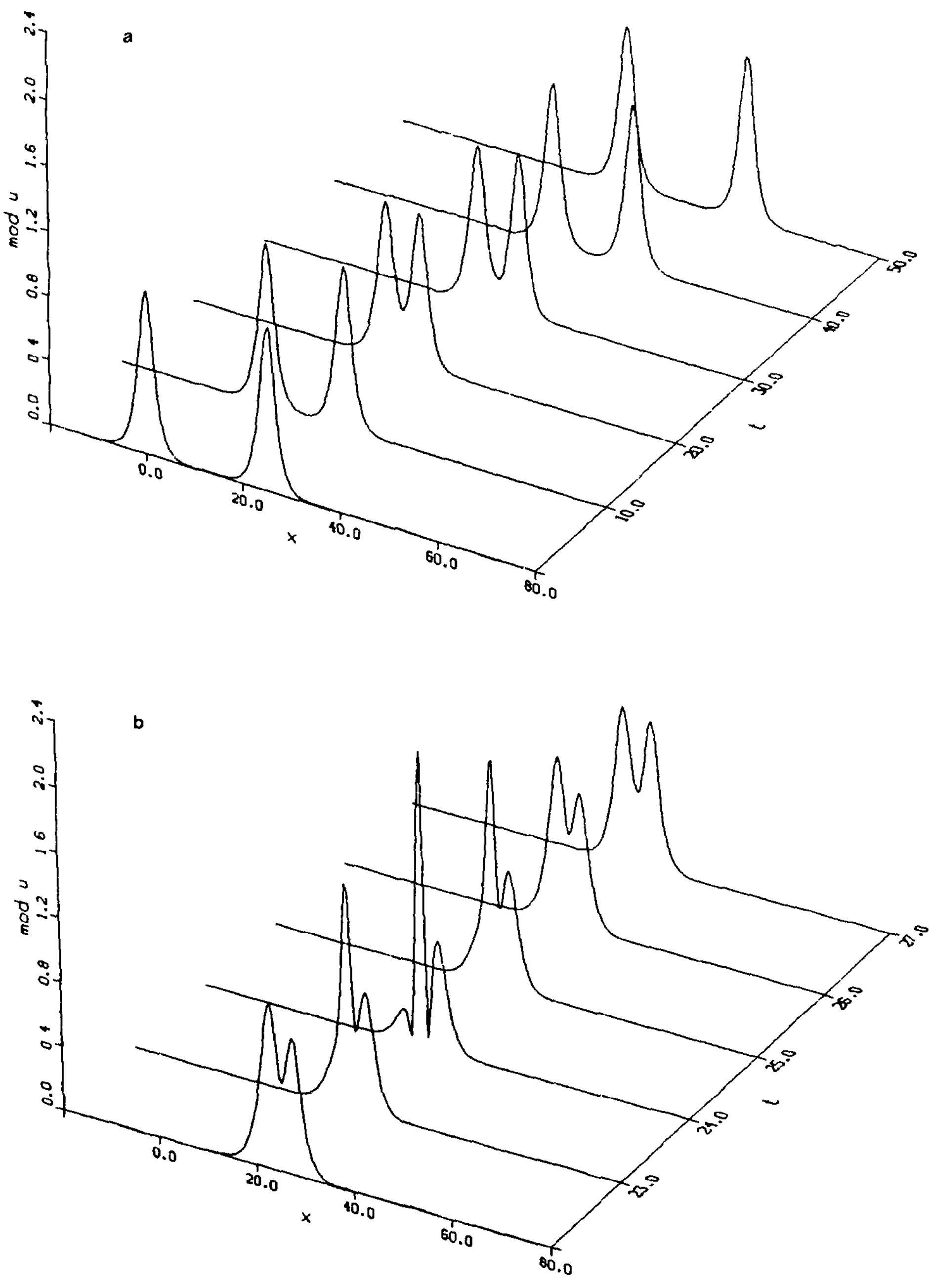

FIG 8. Petrov–Galerkin solution with product approximation Predictor–corrector (1 iteration), $h=\frac{1}{2}$, $\tau=\frac{1}{10}$: (a) modulus from 0 to 50 s, (b) modulus from 22 to 27 s, (c) the two quantities in time

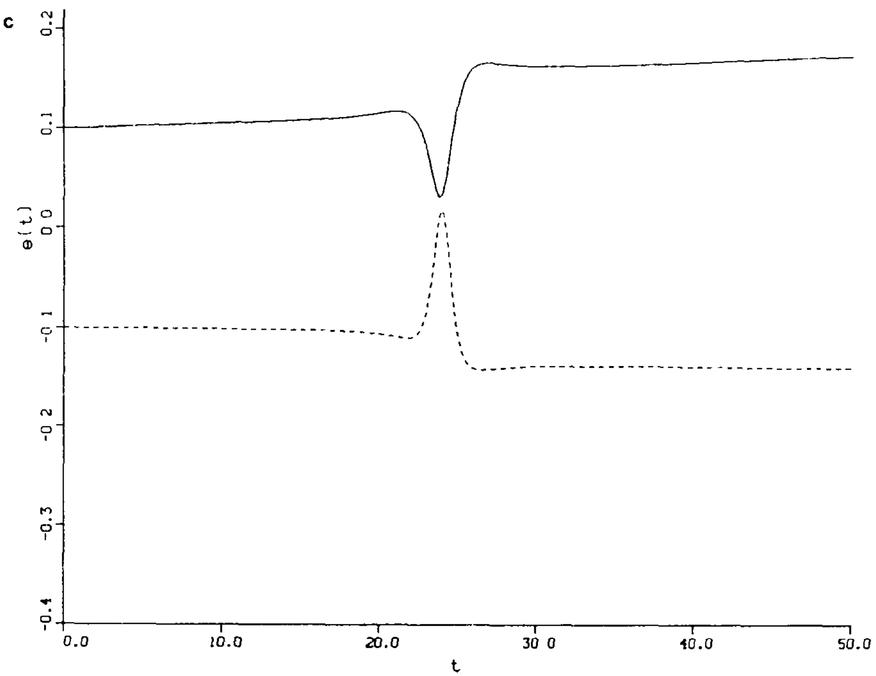

FIGURE 8—*Continued*

Thus, if h_l and h_h are the optimal grid size for (l–l) and (h–h), respectively, the (h–h) method will be more efficient only if

$$\left(448+120\frac{T}{\Delta t}\right)\left(\frac{b-a}{h}-1\right)<\left(36+24\frac{T}{\Delta t}\right)\left(\frac{b-a}{h}-1\right).$$

That is

$$\frac{h_l(b-a-h_h)}{h_h(b-a-h_l)}<\frac{36\,\Delta t+24T}{448\,\Delta t+120T},$$

but since h and Δt are small quantities

$$\frac{h_l}{h_h}<\frac{1}{5}.$$

In other words, the extra complexity will be worth our while only if we can use a grid 5 times coarser than for the piecewise linear functions. We do the same calculation for the (l–c) method and obtain Table I, where the last column tabulates the condition that the optimal grid sizes must fulfill in order for the methods to be more efficient than (l–l).

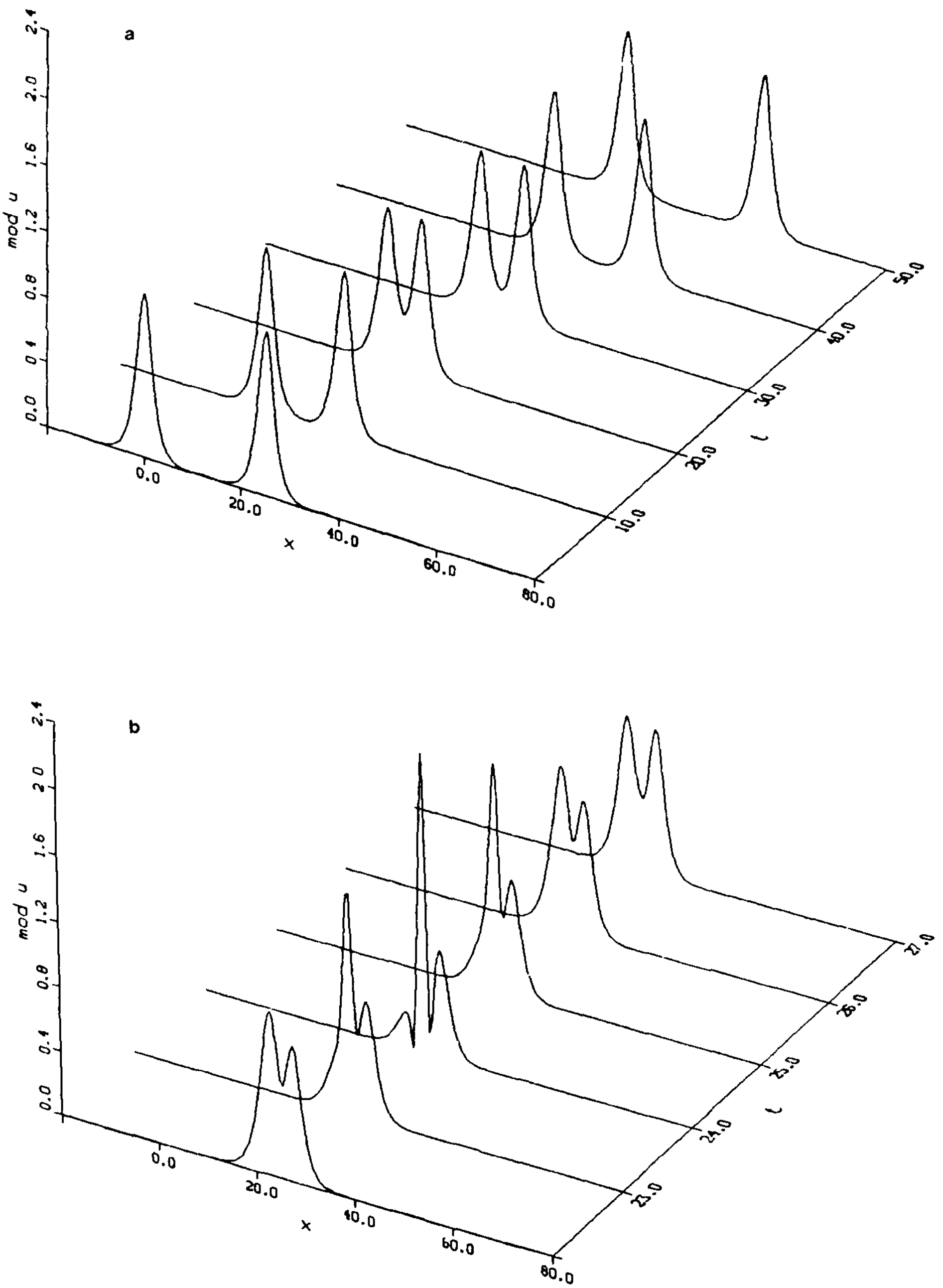

FIG 9. Petrov–Galerkin solution with product approximation Predictor–corrector (2 iterations), $h=\frac{1}{2}$, $\tau=\frac{1}{10}$: (a) modulus from 0 to 50 s, (b) modulus from 22 to 27 s, (c) the two quantities in time.

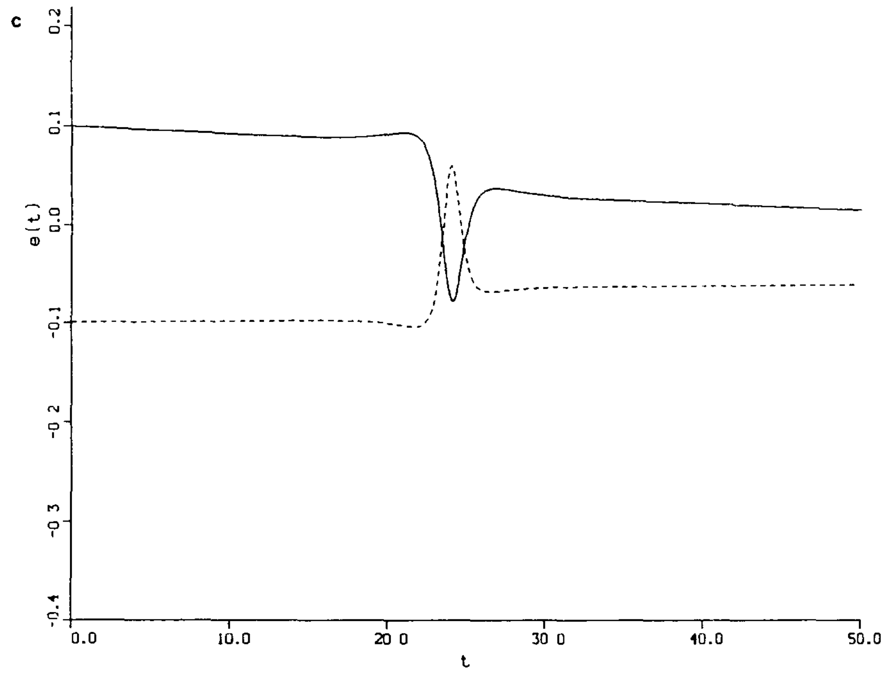

FIGURE 9—*Continued*

Before we go to the actual numerical experiments, let us remark that the number of operations required for the factorization of matrices plays no role in the comparison. Indeed, factorization is performed only once when the predictor–corrector pair is used. The situaton would be different if we used, say, Newton's iteration.

We ran our programs with the same initial condition as in Section 4: $I =]-20, 80[$, $T = 50$, and $\Delta t = \frac{1}{10}$. Table II summarizes the results. Comparing Fig. 1 (the standard Galerkin method with piecewise linear functions) and Fig. 6 (product approximation (l–l)), it appears that the two methods produce very similar results. Even if the behaviour of the first quantity is somewhat different (Fig. 1c and Fig. 6c), the fluctuations in both cases are equally smooth.

The Hermite cubics failed to meet the requirements. There is a significant discrepancy between the accuracy gained and the work involved. The weakness of the Hermite cubics may be attributed to the fact that it yields superfluous information; namely the value of the x-derivative, which is not needed here. This doubles the order of the system to be solved.

However, the (l–c) method performed well and seems to be slightly more efficient than the (l–l) method. It (Figs. 8a, b, and c) compared well with the standard Galerkin method (Figs. 1a, b, and c) despite the fact that the two quantities (Fig. 8c) undergo considerable fluctuations. We also tested our results with two iterations of the corrector (Figs. 7a, b, c; Figs. 9a, b, c). It appears that product approximation, in spite of the absence of conservation properties, is a valuable alternative to the standard Galerkin method.

6. Conclusion

The purpose of this work was to study the effect of the product approximation on the Galerkin solutions of the one-dimensional cubic Schrödinger equation. A Crank–Nicolson scheme was used to discretize in time and the resulting nonlinear system of algebraic equations was solved using a predictor–corrector pair. Under these circumstances, we have found that, provided piecewise linear functions were used as test functions, the product approximation version of the Galerkin method produces results which compare well with those obtained by the standard Galerkin method. Numerical experiments also suggests that more efficiency might be obtained by using piecewise linear functions as trial functions and cubic splines as test functions, even though in this case the two quantities $E_1^{n,m}$ and $E_2^{n,m}$ undergo considerable fluctuations.

Although we have restricted our attention to one particular nonlinear partial differential equation, the conclusions reached in this study lead us to the following conjecture: In the case where we are given a nonlinear problem for which the standard Galerkin method converges in space (in a sense to be specified), the authors expect that the product approximation version of the Galerkin method would produce similar results, provided the numerical solutions remain uniformly bounded in some suitable norm.

Other types of nonlinearity should be investigated. Two-dimensional problems would also be of considerable interest.

Acknowledgments

We acknowledge support from the Committee of Vice-Chancellors and Principals of the Universities of the United Kingdom under an ORS Award, NATO, the Natural Sciences and Engineering Research Council of Canada under a Postgraduate Award and Research Grant A-3597, the University of Dundee under a W. L McKenzie Award

References

1 I Christie, D F Griffiths, A R. Mitchell, and J M Sanz-Serna, *IMA J.* **1**, 253 (1981).
2 M Delfour, M Fortin, and G Payre, *J Comput Phys* **44**, 277 (1981)
3 D F Griffiths, A R Mitchell, and J Ll Morris, *Comp Meth Appl Mech Eng* **45**, 117 (1984)
4 B M Herbst, J Ll. Morris, and A R Mitchell, *J Comput Phys* **60**, 282 (1985)
5 A. R. Mitchell and R Wait, *The Finite Element Method in Partial Differential Equations* (Wiley, London, 1977).
6 Y Tourigny, M Math Thesis, University of Waterloo, 1986 (unpublished).
7 R S Varga, *Functional Analysis and Approximation Theory in Numerical Analysis* (Soc Indus Appl Math., Philadelphia, 1971)
8. V E Zacharov and A B Shabat, *Sov. Phys JETP* **34**, 62 (1972).