Comp. & Maths. with Appls. Vol. 12A, No. 7, pp. 859–884, 1986
Printed in Great Britain

0886–9553/86 $3.00 + .00
Pergamon Journals Ltd

FULLY DISCRETE GALERKIN METHODS FOR THE KORTEWEG-DE VRIES EQUATION†

JERRY L. BONA
Department of Mathematics, The University of Chicago, Chicago, IL 60637, U.S.A.

VASSILIOS A. DOUGALIS
Mathematics Department, University of Tennessee, Knoxville, TN 37996 and Mathematics Department, University of Crete, Iraklion, Crete, Greece

and

OHANNES A. KARAKASHIAN
Mathematics Department, University of Tennessee, Knoxville, TN 37996, U.S.A.

(*Received* 7 *October* 1985)

Communicated by E. L. Wachspress

Abstract—Fully discrete schemes for the numerical simulation of solutions of the periodic initial-value problem for the Korteweg-de Vries equation are introduced, implemented and tested. Of special interest are stable schemes featuring rates of convergence of order higher than two in both the spatial and temporal variable. A careful analysis of the relative and absolute efficiency of these schemes is carried out and one of the schemes is applied to demonstrate that solutions of certain generalized Korteweg-de Vries equations apparently may develop singularities in finite time.

1. INTRODUCTION

Described herein are some numerical methods for approximating the solutions of a class of partial differential equations that model the propagation of small-amplitude, long waves in nonlinear, dispersive media. In this paper, the first of two, the numerical schemes will be described and analyzed in the context of the periodic initial-value problem for the Korteweg-de Vries (KdV) equation, which is to determine a function $u = u(x, t)$ defined for all x and nonnegative t which, for all $t \geq 0$, is periodic of period 1 in x and satisfies

$$u_t + \eta u_x + uu_x + \epsilon u_{xxx} = 0, \quad \text{for } x \in \mathbf{R},\ 0 < t, \tag{1.1a}$$

$$u(x, 0) = u^0(x), \quad \text{for } x \in \mathbf{R}, \tag{1.1b}$$

where u^0 is a given, 1-periodic function and $\epsilon > 0$, $\eta \geq 0$ are constants. In the sequel to this paper the generalized KdV equation with nonlinear term $u^p u_x$, $p \geq 1$, and the KdV-Burgers equation with dissipative term $-\nu u_{xx}$, $\nu > 0$, will also be considered. For the KdV equation itself, the problem expressed in (1.1) has a smooth solution u corresponding to smooth initial data u^0 (cf. [1] or [2]).

In addition to presenting classes of numerical schemes, the existing theory concerning their stability and accuracy will be reviewed to provide a context for a series of numerical experiments on representative problems for (1.1) whose exact solution is known. A special focus of attention will be issues concerning the effective implementation of the proposed methods and an assessment of the resulting computational efficiencies.

Many numerical methods have been proposed, analyzed, and implemented for approximating solutions of (1.1). Zabusky[3] has given an authoritative survey of the literature in a recent review paper. The existing techniques include finite-difference methods (e.g., [4], [5], [6], [7], and [8]), spectral methods (e.g., [9], [10], [11], [12], [13], and [14]) and Galerkin-finite-element methods (e.g., [15], [16], [17], [18], [19], [20], and [21]).

†Work supported in part by the National Science Foundation and in part by the U.S. Army Research Office. Part of this work was done while the first author was visiting the Institute for Mathematics and its Applications at the University of Minnesota.

The numerical schemes studied here are fully discrete Galerkin methods that are based on a standard semi-discretization in the spatial variable x using smooth periodic splines on a uniform mesh on [0, 1]. (Most of the experiments described below were performed with quadratic or cubic splines.) For the temporal discretization various procedures are proposed, mainly second- and third-order accurate, A-stable, diagonally implicit Runge-Kutta methods coupled with Newton's method to solve the attendant nonlinear systems at each time step, and Rosenbrock methods. Because the solutions of problems of the type exemplified in (1.1) are smooth, these methods are well suited to performing stable and accurate computations with relatively large time steps, a feature of considerable practical importance (cf. the discussion in [22] and [23]). Indeed, it will appear that taking k to be of order h suffices in all cases to guarantee good accuracy; here k and h denote the temporal and spatial discretization lengths, respectively. These methods were implemented in a FORTRAN program that gives the user the choice of using spatial discretizations with splines of order r in the range $3 \leqslant r \leqslant 9$, combined with any of the aforementioned time-stepping procedures.

The paper is organized as follows. In Section 2 the numerical schemes are introduced and the rigorously established stability and convergence results for them are quoted. Section 3 is devoted to features of the coding of the schemes. Special attention is given to the description of data structures used to improve the efficiency of the procedure. Work estimates for the various methods are also provided. In Section 4 the outcome of an extensive experimental study of the accuracy and stability of these methods is presented. Using calculations performed over short temporal intervals, values of k and h are determined that minimize the work each method requires to achieve a given error tolerance. The relative computational efficiencies of the thus optimized procedures are then compared in detail. Comparisons are also effected over longer time scales. The overall conclusions are summarized in Section 5 and an interesting sample result is presented that makes use of the techniques in an exploratory mode.

2. THE NUMERICAL METHODS

We introduce here the precise numerical techniques that will be used throughout and provide commentary on theoretical aspects of these procedures.

All the fully discrete methods to be discussed are based on a standard Galerkin semi-discretization using smooth, periodic splines in the spatial variable. Let N be a positive integer, let $h = N^{-1}$ denote the uniform mesh length of the spatial discretization, and for integers j, set $x_j = jh$. If $r \geqslant 3$ is an integer and h is as just defined, denote by S_h^r the N-dimensional space of smooth, 1-periodic splines of order r, that is, the space of 1-periodic, piecewise polynomial functions of degree $r - 1$ on each subinterval $[x_j, x_{j+1}]$ which have $r - 2$ continuous derivatives. An element of S_h^r is determined by its values on [0, 1]. A convenient basis for S_h^r may be constructed as follows (cf. [24]). Let χ denote the characteristic function of the closed interval $[-1/2, 1/2]$ and let $\psi = \chi^{*r}$ be the r-fold convolution of χ. For any $j \in \mathbf{Z}$, set $\psi_j(x) = \psi(h^{-1}x - j)$ and define

$$\tilde{\phi}_j(x) = \sum_{\ell \in \mathbf{Z}} \psi_{j+\ell N}(x).$$

Then $\{\tilde{\phi}_j\}_{j=1}^N$ form a basis for S_h^r with the peak of $\tilde{\phi}_j$ occurring at x_j, $j = 1, 2, \ldots, N$. The basis $\{\phi_j\}_{j=1}^N$ actually used in our computations are obtained from the $\tilde{\phi}_j$ by scaling and a cyclic permutation so that the maximum value of each ϕ_j is one and the peak of ϕ_j occurs at $x = (N - [(r - 1)/2])h + (j - 1)h$, modulo one.

The usual inner product in $L_2([0, 1])$ is denoted $(\cdot, \cdot)$ and the associated norm by $\|\cdot\|$. Let $T > 0$ be given. A semi-discrete approximation $u_h = u_h(x, t)$ to (1.1) lying in S_h^r for each t in $[0, T]$ is defined by requiring that

$$(u_{ht} + \eta u_{hx} + u_h u_{hx}, \phi) - \epsilon(u_{hxx}, \phi') = 0 \quad \text{for all } \phi \in S_h^r \text{ and } 0 \leqslant t \leqslant T, \tag{2.1a}$$

$$u_h(\cdot, 0) = Pu^0, \tag{2.1b}$$

where Pu^0 is an element of S_h^r that approximates u^0 well and the third-order term has been integrated by parts to permit the use of quadratic splines. In practice P was taken to be the

orthogonal projection of u^0 onto S_h^r in L_2, that is $(Pu^0, \phi) = (u^0, \phi)$ for all $\phi \in S_h^r$. Other choices are possible however: Pu^0 could be taken as a polynomial interpolant, or one of the various quasi-interpolants, of u^0. It was established in [17] and [20] (for $\epsilon = 1$ and $\eta = 0$, but the analysis carries over without essential change to the case at hand) that if the initial data $u_h(\cdot, 0)$ is optimally close in L_2 to u^0 in the sense that $\|u_h(\cdot, 0) - u^0\| = O(h^r)$ as $h \to 0$, and if the solution of (1.1) is sufficiently smooth, then $u_h(x, t)$ exists, is unique, and satisfies the relation

$$\max_{0 \leq t \leq T} \|u_h(\cdot, t) - u(\cdot, t)\| = O(h^r)$$

as $h \to 0$. (In [17] the relevant theorem is stated and proved for $r \geq 4$, but if the third-order term is integrated by parts, the proof works in case $r = 3$.)

The system of equations (2.1a) and the initial conditions (2.1b) are equivalent to an initial-value problem for a system of ordinary differential equations. Indeed, setting

$$u_h(x, t) = \sum_{i=1}^{N} c_i(t)\phi_i(x),$$

(2.1) forces the conclusion that the unknown vector $\hat{c}(t) = [c_1, \ldots, c_N]$ satisfies

$$\mathbf{G}\frac{d\hat{c}}{dt} + \eta\mathbf{M}\hat{c} + F(\hat{c}) + \epsilon\mathbf{S}\hat{c} = 0, \quad 0 \leq t \leq T, \tag{2.2a}$$

with

$$\hat{c}(0) = \hat{c}^0. \tag{2.2b}$$

Here $\mathbf{G}$, $\mathbf{M}$, and $\mathbf{S}$ are $N \times N$ matrices whose entries are given by

$$\mathbf{G}_{ij} = (\phi_j, \phi_i), \quad \mathbf{M}_{ij} = (\phi_j', \phi_i), \quad \text{and } \mathbf{S}_{ij} = -(\phi_j'', \phi_i'),$$

for $1 \leq i, j \leq N$, $F(c)$ is an $\mathbf{R}^N$-valued function of $\hat{c}$ whose components are

$$F(\hat{c})_i = \sum_{k,j=1}^{N} c_k c_j(\phi_k\phi_j', \phi_i), \quad \text{for } 1 \leq i \leq N,$$

and $\hat{c}^0$ is the vector of coefficients of $u_h(\cdot, 0) = Pu^0$. Note that if Pu^0 is the L_2-projection of u^0 on S_h^r, then $\hat{c}^0$ is the solution of the linear system $\mathbf{G}\hat{c}^0 = \hat{U}^0$ where $\hat{U}^0 = [(u^0, \phi_1), \ldots, (u^0, \phi_N)]$. The matrix $\mathbf{G}$ is symmetric and positive definite whereas $\mathbf{M}$ and $\mathbf{S}$ are skew-symmetric. One verifies straightforwardly that $\mathbf{G}$, $\mathbf{M}$, and $\mathbf{S}$ are cyclic (circulant) matrices.

To compute an approximation to the solution u of (1.1) the system (2.2) of ordinary differential equations must be discretized. To achieve this several single-step methods were used which reduce, in the context of linear systems of ordinary differential equations, to A-stable schemes. These choices allowed the retention of higher-order accuracy without undue stability restrictions on the temporal discretization as a function of h. In what follows k will denote the constant, positive time step and t^n will stand for nk, $n = 0, 1, \ldots, J$, where it is taken that $T = kJ$ for some positive integer J.

Perhaps the most obvious temporal discretization is a Crank-Nicolson scheme in which one seeks $\{V^n\}_{n=0}^J$ in S_h^r such that

$$(V^{n+1} - V^n + k\eta V^{n+1/2} + kV^{n+1/2}V_x^{n+1/2}, \phi) - k\epsilon(V_{xx}^{n+1/2}, \phi') = 0 \tag{2.3a}$$

for all ϕ in S_h^r, $0 \leq n \leq J - 1$, and

$$V^0 = Pu^0,$$

where $V^{n+1/2} = (V^n + V^{n+1})/2$. For every n in the range $[0, J-1]$, V^{n+1} is obtained from (2.3a) as the solution of a nonlinear system of equations. In [17] it was shown that if the solution u of (1.1) is sufficiently smooth (for $\eta = 0$, $\epsilon = 1$, and $r \geq 4$, but the proof is easily extended to cover the present case, even for $r = 3$) then for k and h sufficiently small the solution of (2.3) exists, is unique, and satisfies the error estimate $\max_n \|u(\cdot, t^n) - V^n\| = O(k^2 + h^r)$. For the uniqueness of V^n, the proof given in [17] requires the weak condition that $kh^{-1/2}$ be sufficiently small. As a practical matter one computes an approximation U^{n+1} to the exact solution V^{n+1} of (2.3) by Newton's method. It is further established in [17] that if $kh^{-3/4}$ is sufficiently small and if a starting value for the Newton method is obtained by extrapolation from known values of U^n, then a single Newton iteration (i.e., solving one linear system of equations) suffices to guarantee stability and to preserve the overall accuracy of the exact solution $\{V^n\}_{n=0}^J$. Thus there emerges a scheme requiring the solution of one system of linear equations at each time level that produces an approximation $\{U^n\}_{n=0}^J$ satisfying the overall error estimate $\max_n \|U^n - u(\cdot, t^n)\| = O(k^2 + h^r)$. We shall return to (2.3) below, interpreting it within a general class of Runge-Kutta-type schemes for the temporal discretization. (A technical aside is warranted here. The proof of convergence of Newton's method for (2.3) given in [17], when adapted to the case $r = 3$, requires the additional assumption $k \geq h^{3/2}$, a requirement that is certainly compatible with the presumption $k \leq ch^{3/4}$ mentioned above.)

Attention is now given to higher-order accurate, single-step methods for use in the temporal discretization of the system (2.2). The first family of schemes considered here are the well known, semi-implicit, Runge-Kutta (RK) methods (cf. [25] or [26] and the references contained therein). A q-stage diagonally implicit RK (DIRK) method for the autonomous, nonlinear system of ordinary differential equations $\dot{y} = f(y)$ is determined by a table of constants $\mathbf{A}|\hat{b}$ where $\mathbf{A} = (a_{ij})$, $1 \leq i, j \leq q$, is a lower triangular $q \times q$ matrix such that $a_{ii} = \beta \neq 0$, $1 \leq i \leq q$, and $\hat{b} = (b_1, \ldots, b_q)$. The matrix $\mathbf{A}$ and vector $\hat{b}$ are used to compute approximations y^n to $y(t^n)$ as follows:

$$y^{n,i} = y^n + k \sum_{j=1}^{q} a_{ij} f(y^{n,j}), \quad 1 \leq i \leq q, \tag{2.4}$$

and

$$y^{n+1} = y^n + k \sum_{j=1}^{q} b_j f(y^{n,j}), \quad 0 \leq n \leq J-1. \tag{2.5}$$

At each time step, such a method requires the solution of one nonlinear system of equations of the form, $y^{n,i} - k\beta f(y^{n,i}) =$ known vector, for each of the q intermediate stages i, $1 \leq i \leq q$. If the off-diagonal nonlinear terms in (2.4) are eliminated and the results substituted into (2.5), there results the usual form of these methods, namely

$$y^{n,i} - k\beta f(y^{n,i}) = y^n + \sum_{j=1}^{i-1} \mu_{ij}(y^{n,j} - y^n), \quad 1 \leq i \leq q, \tag{2.6}$$

$$y^{n+1} = y^n + k \sum_{i,j=1}^{q} b_i (\mathbf{A}^{-1})_{ij} (y^{n,j} - y^n). \tag{2.7}$$

The entries of the strictly lower-triangular, $q \times q$ matrix (μ_{ij}) are $\mu_{ij} = \delta_{ij} - \beta(\mathbf{A}^{-1})_{ij}$ where δ_{ij} is Kronecker's delta function. In (2.6) and henceforth we follow the convention that $\Sigma_{j=m}^{n} = 0$ if $n < m$. In general, a q-stage DIRK method whose order of accuracy is p will be referred to as a (q, p) scheme.

The simplest example of q-stage DIRK schemes is when $q = 1$ and the method is given by the tableau

$$\tfrac{1}{2} \mid 1. \tag{2.8}$$

In this case the method is defined by the equations

$$y^{n,1} = y^n + \frac{k}{2} f(y^{n,1}),$$

$$y^{n+1} = y^n + kf(y^{n,1}).$$

As $y^{n,1} = (y^{n+1} + y^n)/2$, this scheme is equivalent to the second-order accurate, midpoint scheme

$$y^{n+1} = y^n + kf((y^{n+1} + y^n)/2),$$

which, in the case of the semi-discretization (2.1a), coincides with the Crank-Nicolson scheme (2.3). Thus $q = 1$ and $p = 2$, and so (2.8) defines a (1, 2) scheme. Another example of interest here is the (2, 3) DIRK method given by the tableau

$$\begin{array}{cc|c} \beta & 0 & \frac{1}{2} \\ 1-2\beta & \beta & \frac{1}{2} \end{array}, \quad \text{where } \beta = \frac{1}{2}(1 + 3^{-\frac{1}{2}}). \tag{2.9}$$

Our computer code included both (2.8) and (2.9) as time-stepping options. Also included was the well-known (3, 4) DIRK method with diagonal elements

$$\beta = \frac{1}{\sqrt{3}} \cos\left(\frac{\pi}{18}\right) + \frac{1}{2}$$

(cf. [3], [12], [15] or [16]), but its use in the present context was found to be expensive and so it will not be featured further in this exposition.

In the scalar case $\dot{y} + \lambda y = 0$ it is well known that these DIRK methods reduce to A-stable schemes associated with Nørsett rational approximations $r(z)$ to $\exp(-z)$, with denominator $(1 + \beta z)^q$, where $z = \lambda k$ (cf. [25], [26], [27]). In particular the rational approximation corresponding to (2.8) is the (1, 1) Padé approximant to $\exp(-z)$ given by $r_1(z) = (1 - \frac{1}{2}z)/(1 + \frac{1}{2}z)$, whilst the (2, 3) DIRK method (2.9) corresponds to $r_2(z) = [1 + (2\beta - 1)z + (\beta^2 - 2\beta + \frac{1}{2})z^2]/(1 + \beta z)^2$, where $\beta = \frac{1}{2}(1 + 3^{-\frac{1}{2}})$. Both of these rational approximations (and the one corresponding to the neglected (3, 4) DIRK method mentioned above) satisfy $|r(z)| \leq 1$ for all complex z with $\mathrm{Re}(z) \geq 0$, so yielding A-stable schemes. Actually, both $\mathbf{M}$ and $\mathbf{S}$ are skew-symmetric and so possess purely imaginary spectra. Hence, if the nonlinear term F is ignored, the stability of the time stepping procedure in the context of integrating the system (2.2a) may be understood by studying the behavior of $r(z)$ for purely imaginary z. The (1, 2) DIRK method has $|r_1(ix)| = 1$ for any real x, and so this method is conservative for linear ordinary differential equations of such form. The (2, 3) DIRK scheme has

$$\sup_{\gamma < |x|} |r_2(ix)| < 1$$

for real x and any $\gamma > 0$. In fact, as $x \to 0$,

$$r_2(ix) = e^{-ix} - \frac{4\beta - 1}{24} x^4 + O(x^5),$$

and so this method is dissipative in the context of the linearized KdV equation. In the nonlinear case wherein F is not ignored, it is easily seen by taking $\phi = V^{n+1/2}$ in (2.3) that the Crank-Nicolson scheme conserves the L^2 norm of the initial data. Our numerical experiments showed that even when the solution of the nonlinear equations was approximated by a single Newton iteration per time step, a negligible loss of conservation resulted. The (2, 3) DIRK method is dissipative in the nonlinear case as well.

Applying the DIRK methods in their general forms (2.6), (2.7) to the semi-discretization (2.1a) leads to the equations,

$$(V^{n,i} + k\beta\eta V_x^{n,i}, \phi) - k\beta\{(\tfrac{1}{2}(V^{n,i})^2, \phi') + \epsilon(V_{xx}^{n,i}, \phi')\} = (V^n, \phi) + \sum_{j=1}^{i-1} \mu_{ij}(V^{n,j} - V^n, \phi), \quad (2.10a)$$

for all $\phi \in S_h^r$, $1 \leqslant i \leqslant q$, and

$$V^{n+1} = V^n + \sum_{i,j=1}^{q} b_i(\mathbf{A}^{-1})_{ij}(\mathrm{V}^{n,j} - \mathrm{V}^n). \quad (2.10b)$$

Hence to obtain V^{n+1} from V^n via (2.10) one solves q, $N \times N$, nonlinear systems of equations to obtain the $V^{n,i}$ for use in (2.10b). As for the (1, 2) DIRK method already discussed, it has been established in [18] that unique solutions $\{V^n\}$ of the nonlinear systems associated with the (2, 3) and (3, 4) DIRK schemes exist and comprise a stable sequence in S_h^r provided that some weak relations between k and h are satisfied. As before, the solutions of the systems in (2.10a) may be approximated by Newton's method, so yielding approximations $\{U^n\}$ to $\{V^n\}$ in S_h^r. For Newton's method to be effective requires good starting values $U_0^{n,i}$. These are obtained as linear combinations of the form,

$$U_0^{n,i} = \lambda_{i,0}U^n + \lambda_{i,1}U^{n-1} + \cdots + \lambda_{i,q}U^{n-q}, \quad (2.11a)$$

$1 \leqslant i \leqslant q$, which use previously determined values. The coefficients $\lambda_{i,j}$ depend upon the particular (q, p) DIRK method that is in question. (Note that (2.11a) may only be used for $n \geqslant q$. Another starting procedure must be used for the first q steps.) Let $U_1^{n,i}$ denote the result of one Newton iteration performed on (2.10a) with initial guess $U_0^{n,i}$. The $U_1^{n,i}$ are obtained as the solution of the linear systems,

$$(U_1^{n,i} + k\beta\eta U_{1x}^{n,i}, \phi) - \epsilon k\beta(U_{1x}^{n,i}, \phi') - k\beta(U_0^{n,i}U_1^{n,i}, \phi') = (U^n, \phi) + \sum_{j=1}^{i-1} \mu_{ij}(U_1^{n,j} - U^n, \phi) - \tfrac{1}{2}k\beta([U_0^{n,i}]^2, \phi'), \quad (2.11b)$$

for all $\phi \in S_h^r$, $1 \leqslant i \leqslant q$. Then U^{n+1} is computed by the analog of (2.10b)

$$U^{n+1} = U^n + \sum_{i,j=1}^{q} b_i(\mathbf{A}^{-1})_{ij}(U_1^{n,j} - U^n). \quad (2.11c)$$

Hence a total of q, linear, $N \times N$ systems must be solved to compute U^{n+1}. The matrices associated with these systems are nonsymmetric, but positive definite for k and h sufficiently small. The computational issues concerned with the solution of these systems will be discussed in detail in Section 3.

Our computational experience indicates that the approximations $\{U^n\}$ satisfy the error bound $\|U^n - u(\cdot, t^n)\| = O(k^p + h^r)$, where $p = 3$, respectively 4, if the (2, 3), respectively (3, 4) DIRK schemes is used. The theoretical developments in [18] did not quite establish this result, but rather demonstrated that the method represented by (2.11a–c) is stable and that $\|U^n - u(\cdot, t^n)\| = O(k^2 + h^r)$. If the scheme (2.11a–c) is modified by the addition of small order perturbations, then the resulting scheme, producing an approximation $\{\tilde{U}^n\}$, satisfies $\|\tilde{U}^n - u(\cdot, t^n)\| = O(k^p + h^r)$. In all cases, the proofs require that k/h remain bounded as $k, h \to 0$. Thus there is a gap at this point between what is inferred on the basis of numerical experiments and what can be proved unequivocally.

It is evident from (2.11b) that the matrices associated with the linear systems that have to

be solved change not only from step to step, but even from stage to stage. To avoid this, Rosenbrock-type methods[28, p. 223] of second- and third-order accuracy were also employed. In the context of linear, constant-coefficient systems of ordinary differential equations these schemes are A-stable and, like their DIRK counterparts, reduce to the same rational approximation to the exponential when applied to $\dot{y} + \lambda y = 0$. For the system $\dot{y} = f(y)$ the Rosenbrock methods take the form

$$[I - k\beta f_y(y^n)]y^{n,i} = kf\left(y^n + \sum_{j=1}^{i-1} a_{ij}y^{n,j}\right), \tag{2.12a}$$

for $1 \leqslant i \leqslant q$, and

$$y^{n+1} = y^n + \sum_{j=1}^{q} b_i y^{n,i}, \tag{2.12b}$$

where f_y is the Jacobian of the nonlinear map f and β, a_{ij}, $1 \leqslant i \leqslant q$, $1 \leqslant j \leqslant i - 1$, and b_i, $1 \leqslant i \leqslant q$, are constants that are generally different from the analogous constants that define q-stage DIRK methods of the type given in (2.4) and (2.5). Computing with Rosenbrock methods thus requires forming the Jacobian f_y once at each time step and then solving q systems of linear equations with the same matrix. Forming the Jacobian for the system (2.1a) is quite easy. Moreover, this Jacobian was already being used in the Newton iteration associated with the DIRK methods. Specifically, applying (2.12) to the semi-discretization (2.1a) produces a sequence $\{U^n\}_{n=0}^J$ in S_h^r with $U^0 = Pu^0$ which satisfies

$$\begin{aligned}(U^{n,i} + \eta k\beta U_x^{n,i}, \phi) - \epsilon k\beta(U_{xx}^{n,i}, \phi') - k\beta(U^n U^{n,i}, \phi') \\ = \frac{k}{2}\left(\left[U^n + \sum_{j=1}^{i-1} a_{ij}U^{n,j}\right]^2, \phi'\right) \\ - k\eta\left(\left[U^n + \sum_{j=1}^{i-1} a_{ij}U^{n,j}\right]_x, \phi\right) \\ + \epsilon k\left(\left[U^n + \sum_{j=1}^{i-1} a_{ij}U^{n,j}\right]_{xx}, \phi'\right),\end{aligned} \tag{2.13a}$$

for all $\phi \in S_h^r$, $1 \leqslant i \leqslant q$, and

$$U^{n+1} = U^n + \sum_{i=1}^{q} b_i U^{n,i}, \tag{2.13b}$$

for $0 \leqslant n \leqslant J - 1$.

In our computer program two such Rosenbrock methods have been implemented. The first amounts to a linearized version of the trapezoidal rule of the form (2.12) with $q = 1$, $\beta = \frac{1}{2}$, $a_{ij} = 0$, $b_1 = 1$ which has second-order accuracy and which is essentially as economical as the (1, 2) DIRK-Newton method (2.8). Also implemented was the Calahan method[29], [28], a two-stage, third-order accurate scheme having $\beta = \frac{1}{2}(1 + 3^{-\frac{1}{2}})$, $a_{21} = 2 - 4\beta$, $b_1 = \frac{3}{4}$, $b_2 = \frac{1}{4}$.

At present there is no proof of convergence for the Rosenbrock methods in the context discussed here of the KdV equation. Experimentally we have found that the two Rosenbrock methods implemented in our code were accurate if k/h remained bounded as k, $h \to 0$, and that they then yielded optimal-order L_2 error bounds for $\max_n\|U^n - u(\cdot, t^n)\|$ which were $O(k^2 + h^r)$ for the one-stage Rosenbrock method and $O(k^3 + h^r)$ for the two-stage Calahan method. In the context of constant-coefficient linear systems the (1, 2) Rosenbrock method is conservative, whilst the Calahan method is dissipative since they coincide then with the corresponding

DIRK schemes. In the nonlinear situation of our numerical experiments on the KdV equation it was observed that the (1, 2) Rosenbrock scheme induced only a negligible amount of dissipation.

3. COMPUTATIONAL CONSIDERATIONS

Issues are considered here that connect with the practical implementation of the various numerical methods presented in the last section. We describe particular computational algorithms and the associated data structures that are incorporated into our computer program to efficiently compute approximations to solutions of the KdV equation. Also presented are reasonably sharp estimates of the number of arithmetic operations required per time step by each of the suggested numerical schemes, as a function of the number N of spatial intervals and the order r of the underlying spline space.

As a matter of notation a circumflex over a variable connotes that variable to be a vector rather than a scalar. Denote by $\hat{\phi} = \hat{\phi}(x)$ the N-vector whose components are the basis functions, $\phi_1, \ldots, \phi_N$, introduced in Section 2. Thus $\hat{\phi} = (\phi_1, \ldots, \phi_N)$, and if $\zeta \in S_h^r$, there is a unique $\hat{z} = (z_1, \ldots, z_N)$ in $\mathbf{R}^N$ such that $\zeta(x) = \sum z_i \phi_i(x) = \hat{z} \cdot \hat{\phi}$.

In matrix representation relative to the chosen basis for S_h^r, the algorithm (2.11b) for the DIRK method with one Newton iteration is

$$[\mathbf{G} + \eta k\beta \mathbf{M} + \epsilon k\beta \mathbf{S} - \beta k \mathscr{F}(U_0^{n,i})]\hat{U}_1^{n,i} = \mathbf{G}\hat{U}^n + \sum_{j=1}^{i-1} \mu_{ij} \mathbf{G}\hat{U}_1^{n,j} - \frac{k\beta}{2} \hat{f}(U_0^{n,i}, U_0^{n,i}), \quad (3.1)$$

for $1 \leq i \leq q$, where $\hat{U}^n$, $\hat{U}_1^{n,i} \in \mathbf{R}^N$ are the vectors of the coefficients of the S_h^r-functions U^n, $U^{n,i}$, respectively, $\hat{f}(\zeta, \psi) \in \mathbf{R}^N$ is defined by

$$\hat{f}(\zeta, \psi) = (\zeta\psi, \hat{\phi}') \quad (3.2)$$

for $\zeta, \psi \in S_h^r$, and the $N \times N$ matrix $\mathscr{F}(\psi)$ has components $\mathscr{F}(\psi)_{ij}$ defined by

$$\mathscr{F}(\psi)_{ij} = (\psi\phi_j, \phi_i'), \quad (3.3)$$

for ψ in S_h^r. In the same notation, the Rosenbrock method (2.13a) may be written as

$$[\mathbf{G} + \eta k\beta \mathbf{M} + \epsilon k\beta \mathbf{S} - k\beta \mathscr{F}(U^n)]\hat{U}^{n,i} = \frac{k}{2} \hat{f}(Y^{n,i}, Y^{n,i}) - k(\eta \mathbf{M} + \epsilon \mathbf{S})\hat{Y}^{n,i}, \quad (3.4)$$

for $1 \leq i \leq q$, where

$$Y^{n,i} = U^n + \sum_{j=1}^{i-1} a_{ij} U^{n,j}, \quad (3.5)$$

for $1 \leq i \leq q$, and where $\hat{Y}^{n,i}$, $\hat{U}^{n,i} \in \mathbf{R}^N$ are the coefficients of $Y^{n,i}$, $U^{n,i}$, respectively, relative to the basis $\{\phi_j\}_{j=1}^N$.

These formulae allow a count of arithmetic operations to be initiated for the proposed methods. In making such counts, account will only be taken of computations (multiplications) that are repeated every time step. Thus set-up costs such as that of assembling the matrices $\mathbf{G}$, $\mathbf{M}$, $\mathbf{S}$, and the array $(\phi_k\phi_l, \phi_j')$, and that of computing U^0 will be ignored. Moreover, the cost of calls to subroutines that calculate exact solutions, compute errors, and so on, are ignored. With these provisos in force, inspection of (3.1) and (3.4) reveals that the following operations are performed in the DIRK-Newton or Rosenbrock scheme.

(i) Given $\hat{y} \in \mathbf{R}^N$, evaluate $\mathbf{G}\hat{y}$, $\mathbf{M}\hat{y}$, and $\mathbf{S}\hat{y}$.
(ii) Given $\zeta, \psi \in S_h^r$, evaluate the N-vector $\hat{f}(\zeta, \psi)$ given by (3.2).
(iii) Given $\psi \in S_h^r$, evaluate the $N \times N$ matrix $\mathscr{F}(\psi)$ given by (3.3).

(iv) Given $\psi \in S_h^r$, $\hat{g} \in \mathbf{R}^N$, evaluate the elements of the matrix $\mathscr{T}(\psi)$ and solve the linear system $\mathscr{T}(\psi)\hat{z} = \hat{g}$, where

$$\mathscr{T}(\psi) = \mathbf{G} + c_1 k\mathbf{M} + c_2 k\mathbf{S} + c_3 k\mathscr{F}(\psi), \tag{3.6}$$

If $N \geqslant 2r - 2$, the matrices $\mathbf{G}$, $\mathbf{M}$, and $\mathbf{S}$ are cyclic with first rows of the form

$$\hat{a} = (a_1, a_2, \ldots, a_r, 0, \ldots, 0, a_{N-r+2}, \ldots, a_N). \tag{3.7}$$

We focus temporarily on such matrices. To ease the task of handling component-indicating indices, they will always be interpreted modulo N. Thus if $\hat{y} = (y_1, \ldots, y_N)$, then $y_0 = y_N$, $y_{-1} = y_{N-1}$, $y_{N+1} = y_1$, and so on. Define a mapping * that associates to $\hat{v} \in \mathbf{R}^N$ the element $\hat{v}^* \in \mathbf{R}^{N+2r-2}$ given by

$$v_i^* = \begin{cases} v_{i+N-r+1}, & \text{if } 1 \leqslant i \leqslant r-1, \\ v_{i-r+1}, & \text{if } r \leqslant i \leqslant N+r-1, \\ v_{i-N-r+1}, & \text{if } N+r \leqslant i \leqslant N+2r-2. \end{cases} \tag{3.8}$$

That is, $\hat{v}^* = (v_{N-r+2}, \ldots, v_N, v_1, \ldots, v_N, v_1, \ldots, v_{r-1})$.

In terms of the notational provisions just made, we may state the following result which is relevant to the computational problem (i) above.

LEMMA 3.1

Let $\mathbf{C} = (c_{ij})$ be an $N \times N$ cyclic matrix whose first row $\hat{c}$ is of the form indicated in (3.7). Then for any $\hat{y} \in \mathbf{R}^N$, $\mathbf{C}\hat{y}$ can be computed, using only $(2r-1)N$ multiplications, from the identities,

$$\sum_{j=1}^{N} c_{ij} y_j = \sum_{j=1}^{2r-1} c_j^* y_{j+i-1}^*, \tag{3.9}$$

for $1 \leqslant i \leqslant N$. Moreover, if $\mathbf{C}$ is also supposed to be symmetric or skew-symmetric, then the number of multiplications needed to compute $\mathbf{C}\hat{y}$ is rN via the identities,

$$\sum_{j=1}^{N} c_{ij} y_j = c_r^* y_{r+i-1}^* + \sum_{j=1}^{r-1} c_j^* (y_{j+i-1}^* + (-1)^\sigma y_{2r-j+i-1}^*), \tag{3.10}$$

for $1 \leqslant i \leqslant N$, where $\sigma = 0$ or 1 depending on whether $\mathbf{C}$ is symmetric or skew-symmetric, respectively.

Proof. The relations (3.9) are easily established by induction on i. The formulae (3.10) follow immediately since $c_j^* = (-1)^\sigma c_{2r-j}^*$ for $1 \leqslant j \leqslant r-1$, with $\sigma = 0$ or 1 depending on whether $\mathbf{C}$ is symmetric or skew-symmetric, respectively. The stated multiplication counts follow instantly from (3.9) and (3.10).

When coding the sums on the right-hand side of (3.9) the nonzero elements of the first row of $\mathbf{C}$ are stored in the order $c_{N-r+2}, \ldots, c_N, c_1, \ldots, c_r$. Moreover, rather than creating the $(N + 2r - 2)$-vector $\hat{y}^*$ from $\hat{y}$ by using IF statements, the index vector $\hat{n}^*$ is created once and stored, where $\hat{n} = (1, 2, \ldots, n)$, and y_j^* is obtained as $y_{n_j^*}$. This convention is followed whenever computations with the $(N + 2r - 2)$-vectors $\hat{y}^*$ are effected.

The just-described data structures are also useful in evaluating the nonlinear term $\hat{f}(\zeta, \psi)$. The calculation of $\hat{f}(\zeta, \psi)$ is described here in general, even though the schemes in view utilize only terms of the form $\hat{f}(\psi, \psi)$. There are methods (e.g., the theoretically important modifications of the DIRK schemes mentioned in Section 2) for which terms of the form $\hat{f}(\zeta, \psi)$ with $\zeta \neq \psi$

are encountered, and it has therefore seemed useful to keep the discussion unrestricted. Write $\zeta = \hat{z} \cdot \hat{\phi}$ and $\psi = \hat{y} \cdot \hat{\phi}$, so that

$$\hat{f}(\zeta, \psi)_i = (\zeta\psi, \phi_i') = \sum_{m,n=1}^{N} z_m y_n (\phi_m \phi_n, \phi_i') = \sum_{m,n=1}^{N} z_m y_n f_{mn}^i, \tag{3.11}$$

where

$$f_{mn}^i = (\phi_m \phi_n, \phi_i'), \tag{3.12}$$

for $1 \leqslant i, m, n \leqslant N$. Because the ϕ_i are 1-periodic and are related to one another by translation, it follows that

$$f_{m+l\ n+l}^{i+l} = f_{mn}^i, \tag{3.13}$$

for any integer l. Moreover, since the support of the ϕ_j has a length of r spatial intervals, the $N \times N$ array (f_{mn}^i) has less than $(2r-1)^2$ nonzero elements, for each i. Actually, it is easily determined from (3.12) and the properties of the ϕ_j that (f_{mn}^i) has $3r(r-1)$ nonzero elements. However, it is much easier from a programming point of view to consider (f_{mn}^i) as a $(2r-1) \times (2r-1)$ square array. The advantage of the additional zeroes is thereby sacrificed when multiplying (f_{mn}^i) by $(2r-1)$-vectors, though this advantage could only be exacted at the cost of heavy use of IF statements. For similar reasons, we shall use only the obvious symmetry $f_{mn}^i = f_{nm}^i$, whilst recognizing that these arrays possess other symmetries.

In view of (3.13) only those elements of f_{mn}^i that correspond to some fixed value of i (we took $i = r$) need be computed and stored. Moreover, it is a consequence of the chosen ordering of the basis function $\{\phi_j\}_{j=1}^N$ that $f_{mn}^r = 0$ if either m or n exceeds $2r-1$. The following result is directly applicable to the computational problem (ii) above.

LEMMA 3.2

Let $\zeta = \hat{z} \cdot \hat{\phi}$, $\psi = \hat{y} \cdot \hat{\phi}$ lie in S_h^r. The identity

$$\hat{f}(\zeta, \psi)_i = \sum_{m,n=1}^{2r-1} f_{mn}^r z_{m+i-1}^* y_{n+i-1}^*, \tag{3.14}$$

which holds for $1 \leqslant i \leqslant N$, may be used to evaluate $\hat{f}(\zeta, \psi)$ at a cost of $2r(2r-1)N$ multiplications. Furthermore, $\hat{f}(\psi, \psi)$ may be evaluated with $(r+2)(2r-1)N$ multiplications using the relation

$$\hat{f}(\psi, \psi)_i = \sum_{m=1}^{2r-1} f_{mm}^r (y_{m+i-1}^*)^2 + 2 \sum_{1 \leqslant n < m \leqslant 2r-1} f_{mn}^r y_{m+i-1}^* y_{n+i-1}^*, \tag{3.15}$$

for $1 \leqslant i \leqslant N$.

Proof. Since $f_{mn}^r = 0$ for $m, n \geqslant 2r$, the formula (3.14) is equivalent to

$$\hat{f}(\zeta, \psi)_i = \sum_{m,n=1}^{N} f_{mn}^r z_{m+i-1}^* y_{n+i-1}^*,$$

and this is easily established, first for $i = r$, and then inductively for $i > r$ or $i < r$. Once (3.14) is in hand, (3.15) follows and the multiplication counts are obvious consequences of these two sets of formulae.

Attention is now turned to the computational problem (iii), the evaluation of the matrix $\mathscr{F}(\psi)$

given by (3.3). If $\psi = \hat{y} \cdot \hat{\phi}$ it follows from (3.2) that $\mathscr{F}(\psi)$ may be assembled by first computing the array $\tilde{\mathscr{F}}(\psi)$ given by

$$\tilde{\mathscr{F}}(\psi)_{ij} = \sum_{m=1}^{2r-1} f^i_{mj} y^*_{m+i-1}, \tag{3.16}$$

$1 \leqslant i \leqslant N$, $1 \leqslant j \leqslant 2r - 1$, and then defining $\mathscr{F}(\psi)$ by

$$\mathscr{F}(\psi)_{ij} = \begin{cases} \tilde{\mathscr{F}}(\psi)_{ij-i+r} & \text{if } 1 \leqslant i \leqslant N,\ i - r + 1 \leqslant j \leqslant i + r - 1, \\ 0 & \text{otherwise.} \end{cases} \tag{3.17}$$

In (3.17) the indices are interpreted modulo N following the convention in force here. The matrix $\mathscr{F}(\psi)$ has the same structure of zeros as the cyclic matrices $\mathbf{G}$, $\mathbf{M}$, and $\mathbf{S}$. Moreover, its $(2r - 1)N$ nonzero elements may be evaluated via (3.16) with no more than $(2r - 1)^2 N$ multiplications. This completes the discussion of the computational issue (iii).

Consider now the matrix $\mathscr{T}(\psi)$ defined in (3.6). As $\mathbf{G}$, $\mathbf{M}$, and $\mathbf{S}$ are computed but once, $\mathscr{T}(\psi)$ may, for a given $\psi \in S^r_h$, be assembled using $2r(2r - 1)N$ multiplications. For both the DIRK-Newton methods (3.1) and the Rosenbrock methods (3.4) it is necessary to solve q linear systems involving matrices of the form $\mathscr{T}(\psi)$ in order to advance the solution by one time step. As the matrices $\mathscr{T}(\psi)$ change from step to step for the Rosenbrock methods and even from stage to stage for DIRK-Newton methods, it is fortunate that their calculation may be accomplished efficiently. Moreover, the linear systems that arise may also be solved efficiently as is now indicated.

Given $\psi \in S^r_h$, a system of equations of the form

$$\mathscr{T}(\psi)\hat{z} = \hat{g} \tag{3.18}$$

may be solved in the following way.† As mentioned above, $\mathscr{T} = \mathscr{T}(\psi)$ has the same zero structure as the cyclic matrix $\mathbf{G}$, and so it may be written in the form $\mathscr{T} = \mathscr{T}_b + \mathscr{T}_c$ where $\mathscr{T}_b$ is a diagonally banded matrix with bandwidth $2r - 1$ and $\mathscr{T}_c$ consists only of the upper right and lower left corners of $\mathscr{T}$. To solve (3.18), the following steps are effective.

(a) Factor $\mathscr{T}_b$ (without pivoting) into upper and lower triangular banded matrices using the standard banded factoring routine. This costs $r(r - 1)N$ multiplications. (N.B. The matrix $\mathscr{T}$ is real and positive for k and h small enough (cf. [18]). At no time in our calculations did we perceive any need for pivoting.)

(b) Solve $\mathscr{T}_b\hat{z}_b = \hat{g}$. This costs $(2r - 1)N$ multiplications since $\mathscr{T}_b$ has already been factored.

(c) Compute the $2r - 2$ N-vectors $\hat{\epsilon}^j$, $j = 1, \ldots, r - 1, N - r + 2, \ldots, N$ that satisfy $\mathscr{T}_b\hat{\epsilon}^j = \hat{e}^j$, where the i^{th} component of $\hat{e}^j$ is δ_{ij}, the Kronecker δ-function. This costs $(2r - 2)(2r - 1)N$ multiplications.

(d) Evaluate the $N \times N$ matrix $\mathbf{C}$ whose j^{th} column is $\hat{e}^j + \mathscr{T}_c\hat{\epsilon}^j$ if $j = 1, \ldots, r - 1$, $N - r + 2, \ldots, N$, and whose j^{th} column is zero otherwise. Note that only the four $(r - 1) \times (r - 1)$ corners of $\mathbf{C}$ are nonzero. The matrix $\mathbf{C}$ is compressed into a $(2r - 2) \times (2r - 2)$ array and factored as a product of upper and lower triangular matrices. This may be accomplished at a total expenditure of order r^3 multiplications. Being independent of N, this cost is ignored.

(e) The factored form of the compressed matrix $\mathbf{C}$ may be used to evaluate the $2r - 2$ nonzero entries of the N-vector $\hat{\lambda} = (\lambda_1, \ldots, \lambda_{r-1}, 0, \ldots, 0, \lambda_{N-r+2}, \ldots, \lambda_N)$ that satisfies $\mathbf{C}\hat{\lambda} = -\mathscr{T}_c\hat{z}_b$. The cost of this determination is of order r^2, and so is ignored.

(f) Finally the solution $\hat{z}$ of (3.18) is computed via the formula

$$\hat{z} = \hat{z}_b + \sum_{j=1}^{r-1} \lambda_j\hat{\epsilon}^j + \sum_{j=N-r+2}^{N} \lambda_j\hat{\epsilon}^j.$$

This takes $(2r - 2)N$ multiplications.

†The authors wish to record their thanks to T. Dupont for bringing this implementation to their attention.

Briefly summarized, steps (a), (c), and (d) are calculations that do not involve the right-hand side $\hat{g}$ of (3.18). Ignoring calculations whose cost depends only on r, the total number of multiplications involved in carrying out these steps is $(5r^2 - 7r + 2)N$. Steps (b), (e), and (f) involve $\hat{g}$ and cost $(4r - 3)N$ multiplications in total. This completes the analysis of the implementation of (iv).

It is now a straightforward task to count the total number of multiplications needed to advance the numerical approximation to the solution of (1.1) by one time step using a q-stage DIRK-Newton or Rosenbrock method. In the case of the DIRK-Newton method our scheme requires the calculation of the starting values $U_0^{n,i}$ for the Newton iteration by (2.11a), evaluation of q matrices of the form (3.6), assemblage of the right-hand sides of the q linear systems (3.1) and the determination of their solutions, and finally the computation of U^{n+1} by (2.11c). The total number of multiplications needed for these steps is $(q^2 + q[11r^2 - r + 1])N$. For the Rosenbrock methods the matrix on the left-hand side of (3.4) is evaluated and factored once, the right-hand sides of the q linear systems (3.4) are formed and their solutions determined, and U^{n+1} is then computed via (2.13b). The total number of multiplications for these steps is $([9r^2 - 9r + 1] + q[2r^2 + 8r - 2])N$. The multiplication counts for the two classes of methods are shown for some practically interesting values of q and r in Table 1.

Table 1. Number of operations per time step per spatial mesh interval for the DIRK-Newton and Rosenbrock methods.

r	q	DIRK-Newton	Rosenbrock
3	1	98	95
	2	198	135
	3	300	
4	1	174	171
	2	350	233
	3	528	
6	1	392	389
	2	786	507
	3	1182	

4. ACCURACY AND EFFICIENCY

This section is devoted to reporting the results of computations performed using the schemes introduced in Section 2 and analyzed in Section 3. The stability and convergence rates of the various methods were verified, both as a check on the analysis and to insure that the schemes were correctly coded. The efficiency of the numerical schemes as regards accuracy achieved versus computational effort expended, was also determined. These properties of the schemes were obtained by comparison with the exact solitary-wave solutions of (1.1).

For any value of η and nonzero value of ϵ, equation (1.1a) possesses a one-parameter family of travelling-wave solutions called solitary waves. Taking $\eta = 1$ and $\epsilon > 0$, these special solutions have the form

$$u(x, t) = A \operatorname{sech}^2\left[K\left(x - \frac{1}{2}\right) - \omega t\right], \tag{4.1}$$

where $A > 0$, $K = \frac{1}{2}(A/3\epsilon)^{1/2}$, and $\omega = K(1 + \frac{1}{3}A)$. In the experiments reported here we took $\epsilon = .2058 \times 10^{-4}$ and $A = .22755$, values that correspond to the evolution of water waves in a channel in a regime to which the Korteweg-de Vries equation should apply (cf. [30], [22], and [23]). This choice of parameters corresponds to a solitary wave centered at $x = 1/2$ at $t = 0$ whose height decreases to about 5 percent of its maximum excursion from the undisturbed level at a distance $S \cong .072$ from its peak.

All the numerical experiments reported here were performed in double precision using the FORTRAN Q compiler on an IBM 3031 computer at the University of Tennessee, Knoxville. The smallest number N of spatial intervals used in these calculations was 96, which was easily adequate to resolve the aforementioned solitary wave with either quadratic or cubic splines

Table 2. The errors $E(T)$ and rates of convergence induced in integrating a solitary wave using the Calahan method with $k = 10^{-5}$ and $T = 10^{-3}$.

h^{-1}	Quadratic Splines		Cubic Splines	
	E(T)	rate	E(T)	rate
96	0.8210(-3)		0.1687(-3)	
		3.32		4.71
144	0.2140(-3)		0.2495(-4)	
		3.16		4.37
192	0.8626(-4)		0.7090(-5)	
		3.09		4.32
256	0.3546(-4)		0.2107(-5)	

without spurious oscillations. The error at time t, denoted by $E(t)$, is the normalized L_2-error of the fully discrete approximation at the time level t, that is

$$E(t) = \frac{\|U^n - u(\cdot, t)\|}{\|u^0\|},$$

if $t = nk$. If t is not an integral multiple of k, $E(t)$ is defined by linear interpolation of the values of E at nk and $(n + 1)k$, where $n = [t/k]$. All the integrals occurring in the determination of the spatial L_2-norm of functions as well as integrals arising in L_2-inner products were computed by Gaussian quadrature with 16 nodes on every interval $[x_j, x_{j+1}]$. In all cases, U^0 was taken as the L_2-projection of $u^0(x)$ on S_h^r. The normalizing factor $\|u^0\|$ was about 0.0477.

First the rates of convergence in both space and time of the various schemes were investigated and the regimes in which the existing theoretical results apply were delimited. To verify the order of accuracy of the spatial discretization, the temporal error was effectively set to zero by the choice $k = 10^{-5}$ and use of the third-order Calahan method, and then $h = N^{-1}$ was varied. The errors $E(T)$ at $T = 10^{-3}$ that are observed using quadratic and cubic splines are tabulated in Table 2, along with the implied convergence rates. As usual, the observed rate of convergence determined by two computations with errors E_1 and E_2 corresponding to discretizations h_1 and h_2, respectively, is defined as $\log(E_1/E_2)/\log(h_1/h_2)$. Similar behavior of the spatial errors was found when the temporal integration of the solitary wave was instead effected using the 1-stage Rosenbrock or the 1- or 2-stage DIRK-Newton schemes with $k = 10^{-5}$, $T = 10^{-3}$, and quadratic or cubic splines.

As a test of the accuracy of the temporal integration techniques the solitary wave was numerically integrated holding h fixed at 1/192 for various values of k. A representative sample of the outcome of this test is presented in Table 3, wherein the value in the rate column between adjacent errors E_1 and E_2 is $\log(E_1/E_2)/\log(k_1/k_2)$. Reported here are computations using the 1-stage Rosenbrock method with quadratic splines and the Calahan method with quadratic and cubic splines. As set forth in Section 3, the expected temporal orders of accuracy for these methods are 2, 3, and 3, respectively. If k is not too small, the error induced by the spatial

Table 3. The errors $E(T)$ and rates of convergence induced in integrating a solitary wave using three methods, with $T = 1.0$ and $N = h^{-1} = 192$.

k/h	1-stage Rosenbrock, r = 3		Calahan, r = 3		Calahan, r = 4	
	E(T)	Rate	E(T)	Rate	E(T)	Rate
3	0.6990		0.4225		0.4255	
2	0.4389		0.2445		0.2445	
3/2	0.2852		0.1423		0.1424	
1	0.1414		0.5422(-1)		0.5425(-1)	
		1.91		2.85		2.84
1/2	0.3775(-1)		0.7519(-2)		0.7568(-2)	
		1.98		3.03		3.00
1/3	0.1694(-1)		0.2198(-2)		0.2244(-2)	
		2.00		3.10		3.02
1/4	0.9537(-2)		0.9011(-3)		0.9413(-3)	
		2.02		3.08		3.02
1/6*	0.4207(-2)		0.2584(-3)		0.3107(-3)	
		2.05				3.02
1/8	0.2334(-2)		0.1378(-3)		0.1158(-3)	
		2.10				2.99
1/12	0.9957(-3)		0.1196(-3)		0.3441(-4)	
1/16	0.5300(-3)		0.1231(-3)			
1/20	0.3186(-3)		0.1251(-3)			

*Note the 1/6 was actually 1/5.77 in the last column (Calahan with $r = 4$).

discretization is negligible in comparison with that generated by the temporal discretization. For quadratic splines the exact magnitude of the spatial error may be discerned in the last few entries in column one or column two, whereas for the cubic splines the spatial error was apparently never significant. (Note that the errors in columns two and three are nearly identical for $k/h \gtrsim 1/3$ where the order k^3 temporal error dominates, but that below this value the Calahan method with quadratic splines has errors that are limited by the fixed spatial discretization whilst the use of cubic splines obviates this problem in the presented range of values of k.) As k was decreased below the fixed value of h the expected rates of convergence were indeed evident until the errors were dominated by the spatial discretization. Results similar to those in Table 3 were obtained for different values of h, as well as for the 1- and 2-stage DIRK methods with one Newton iteration per stage. For the latter two temporal discretizations the expected orders of accuracy, 2 and 3, respectively, were observed. We also verified the temporal order of accuracy by holding k/h fixed suitably and decreasing h and k simultaneously. For example, using the Calahan method with $k = h^{4/3}$, it was found that for $h = 1/96$ the error E_1 at time $T = 1$ induced when approximating the evolution of a solitary wave was $0.5013(-2)$, whilst for $h = 1/192$ the error E_2 at time $T = 1$ was $0.3107(-3)$. The resulting rate, $\log(E_1/E_2)/\log(h_1/h_2)$ was 4.012, corresponding very closely to the expected cubic power of k in the asymptotic error estimate. Similar experiments were performed using the other temporal discretizations discussed heretofore.

Another issue that was investigated concerned comparisons regarding accuracy and stability of the DIRK methods with their Rosenbrock counterparts. As mentioned in Section 2 there are no theoretical results regarding stability and convergence of time-stepping via Rosenbrock methods of order greater than two in the context of the KdV equation (or any other nonlinear partial differential equation as far as we know). Moreover, there are conjectures motivated by the theory of approximation of first-order systems of ordinary differential equations that the DIRK methods used here enjoy better stability properties in the context of nonlinear problems than do the Rosenbrock methods. (The latter methods are not B-stable in general—see [31].) In Table 4 are recorded an illuminating set of comparative calculations, namely the error $E(T)$ at time $T = 1$ induced by integrating a solitary wave using 1- and 2-stage Rosenbrock and DIRK methods. The DIRK schemes featured both one or two Newton iterations per stage. The errors in the first group of three columns were obtained with one-stage methods and quadratic splines, so methods having an accuracy of order $k^2 + h^3$. The second group of three columns were obtained using two-stage methods with quadratic splines whilst the third group of three columns were computed with two-stage methods and cubic splines. In each group of three columns, the resulting errors are recorded for the appropriate Rosenbrock and DIRK schemes, the latter with one Newton iteration (DIRK-1N) and two Newton iterations (DIRK-2N) per stage. Observe that within each group of three columns the values of the errors are quite similar for the same value of k/h. This phenomenon persists for smaller values of k/h, not shown in Table 4, but for such values the spatial component of the error figures strongly. In the last row of Table 4 are recorded the average CPU times in seconds per time step for each column of runs. As expected, the DIRK methods are more expensive (consult Table 1) than their Rosenbrock counterparts. On the basis of Tables 3 and 4, it is concluded that as far as accuracy is concerned, at least for relatively small T in the problem at hand, the two-stage, third-order Calahan method holds a clear advantage over the corresponding DIRK methods, whilst there is not much to choose between the second-order Rosenbrock method and its DIRK-1N counterpart.

Another interesting conclusion may be drawn from the data presented in Tables 3 and 4, concerning the variation in the error generated by the fully discrete schemes under consideration as k/h varies. Recall from the discussion in Section 2 that for certain of the schemes used here there is available a rigorous convergence proof. The relevant theorems featured restrictions on the relative size of k and h. Two aspects of our numerical experiments indicate that no stronger restriction than one of the form $k/h \leqslant$ constant should be required to obtain optimal order convergence rates for any of the schemes considered herein. First, in all our calculations it transpires that taking $k/h \leqslant 1/2$ was adequate to guarantee that for small h the observed errors were dominated by the temporal asymptotic rate. On the other hand, no catastrophic instability was ever observed for calculations made with large values of k/h.

The work estimates developed in Section 3 were also subjected to comparison with the

Table 4. The errors $E(T)$ at $T = 1.0$ induced by integrating a solitary wave with $h = 1/96$; a comparison of the Rosenbrock and the DIRK methods.

	$O(k^2+h^3)$, q = 1			$O(k^3+h^3)$, q = 2			$O(k^3+h^4)$, q = 2		
k/h	Rosen.	DIRK 1N	DIRK 2N	Rosen.	DIRK 1N	DIRK 2N	Rosen.	DIRK 1N	DIRK 2N
2	0.8686	0.8700	0.8641	0.5561	0.5704	0.5533	0.5560	0.5704	0.5533
3/2	0.6990	0.6968	0.6881	0.4255	0.4379	0.4233	0.4255	0.4379	0.4233
1	0.4387	0.4346	0.4279	0.2444	0.2502	0.2433	0.2445	0.2503	0.2434
1/2	0.1408	0.1380	0.1369	0.5381(-1)	0.5422(-1)	0.5375(-1)	0.5423(-1)	0.5463(-1)	0.5417(-1)
1/3	0.6528(-1)	0.6369(-1)	0.5585(-1)	0.1708(-1)	0.1715(-1)	0.1710(-1)	0.1759(-1)	0.1765(-1)	0.1760(-1)
CPU secs per step	0.203	0.214	0.268	0.266	0.427	0.536	0.394	0.654	0.820

Table 5. CPU seconds per time step per spatial interval determined by integrating a solitary wave to $T = 1.0$ with $N = 96$ versus the estimated number C_3 of multiplications per time step per spatial interval as presented in Table 1.

Method	One-stage, r = 3			Two-stage, r = 3			Two-stage, r = 4		
	Rosen-brock	DIRK 1N	Ratio	Calahan	DIRK 1N	Ratio	Calahan	DIRK 1N	Ratio
CPU secs $\times 10^3$	2.115	2.229	.949	2.771	4.448	.623	4.104	6.813	.602
C_3	95	98	.969	135	198	.682	233	350	.665
Ratio	44.92	43.97		48.72	44.51		56.77	51.37	

results of actual computational experience. Table 5 provides comparison of the ratio of actual CPU time in milliseconds used per time step by the various schemes to the number N of spatial intervals with the numbers in Table 1 which express the approximate number of multiplications per time step per spatial interval for the same schemes. The actual timings were determined by runs in which $T = 1.0$, $h = 1/96$, and with various time steps. The CPU-seconds per time step were determined as the averages of the timings for all these runs; the result was then divided by N.

The efficacy of the estimates obtained in Section 3 is seen in the relative constancy of the ratios of the number of multiplications with the actual CPU seconds per step per interval (the row labelled "ratio"). The discrepancies owe at least to the facts that C_3 measures only multiplications, and then only those relating to calculations performed on each spatial interval during each time step. Consideration of the relative times per step per interval for the various schemes shows the predictions of Table 1 to be within about 10 percent of the computationally obtained ratios (the columns labelled "ratio"). The general picture that emerged from Table 1 is borne out by the computationally obtained information presented in Table 5.

With these preliminary but important considerations in hand, we turn now to comparing the computational efficiency of the various schemes. Distinguished below are comparisons made by integrating a solitary wave over a relatively short time interval, from $T = 0.0$ to $T = 1.0$, and integrations over longer time scales.

For the approximation of solitary-wave solutions of the KdV equation over a relatively short time interval, it is evident from the results reported in Tables 1, 4 and 5 that the 1-stage Rosenbrock method is always as efficient as its DIRK-1N counterpart and that the Calahan method is more efficient than the two-stage DIRK-1N method. Hence it seemed appropriate to compare only the two Rosenbrock methods. Also, because the one-stage method coupled with cubic splines requires very small time steps in order that the spatial and temporal accuracy be balanced, it was not considered in the detailed comparisons, so leaving three fully discrete schemes, the one-stage Rosenbrock with quadratic splines and the Calahan method with quadratic and cubic splines.

A standard way to compare the relative efficiency of various numerical techniques for one-dimensional evolution equations is the following (cf. [32]). First a suitable measure of the error is fixed; in our context this is the function $E(T)$ given at the beginning of the section. Then an approximate expression for $E(T)$ as a function of k and h is needed. Our tests assure that for k and h small enough the error can be expressed to excellent approximation as

$$E(T) \cong C_1 h^r + C_2 k^p, \tag{4.2}$$

where C_1 and C_2 depend on the particular scheme used, on T, on the solution of (4.1) that is being approximated, and on k and h. It will be taken as valid that C_1 depends only on the degree of splines used and not on the time-stepping method, and that C_2 depends on the time-stepping scheme and not on the degree of splines used in the spatial approximation. This presumption was checked in practice and found to hold to a high degree of approximation. By computing the errors at $T = 1$ for various values of k and h and suitably extrapolating over several experiments, the following values of C_1 and C_2 were determined.

$$C_1 = 1010 \quad \text{if } r = 3, \quad C_1 = 16{,}947 \quad \text{if } r = 4;$$

$$C_2 = 5467 \quad \text{if } p = 2, \quad C_2 = 424{,}073 \quad \text{if } p = 3.$$

These values proved to be quite robust for small values of k and h. A second ingredient needed to compare the efficiency of various schemes is a measure W of the work required to achieve the error $E(T)$. For this we took the number of multiplications that are required to obtain the given error $E(T)$, which to a good approximation is given by C_3NJ, where $J = T/k$ as before and C_3 is a constant that depends on the particular scheme under consideration, and whose values are provided in Table 1. In the special case where $T = 1.0$, the work estimate is

$$W = C_3/kh. \tag{4.3}$$

If the error $E(T)$ is held at a fixed level, then it is a simple calculus problem to determine the values of k and h that minimize the value of W as defined in (4.3). These optimal values, k_{opt} and h_{opt} say, are given by

$$k_{opt} = \left\{\frac{E(T)r}{C_2(r+p)}\right\}^{1/p}, \quad h_{opt} = \left\{\frac{E(T)p}{C_1(r+p)}\right\}^{1/r}. \tag{4.4}$$

The optimal values of k and h, when substituted into (4.3), determine a minimal value W_{min} for the work level required by each method of approximation to obtain the given error level. The various numerical schemes may be compared at each error level on the basis of the associated optimal work estimate W_{min}.

The three, fully discrete schemes were tested using the criteria, just explained, of computing the optimal work estimates. The results for six different levels of error are presented in Table 6. The outcome is not surprising and may be summarized as follows.

(i) The second-order, one-stage Rosenbrock method with quadratic splines is competitive only at low levels of accuracy.
(ii) At accuracies between 10^{-1} and 10^{-3} the Calahan method with quadratic splines appears to be the most efficient combination.
(iii) For higher accuracies the Calahan method with cubic splines holds the advantage.
(iv) The values of h_{opt} and k_{opt} in Table 6 all gave quite acceptable ratios of k/h. In the first column, k_{opt}/h_{opt} ranged from 0.1 to 0.14, in the second column it was always 0.13, whilst in the third column it ranged from 0.13 to 0.05.

Considering the concatenation of approximations that underlies Table 6, it seemed appropriate to devise independent checks of its validity. We used the values of k_{opt} and h_{opt} determined in Table 6 to actually compute an approximation to the solitary wave from $T = 0.0$ to $T = 1.0$ and recorded the error. The values of k_{opt} and h_{opt} were then systematically perturbed whilst holding their product constant, so that the associated value of W was fixed, and the solitary wave again approximated up to $T = 1.0$ using the perturbed values of the parameters k and h. In all the cases tested, the error associated to the perturbed values was larger than that obtained using the theoretically determined optimal values. A typical example is recounted in Table 7, which was constructed using the Calahan method with quadratic splines and $E(T) = 10^{-3}$. The

Table 6. The values of h_{opt}, k_{opt}, and W_{min}, respectively, for three, fully discrete schemes for six given levels of error $E(T)$ where $T = 1.0$ and the error is that generated by using the scheme to approximate the solitary-wave solution (4.1) of the KdV equation.

Method / Error level E(T)	One-stage Rosenbrock with quadratic splines	Calahan with quadratic splines	Calahan with cubic splines
10^{-1}	0.3409(-1) 0.3313(-2) 8.4×10^5	0.3672(-1) 0.4904(-2) 7.5×10^5	0.3988(-1) 0.5127(-2) 1.14×10^6
10^{-2}	0.1582(-1) 0.1048(-2) 5.7×10^6	0.1704(-1) 0.2276(-2) 3.4×10^6	0.2243(-1) 0.2380(-2) 4.4×10^6
10^{-3}	0.7344(-2) 0.3313(-3) 3.9×10^7	0.7911(-2) 0.1056(-2) 1.62×10^7	0.1261(-1) 0.1105(-2) 1.67×10^7
10^{-4}	0.3409(-2) 0.1048(-3) 2.7×10^8	0.3672(-2) 0.4904(-3) 7.5×10^7	0.7091(-2) 0.5127(-3) 6.4×10^7
10^{-5}	0.1582(-2) 0.3313(-4) 1.8×10^9	0.1704(-2) 0.2276(-3) 3.5×10^8	0.3988(-2) 0.2380(-3) 2.5×10^8
10^{-6}	0.7344(-3) 0.1048(-4) 1.2×10^{10}	0.7911(-3) 0.1056(-3) 1.6×10^9	0.2243(-2) 0.1105(-3) 9.4×10^8

Table 7. A check of an entry in Table 6. The error associated with several values of k and h using the Calahan method with quadratic splines to integrate the solitary wave. Here $T = 1.0$ and $E(T) = 10^{-3}$.

N = 1/h	J = 1/k	E(T)	CPU secs
126 (opt)	948 (opt)	0.5798(-3)	345
150	796	0.7731(-3)	340
100	1194	0.1161(-2)	333

last column reports the actual number of CPU seconds that the run required using the stated values of k and h.

Experiments using the solitary-wave initial data were also performed over longer time intervals from $T = 0.0$ to $T = 5.0$. The computed approximate solutions were compared in several ways with the exact solution given in (4.1). In addition to the normalized L_2-error $E(T)$, we kept track of amplitude, phase, and shape errors (cf. [23]). The *shape error* E^n is defined for each time step $n = 0, 1, \ldots, J$ as follows. Fix n and consider the quantity

$$\xi^2(\tau) = \frac{\int_0^1 [u(x, \tau) - U^n(x)]^2 \, dx}{\int_0^1 u^2(x, 0) \, dx}, \tag{4.5}$$

where $u(x, \tau)$ is given in (4.1) and U^n is the computed solution at the time step n. Let τ^* denote the value of τ near nk where $\xi^2(\tau)$ takes its minimum value. If U^n resembles a solitary wave in shape, it follows that τ^* is well defined. Then $E^n = \xi^2(\tau^*)$ measures by how far the computed solution differs from the original solitary wave as regards its shape, as measured by the normalized L_2 norm. The *phase error* P^n at any time step n, $0 \leq n \leq J$, is defined to be $nk - \tau^*$. It measures the error in the position at which the wave is located. The *amplitude error* A^n is defined to be $(A - U^n_{\max})/A$ where A is as in (4.1) and $U^n_{\max}$ is the maximum value of $U^n(x)$.

In our computer program the shape, phase, and amplitude errors were determined as follows. The quantity $\xi^2(\tau)$ was minimized by finding a zero of its derivative using Newton's method.

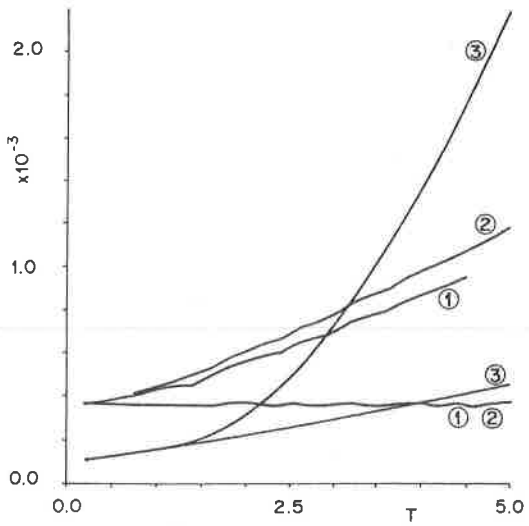

Fig. 1. The L_2-error $E(t)$ (curves 1, 2, and 3) and the shape error E^n (curve 1′, 2′, and 3′) resulting from the approximation over the time interval from $T = 0.0$ to $T = 5.0$ of a solitary-wave solution, as specified in (4.1), of the KdV equation. Curves 1 and 1′ were obtained using the DIRK-1N scheme with quadratic splines and $N = 128$, $J = 15{,}200$; curves 2 and 2′ were obtained using the (1, 2)-Rosenbrock scheme with quadratic splines and $N = 128$, $J = 15{,}200$; and curves 3 and 3′ were obtained using the Calahan method with cubic splines and $N = 192$, $J = 7{,}250$.

Taking $\tau_0 = nk$, a sequence $\{\tau_j\}_{j=0}^{K}$ is generated by

$$\tau_{j+1} = \tau_j - \frac{\frac{d}{d\tau}\xi^2(\tau_j)}{\frac{d^2}{d\tau^2}\xi^2(\tau_j)}, \quad j = 0, 1, \ldots. \tag{4.6}$$

Of course, up to a multiplicative constant,

$$\frac{d}{d\tau}\xi^2(\tau_j) = 2\int_0^1 [u(x, \tau_j) - U^n(x)]u_x(x, \tau_j)\,dx,$$

and this quantity is actually calculated using a Riemann sum with 1024 equidistant points on [0, 1]. A similar remark applies to $d^2\xi^2(\tau)/d\tau^2$. The iteration (4.6) is terminated when $|\tau_{j+1} - \tau_j| < 10^{-10}$ and τ_{j+1} is then declared to be τ^*. The shape error E^n is then computed using the same subroutine that approximates the normalized L_2-error $E(T)$. Once τ^* has been approximately determined as τ_{j+1}, then P^n is given as $nk - \tau_{j+1}$. The quantity $U^n_{\max}$ is simply taken to be the maximum of U^n, a number that is easily determined.

The two Rosenbrock methods and their DIRK-1N counterparts were compared first. The normalized L_2-errors and the shape errors for the one-stage versions of these methods with quadratic splines are plotted versus time in Fig. 1 (curves 1 and 2). The plotted data was obtained taking $N = 128$ and $J = 15{,}200$, corresponding therefore to $h \cong 0.781 \times 10^{-2}$ and $k \cong 0.329 \times 10^{-3}$, values that are very close to optimal for these methods to achieve $E(1) = 10^{-3}$. The shape errors for both methods are practically identical and remain sensibly constant in time, whereas the total L_2-error appears to increase linearly with time. The DIRK-1N method has the smaller L_2-error of the two, with the difference between the two errors being some 11 percent at $T = 5.0$. In Fig. 2 the phase errors for both methods are plotted for the same run. Both techniques display linearly growing phase errors, and again the DIRK-1N method holds a small advantage with the difference in the two errors reaching about 14 percent at $T = 5.0$. The relative amplitude errors were not plotted as they were essentially identical for both methods and remained very small, fluctuating in sign with a maximum value of about 0.8×10^{-3}. The DIRK-1N method was slightly more expensive, requiring 0.285 CPU seconds per time step compared with 0.270 CPU seconds for the one-stage Rosenbrock method. (The one-stage DIRK-2N was also tried on the same time interval with results nearly identical to those obtained with the one-stage DIRK-1N, but at a cost of 0.354 CPU seconds per time step.)

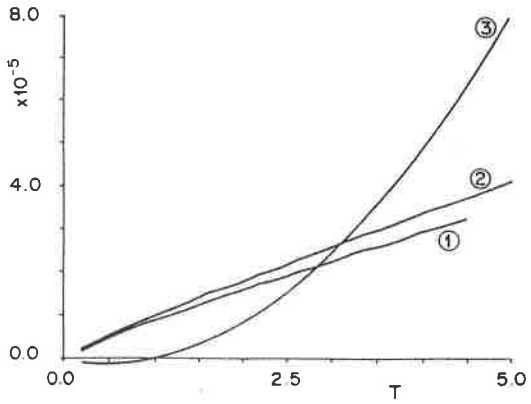

Fig. 2. The phase error P^n resulting from the approximation over the time interval from $T = 0.0$ to $T = 5.0$ of a solitary-wave solution, as specified in (4.1), of the KdV equation. Curve 1 was obtained using the DIRK-1N scheme with quadratic splines and $N = 128$, $J = 15{,}200$; curve 2 was obtained using the (1, 2)-Rosenbrock scheme with quadratic splines and $N = 128$, $J = 15{,}200$; and curve 3 was obtained using the Calahan method with cubic splines and $N = 192$, $J = 7{,}250$.

The results for the two one-stage methods point to the interesting possibility that these schemes may possess numerical solitary waves. That is, the numerical schemes themselves may have exact discrete solutions that are travelling waves with a shape and a phase speed very near to that of the solitary-wave solution of the KdV equation. This interpretation is consistent with the constancy of the shape error. In this light, the linear growth of the relative L_2-error is attributable almost entirely to the small difference between the discrete and the continuous phase speeds. Further support for this view is gleaned from the observation that the difference in the L_2-errors between the one-stage DIRK-1N and Rosenbrock methods corresponds very closely to the difference in their approximation to the phase speed of the solitary wave.

Turning to the two-stage, third-order time-stepping techniques with cubic splines, the Calahan method was compared with its two-stage DIRK-1N counterpart in a run with $N = 128$ and $J = 4700$. With $T = 5.0$, this corresponds to $k \cong 1.06 \times 10^{-3}$ while $h \cong 0.781 \times 10^{-2}$ as before. Unlike the situation that arose with one-stage methods, there was no practical difference in the various errors generated by the two methods. For example, at $T = 5.0$ the difference between the normalized L_2-errors was about 0.1 percent, the difference between the shape errors was about 0.06 percent, the difference in the phase errors was about 0.07 percent, and the relative amplitude errors were identical. For the present problem, the cost of the Calahan method was some 0.366 CPU seconds per step as compared with 0.427 CPU seconds per step for the two-stage, DIRK-1N method, and in consequence the latter method was excluded from further consideration in longer-time experiments.

The stage was now set for a confrontation between the Calahan method with the one-stage techniques. The Calahan method was run on our standard solitary wave to $T = 5.0$ with $N = 192$ and $J = 7250$, so $h \cong 0.521 \times 10^{-2}$ and $k \cong 0.69 \times 10^{-3}$. The ratio k/h for this experiment was about 0.13, so approximately optimal at $T = 1.0$ for a normalized L_2-error level between 10^{-3} and 10^{-4}. These particular values of k and h were chosen so that the total processing time for this experiment, some 4.21×10^3 CPU seconds, was about the same as the total processing time, 4.10×10^3 CPU seconds, for the one-stage Rosenbrock method in the run described earlier and reported on in Figs. 1 and 2. The errors associated with the run using the Calahan method are also recorded in Figs. 1 and 2 (curves 3). The relative L_2-error increases nearly linearly for $T \leqslant 1.0$, but then shows a superlinear growth, reaching 2.191×10^{-3} at $T = 5.0$, about twice the value generated by the one-stage Rosenbrock method. The shape error showed a slow but definite linear growth throughout the time interval, overtaking the constant shape error of the second-order methods at about $T = 4.0$. The phase error is shown in Fig. 2. It is initially negative, but by time 5.0 is positive and about double that of the one-stage Rosenbrock scheme. The relative amplitude error was small, but positive, reflecting the dissipativity of the third-order scheme. It did not increase appreciably on this time interval, and its maximum observed value was 1.2×10^{-3}.

These comparisons afford several conclusions. First, the third-order schemes surely do not possess numerical solitary waves, as the shape errors continue to grow. Secondly, while the third-order Calahan method was superior to all other schemes tested over shorter time intervals, the second-order schemes appear to be more efficient over longer intervals due to the linear increase in their phase errors and their constant shape error. In some sense, the second-order methods seem to capture important qualitative features of the overlying differential equation not shared by the higher-order schemes, and in long runs this may be more important than higher-order convergence rates.

This last remark may be amplified a little by consideration of how the first few integral invariants of the KdV equation respond to the various numerical schemes. Considered here are

$$I_1 = \int_0^1 u(x, t)\, dx, \quad I_2 = \int_0^1 u^2(x, t)\, dx, \quad \text{and } I_3 = \int_0^1 [u^3(x, t) - 3\epsilon u_x^2(x, t)]\, dx.$$

It is straightforward to verify that for smooth solutions of the KdV equation which are periodic of period 1, I_1, I_2, and I_3 are independent of time (cf. [1]). It is also easy to see that all the schemes considered herein preserve I_1 up to round-off error. Hence attention is restricted to the variation of I_2 and I_3. The schemes represented in Figs. 1 and 2 were used to approximate the solitary-wave solution (4.1) of the KdV equation and the values of I_2 and I_3 corresponding to

Table 8. The variation with time of the functionals I_2 and I_3 for the (1, 2) DIRK and Rosenbrock methods and the Calahan method for the integrations reported in Figs. 1 and 2.

	I_2		I_3	
	(1, 2) methods	Calahan	(1, 2) methods	Calahan
t = 0	.227440(-2)	.227440(-2)	.310516(-3)	.310523(-3)
t = 1	.227440(-2)	.227405(-2)	.310516(-3)	.310444(-3)
t = 5	.227440(-2)	.227266(-2)	.310516(-3)	.310128(-3)

this approximation were computed and recorded at various times. A typical sample of the outcome of this experiment is provided in Table 8 where the values to six digits of I_2 and I_3 are recorded at $T = 0.0$, $T = 1.0$, and $T = 5.0$. The values of I_2 and I_3 at $T = 0.0$ are obtained from $u_h^0 = Pu^0$ rather than from u^0 itself.

The results obtained from these experiments are revealing. The two second-order schemes were indistinguishable, and so are reported as a group. Both of the second-order methods appeared to conserve I_2 and I_3, whereas these functionals suffered a small but steady decrease when the third-order Calahan time-stepping was used (with either quadratic or cubic splines). The existence and stability theory for solitary waves in a broad class of continuous systems like the KdV equation relies upon a pair of conserved quantities analogous to I_2 and I_3 (cf. [33]). Thus the results in Table 8 may be interpreted as further evidence that the second-order numerical schemes considered here possess travelling-wave solutions analogous to solitary waves, and that the more accurate Calahan method has long-term effects which are not reflections of aspects of the partial differential equation, but instead reflect the numerical modeling.

We digress for a moment to discuss in more detail the work of Taha and Ablowitz[6]. They have provided a careful, comparative view of a wide range of techniques for approximating solutions of the KdV equation. In one set of experiments, they took the equation in the form,

$$u_t + 6uu_x + u_{xxx} = 0, \tag{4.7a}$$

with solitary-wave initial data,

$$u(x, 0) = A \operatorname{sech}^2(kx), \tag{4.7b}$$

where $A = 2k^2$. The solution of (4.4) is written explicitly as

$$u_s(x, t) = A \operatorname{sech}^2(kx - \omega t) \tag{4.8}$$

where $\omega = 4k^3$. For several values of A the solution of the initial-value problem (4.7) was approximated over the time interval [0, 1] by eight different, uniform mesh, fully discrete schemes (see [6, Tables I, II, and III]). The accuracy achieved was measured by the quantity,

$$e(t) = \max_i \{|u_s(x_i, t) - U_i^j|\},$$

where $j = t/\Delta t$ and U_i^j denotes the relevant discrete approximation to u_s at the point $(x, t) = (i\Delta x, j\Delta t)$. The error, $e(1)$, was in each case specified to be at most a given value and Δt and Δx were adjusted to yield about the smallest CPU time a particular scheme needed to achieve this level of error. All the methods examined in [6] were coded in PL1 and run on an IBM 4341 computer using the optimizing compiler.† The best performances were obtained using a local scheme proposed by Taha and Ablowitz, though the pseudo-spectral scheme of Fornberg and Whitham[11] was also quite competitive (see, again, [6]).

It seemed appropriate to make a direct comparison of the results obtained by Taha and Ablowitz with those obtainable with the schemes studied herein. At the time these comparisons were made, we were working on an IBM 3081 instead of the IBM 3031 that was available

†The authors thank Professors Taha and Ablowitz for this information, and for several helpful discussions regarding their work.

during the rest of our study. In consequence, account must be taken of the differing machines utilized in generating the respective approximate solutions. As both machines are standard mainframe computers, and as the codes are both of the same numerically intense character, a pretty accurate constant of proportionality is known to relate the speeds of execution on the two machines. For the IBM 4341 used by Taha and Ablowitz versus the IBM 3081 used by us in the present comparisons, this constant is about six.

Three cases are considered, namely, $A = 1.0$, $A = 2.0$, and $A = 4.0$ in (4.7b). We used the Calahan method with quadratic splines to approximate (4.8) by integrating the initial-value problem (4.7). Considering the relatively large errors that Taha and Ablowitz specified, this choice seemed to be dictated by the results reported in Table 6. The outcome of our runs were compared with the best results obtained in [6]. For $A = 1.0$ the best performance computed on the IBM 4341 and reported in [6] was an error of 0.00173 at $t = 1.0$ in 7 CPU seconds. Using $N = 96$ and $J = 25$, we obtained an error of 0.00178 on the IBM 3081 in 1.02 CPU seconds, a time that corresponds to about 6 CPU seconds on the IBM 4341. For $A = 2.0$ the best performance given in [6] was an error of 0.00332 at $t = 1.0$ which was obtained in 23 CPU seconds. Taking $N = 144$ and $J = 45$, an error of 0.00288 was obtained on the IBM 3081 in 2.81 CPU seconds, a time that corresponds to about 17 CPU seconds on the IBM 4341. Finally, for $A = 4.0$ the error level $e(1)$ achieved in [6] was 0.01747 in 140 CPU seconds on the IBM 4341. We took $N = 172$ and $J = 140$ and found an error $e(1)$ of 0.0171 at $t = 1.0$ in 10.2 CPU seconds, so corresponding to some 61 CPU seconds on an IBM 4341.

Thus it seems that even for relatively coarse calculations on comparatively small solutions such as those reported in [6], the best of the schemes proposed here are competitive with others in the literature. For larger amplitudes, or for smaller values of specified accuracy the trend appears to favor our techniques, though the data available are too sparse to justify any categorical conclusion in this direction.

5. DISCUSSION

A range of fully discrete, numerical techniques for the approximation of solutions of the KdV equation has been implemented and tested, especially as regards stability, accuracy, and efficiency. Because solutions of the KdV equation that are relevant to wave phenomena are smooth, it was appropriate to consider Galerkin-type spatial approximations based on smooth splines. The temporal discretizations used were diagonally-implicit Runge-Kutta methods and Rosenbrock methods, mostly of second and third order. When these spatial and temporal discretizations were combined, there resulted schemes that are stable and accurate even with relatively large time steps. In addition to verifying these general attributes for each of the competing schemes, optimal values of k and h required to achieve given error bounds were determined. This latter information made it possible to give an accurate assessment of the efficiency of the various schemes. In what follows in this Section, we summarize the substantive conclusions derived from this study and present an interestng sample computation that relies on the methods introduced heretofore.

If the aim is to approximate solutions of the initial-value problem (1.1) over a relatively short period of time, our recommendation depends on the accuracy desired. If relatively low accuracy suffices, then the Calahan method (a third-order Rosenbrock method) with quadratic splines seems to be most efficient. If higher accuracy is desired, however, it is warranted to shift to cubic splines whilst keeping the third-order Calahan time-stepping technique. The success of the Calahan method is especially useful as regards the prospect of comparing the model's predictions with data collected in the laboratory or field, where the use of comparatively large time steps is very convenient (cf. [34] and [22]).

For longer time spans our experiments indicate that the second-order accurate Rosenbrock and Runge-Kutta methods, both of which may be thought of as nonlinear versions of the classical Crank-Nicolson scheme, are preferable to the higher-order techniques, both as far as accuracy is concerned, and as regards capturing the general structure of solutions of the KdV equation. The latter point is especially potent when investigations into the asymptotic structure of solutions for large time is in question.

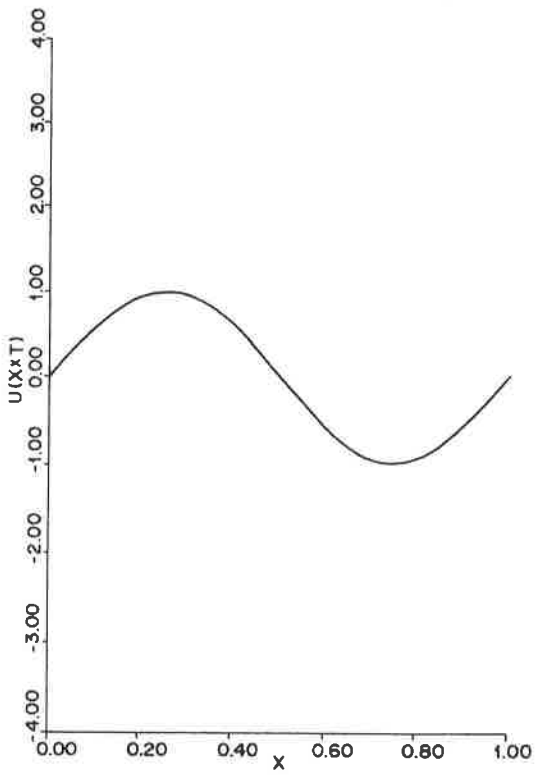

Fig. 3(a). $t = 0.0$.

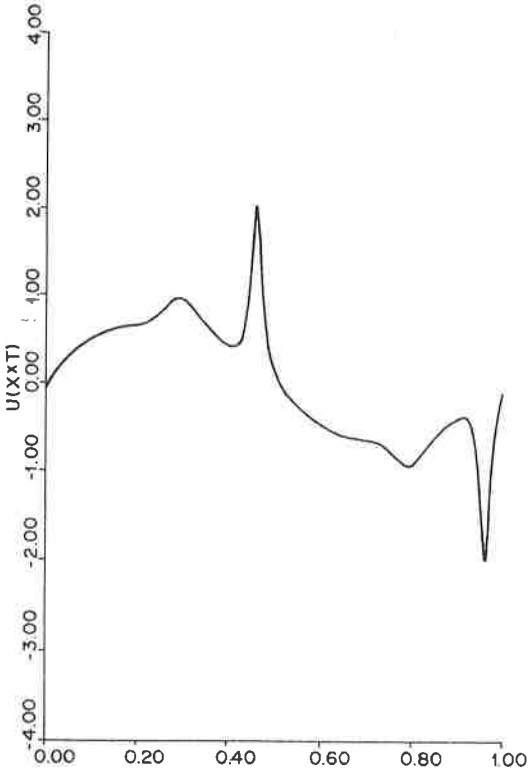

Fig. 3(b). $t = 0.3$.

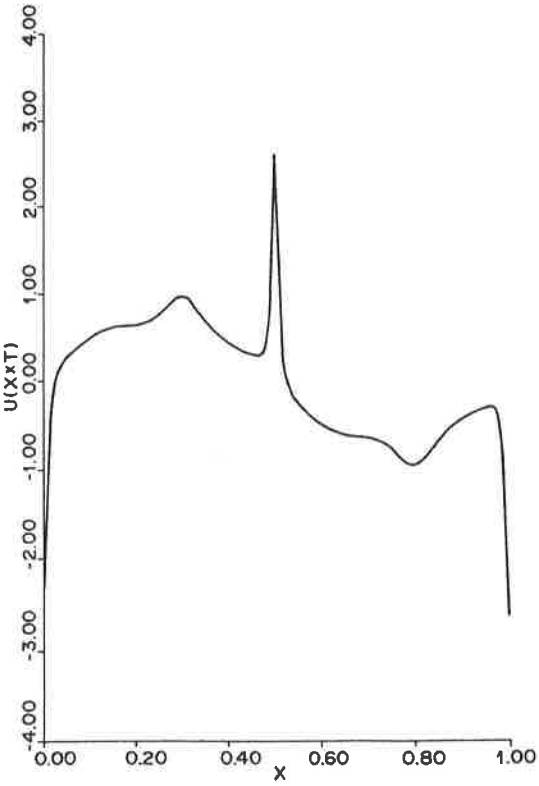

Fig. 3(c). $t = 0.31908$.

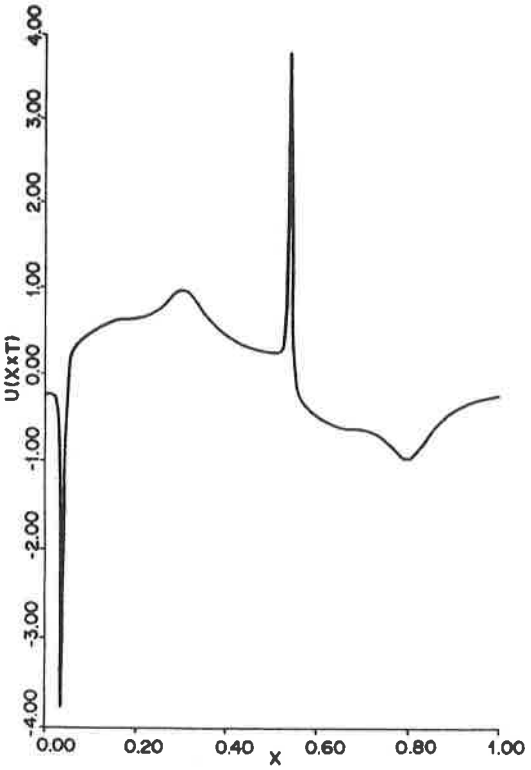

Fig. 3(d). $t = 0.32511$.

Fig. 3. The evolution under equation (5.1) with $p = 4$ and $\nu = 0$ of a sine-wave initial profile as seen via the Calahan method with cubic splines and $N = 1{,}024$. The time step was adaptively determined. (a) $t = 0.0$, (b) $t = 0.3$, (c) $t = 0.31908$, (d) $t = 0.32511$, (e) $t = 0.32594$, (f) $t = 0.32604$.

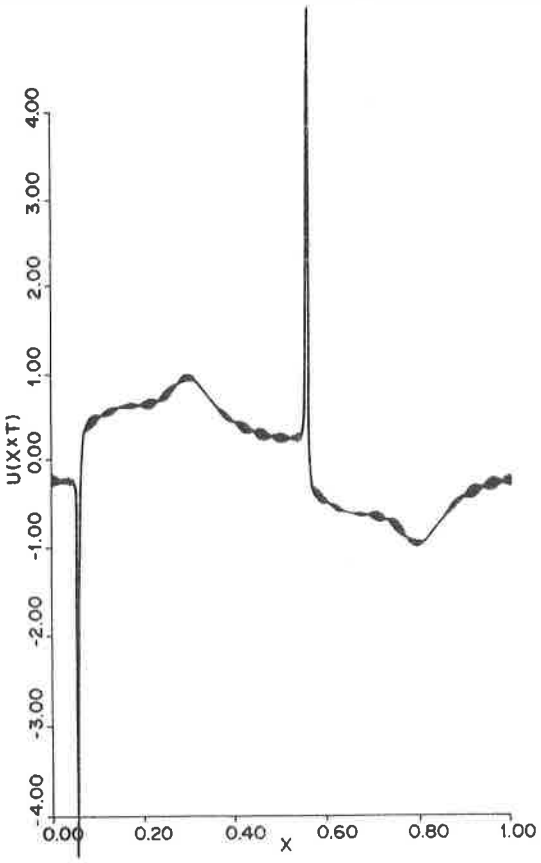

Fig. 3(e). $t = 0.32594$.

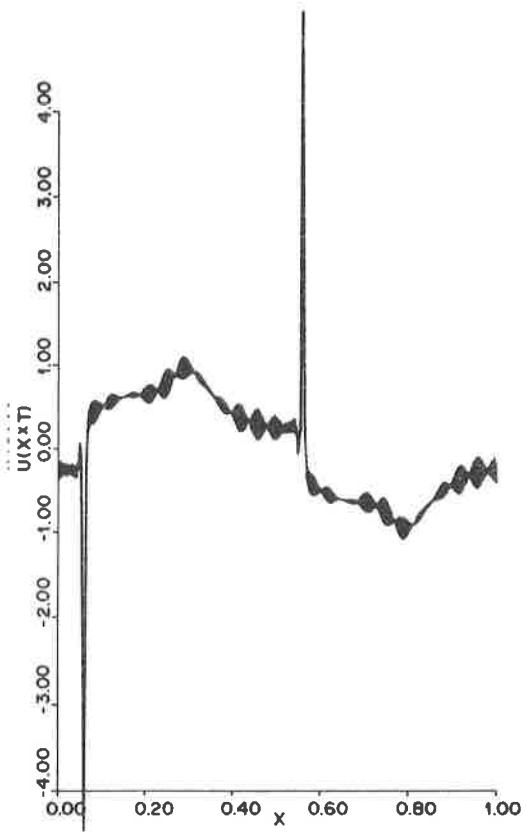

Fig. 3(f). $t = 0.32604$.

If interest is centered on some delicate, detailed aspect of a solution, whether or not it manifests itself in a relatively short time, then the Calahan method with cubic splines is again the superior choice among those considered herein. An example illustrating this remark will be presented presently.

In a subsequent paper, issues of all the above types will be examined for the KdV equation in the context of the general class of equations of the form,

$$u_t + u^p u_x - \nu u_{xx} + u_{xxx} = 0, \tag{5.1}$$

where $\nu \geq 0$ and p is a positive integer. It is worth note, therefore, that the conclusions of the present study go over intact for the initial-value problem for equation (5.1), so providing a basis for this forthcoming work.

In closing, the presentation of an example is perhaps merited in which a detailed study such as that given here is useful. An issue that is mathematically rather interesting arises for equation (5.1) with $\nu = 0$ and $p \geq 4$. The standard theory for the initial-value problem of (5.1) insures that there exists a unique smooth solution u corresponding to given, smooth initial data u^0, at least over some time interval $[0, T^*)$, where $T^* = T^*(u^0) > 0$ (cf. Kato[2]). If $p < 4$, then T^* may be taken to be $+\infty$, because of certain *a priori* bounds that are available in this case. However, the question of whether or not T^* can be taken to be $+\infty$ in case $p \geq 4$ is open, save for the case in which u^0 is sufficiently small in L_2-norm (see Strauss[35]). In the particular case $p = 4$, Weinstein[36] has characterized the singularity that must form if the solution does indeed lose smoothness at some finite time. Deciding whether or not a solution blows up in finite time is a rather delicate issue, both analytically and numerically, and so following our own advice, this point was studied using the Calahan method with cubic splines. In Fig. 3 we present the outcome of an example numerical experiment performed on (5.1) with $p = 4$, $\nu = 0$, and smooth initial data. The singularity that apparently forms at about $t = 0.326$ required the higher accuracy scheme in order that it be properly resolved. In addition,

as the spatial gradients grew, we found it necessary to refine k in order not to simply step over the singularity. In the elucidation of this phenomenon, the preliminary study of the numerical scheme as reported here was invaluable, both technically and as a means of generating confidence in the outcome of the simulation. Other numerical evidence points in the same direction as that displayed in Fig. 3, and thereby it is tentatively concluded that solutions of the initial-value problem (5.1) do not necessarily remain smooth for all time.

REFERENCES

1. J. L. Bona and R. Smith, The initial-value problem for the Korteweg-de Vries equation. *Philos. Trans. Roy. Soc. London, Ser. A* **278**, 555–604 (1975).
2. T. Kato, On the Cauchy problem for the (generalized) Korteweg-de Vries equation. *Studies in Appl. Math., Advances in Mathematics Supplementary Studies* **8**, Academic Press, NY, 93–130 (1983).
3. N. J. Zabusky, Computation: Its role in mathematical physics innovation. *J. Comput. Phys.* **43**, 195–249 (1981).
4. I. S. Greig and J. Ll. Morris, A hopscotch method for the Korteweg-de Vries equation. *J. Comput. Phys.* **20**, 64–80 (1976).
5. J. M. Sanz-Serna, An explicit finite-difference scheme with exact conservation properties. *J. Comput. Phys.* **47**, 199–210 (1982).
6. T. R. Taha and M. J. Ablowitz, Analytical and numerical aspects of certain nonlinear evolution equations III. Numerical, Korteweg-de Vries equation. *J. Comput. Phys.* **55**, 231–253 (1984).
7. A. C. Vliegenthart, On finite-difference methods for the Korteweg-de Vries equation. *J. Engrg. Math.* **5**, 137–155 (1971).
8. N. J. Zabusky and M. D. Kruskal, Interaction of 'solitons' in a collisionless plasma and the recurrence of initial states. *Phys. Rev. Lett.* **15**, 240–243 (1965).
9. K. Abe and O. Inoue, Fourier expansion solution of the Korteweg-de Vries equation. *J. Comput. Phys.* **34**, 202–210 (1980).
10. T. F. Chan and T. Kerkhoven, Fourier methods with extended stability intervals for the Korteweg-de Vries equation. *SIAM J. Numer. Anal.* **22**, 441–454 (1985).
11. B. Fornberg and G. B. Whitham, A numerical and theoretical study of certain nonlinear wave phenomena, *Philos. Trans. Roy. Soc. London, Ser. A* **289**, 373–404 (1978).
12. J. M. Hyman, The evolution of almost periodic solutions of the Korteweg-de Vries equation. *Rocky Mntn. J. of Math.* **8**, 95–104 (1978).
13. J. E. Pasciak, Spectral methods for a nonlinear initial value problem involving pseudo-differential operators, *SIAM J. Numer. Anal.* **19**, 142–154 (1982).
14. F. Tappert, Numerical solutions of the Korteweg-de Vries equation and its generalizations by the split-step Fourier method, in *Nonlinear Wave Motion* (ed. A. C. Newell), Lectures in Appl. Math., v. 15, 215–216 AMS, Providence, RI, (1974).
15. M. E. Alexander and J. Ll. Morris, Galerkin methods applied to some model equations for nonlinear dispersive waves. *J. Comput. Phys.* **30**, 428–451 (1979).
16. D. N. Arnold and R. Winther, A superconvergent finite element method for the Korteweg-de Vries equation. *Math. Comp.* **38**, 23–36 (1982).
17. G. A. Baker, V. A. Dougalis and O. A. Karakashian, Convergence of Galerkin approximations for the Korteweg-de Vries equation. *Math. Comp.* **40**, 419–433 (1983).
18. V. A. Dougalis and O. A. Karakashian, On some high order accurate fully discrete Galerkin methods for the Korteweg-de Vries equation. *Math. Comput.* **45**, 329–345 (1985).
19. J. M. Sanz-Serna and I. Christie, Petrov-Galerkin methods for nonlinear dispersive waves. *J. Comput. Phys.* **39**, 94–102 (1981).
20. L. B. Wahlbin, A dissipative Galerkin method for the numerical solution of first order hyperbolic equations, in *Mathematical Aspects of Finite Element Methods in Partial Differential Equations* (ed. C. de Boor) 147–169, Academic Press, NY (1974).
21. R. Winther, A conservative finite element method for the Korteweg-de Vries equation. *Math. Comp.* **34**, 23–43 (1980).
22. J. L. Bona, W. G. Pritchard and L. R. Scott, An evaluation of a model equation for water waves. *Philos. Trans. Roy. Soc. London, Ser. A* **302**, 457–510 (1981).
23. J. L. Bona, W. G. Pritchard and L. R. Scott, Numerical schemes for a model for nonlinear, dispersive waves. *J. Comput. Phys.* **60**, 167–186 (1985).
24. V. Thomee and B. Wendroff, Convergence estimates for Galerkin methods for variable coefficient initial value problems. *SIAM J. Numer. Anal.* **11**, 1059–1068 (1974).
25. R. Alexander, Diagonally implicit Runge-Kutta methods for stiff O.D.E.'s. *SIAM J. Numer. Anal.* **14**, 1006–1021 (1976).
26. M. Crouzeix, Sur l'approximation des équations différentielles opérationelles linéaires par des méthodes de Runge-Kutta. Thése, Université de Paris VI (1975).
27. S. P. Nørsett, One-step methods of Hermite type for numerical integration of stiff systems. *BIT* **14**, 63–77 (1974).
28. G. W. Gear, *Numerical initial value problems in ordinary differential equations*, Prentice-Hall, Englewood Cliffs, NJ (1971).
29. D. A. Calahan, A stable, accurate method of numerical integration for nonlinear systems. *Proc. I.E.E.E.* **56**, 744 (1968).
30. J. L. Bona, W. G. Pritchard and L. R. Scott, Solitary-wave interaction. *Phys. Fluids* **23**, 438–441 (1980).
31. K. Burrage and J. C. Butcher, Stability criteria for implicit Runge-Kutta methods. *SIAM J. Numer. Anal.* **16**, 46–57 (1979).

32. B. Swartz and B. Wendroff, The relative efficiency of finite difference and finite element methods. I: Hyperbolic problems and splines. *SIAM J. Numer. Anal.* **11**, 979–993 (1974).
33. T. B. Benjamin, J. L. Bona and D. K. Bose, Solitary-wave solutions for some model equations for waves in nonlinear dispersive media. Proceedings of the IUTAM/IMU Symposium on applications of methods of functional analysis to problems in mechanics, Marseille, Sept. 1975 (ed. P. Germain and B. Nayroles), *Springer Lecture Notes in Mathematics* **503**, 207–218 Springer-Verlag, NY (1976).
34. B. Boczar-Karakiewicz and J. L. Bona, Wave-dominated shelves: a model of sand-ridge formation by progressive infragravity waves. In *The Canadian Society for Petroleum Geology Memoir on Shelf Sands and Sandstones*.
35. W. A. Strauss, Dispersion of low-energy waves for two conservative equations. *Arch. Rat. Mech. Anal.* **55**, 86–92 (1974).
36. M. I. Weinstein, On the structure and formation of singularities in solutions to nonlinear dispersive evolution equations. To appear.