A boundary value problem for the KdV equation: Comparison of finite-difference and Chebyshev methods

Jan Ole Skogestad*, Henrik Kalisch

Department of Mathematics, University of Bergen, Johannes Brunsgate 12, 5008 Bergen, Norway

Available online 18 June 2009

Abstract

Solutions of a boundary value problem for the Korteweg–de Vries equation are approximated numerically using a finite-difference method, and a collocation method based on Chebyshev polynomials. The performance of the two methods is compared using exact solutions that are exponentially small at the boundaries. The Chebyshev method is found to be more efficient.

© 2009 IMACS. Published by Elsevier B.V. All rights reserved.

Keywords: KdV equation; Boundary-value problem; Finite-difference method; Chebyshev method

1. Introduction

The purpose of this note is the comparison of two numerical schemes for the discretization of the KdV equation in the presence of boundaries. Since it was first derived by Boussinesq [3] and Korteweg and de Vries [22] as a model for surface water waves in a channel, the Korteweg–de Vries (KdV) equation

$$u_t + u_x + uu_x + u_{xxx} = 0 \quad (1.1)$$

has been useful as an approximate model equation in a number of important physical situations. Applications include the study of internal waves in coastal waters, waves in plasma physics, flow in blood vessels and magma flow and conduit waves [7,10,20].

For dispersive equations like the KdV equation, Fourier-collocation schemes have been popular [16,21,24]. However, a disadvantage of the use of the Fourier basis is the confinement to periodic boundary conditions. For studies of solutions that are exponentially small, such as solitary waves, this is not a problem, as one may take a big enough periodic domain so that the tails of the solutions are below machine precision throughout the computation. However, in some situations, one may want to consider problems involving boundary conditions, for instance in the study of an undular bore. In this case, one may turn to finite-difference approximations, as well as finite-element or finite-volume methods [1,6,30]. In the realm of spectral methods, one may use a collocation method based on a certain family of polynomials. There have been relatively few studies pursuing this approach. Pavoni [25] defined a Chebyshev-collocation method. Heinrichs [18] put forward a scheme for the adjoint of the linearized problem, and also presented a numerical investigation of the eigenvalues of the spectral differentiation operator. There are also recent articles by Ma and Sun [23] and by Shen [26] presenting spectral methods for third-order problems with boundary conditions. In the present article, the focus

* Corresponding author.

E-mail addresses: jan.skogestad@math.uib.no (J.O. Skogestad), henrik.kalisch@math.uib.no (H. Kalisch).

will be on a Chebyshev-collocation method, with the concrete aim of comparing it with a simple finite-difference method.

A well known advantage of a spectral method is that it achieves high accuracy with relatively fewer spatial grid points when compared with a finite-difference method. On the other hand, spectral methods typically give rise to full matrices, partially negating the gain in efficiency due to the fewer number of grid points. In principle, the Chebyshev basis is potentially more useful than some other polynomial bases because it sometimes grants the use of the fast Fourier transform, thereby obviating the negative impact of full differentiation matrices. On the other hand, the Chebyshev-collocation points are concentrated near the boundary, thus forcing the use of a rather small time step. Weighing the advantages and drawbacks of the Chebyshev method one may ask the question of whether the Chebyshev method is actually able to outdo a simple finite-difference method in efficiency. Moreover, in the present case, it turns out that the use of the fast Fourier transform is not possible. Since all of the above seem to weigh heavily against the Chebyshev projection, our findings may come as a surprise. But indeed, as we shall see in Section 4, despite all the negatives, the Chebyshev method is more efficient than the finite-difference method.

Since the equation is of third order in the spatial derivatives, it is natural that three boundary conditions be required. If two Dirichlet conditions are used, then the question arises whether to place a Neumann condition on the left or on the right. Here, we choose to place a homogeneous Neumann condition on the right boundary. A motivation for this is given for example by Holmer [19]. In fact, using the linear problem, it may be shown that uniqueness holds only for the problem with two homogeneous Dirichlet conditions and a homogeneous Neumann condition on the right hand side. Consequently, the problem to be studied here is as follows:

$$\begin{cases} u_t + uu_x + u_{xxx} = 0, & x \in [-l, l], t \geq 0, \\ u(x, 0) = f(x), \\ u(-l, t) = 0, \\ u(l, t) = 0, \\ u_x(l, t) = 0. \end{cases} \quad (1.2)$$

Two remarks are in order. First, note that when compared to (1.1), (1.2) does not contain the convection term u_x . To find the solution of (1.1) from the solution of (1.2) amounts to shifting the solution to the right by an appropriate amount at each time. Thus it is not necessary to include the convection term in the numerical scheme. It should be mentioned however that in the context of nonhomogeneous boundary conditions on a finite domain, a simple translation would not be sufficient, because changing to a traveling frame of reference introduces moving boundary conditions. Since in the present case, homogeneous boundary conditions are given, and the solution u is supported in a set which stays well away from the boundary, it is appropriate to omit the convection term in the numerical integration.

Secondly, existence and uniqueness of solution of problems like (1.2) have been studied in [2,9,19]. It should also be mentioned that a different set of boundary conditions has been used by Colin and Ghidaglia [8]. They have placed a Dirichlet condition on the left, and a Neumann condition on the right. In addition, they pose a condition for the second derivative also on the right. Apparently, this combination of boundary conditions also leads to a well posed problem.

Our work is organized in the following way. In the next two sections, the finite-difference and the Chebyshev are explained, including the respective time discretizations. In the final section, the two methods are compared with respect to computational efficiency, a process we refer to as benchmarking. As a final application, the Chebyshev method is put to work in the numerical approximation of a two-soliton interaction. The result is shown in Fig. 6.

2. The finite-difference method

Finite-difference methods have been used for dispersive equations, for example in [12,30]. However, they appear to have been out of favor in recent years. Since the finite-difference discretization of (1.2) is straightforward, it is only briefly explained.

The spatial domain, in our case $[-l, l]$, is discretized by using function values on a finite set of points $\{x_j\}_{j=0}^N \subset [-l, l]$, where $x_0 = -l$, $x_N = l$ and $\Delta x = 2l/N$ is the uniform distance between two points. We discretize the time domain uniformly by $t_n = n\Delta t$. The derivatives at the point x_j are then approximated by using function values from a finite number of neighboring points. We define $v_j^n = u_N(x_j, t_n)$ for convenience. The first derivative is approximated using a second-order accurate central difference formula. The third derivative is also standard, but since it is less commonly used, we

provide the formula here. To approximate the third derivative at a point x_j , we use the formula

$$\frac{v_{j+2} - 2v_{j+1} + 2v_{j-1} - v_{j-2}}{2\Delta x^3}. \quad (2.1)$$

It can be seen from a simple analysis using Taylor series, that this approximation is second-order accurate in Δx if the function to be approximated is five times continuously differentiable. The Dirichlet conditions are $v_0^n = 0$ and $v_N^n = 0$. Hence, we need only solve the equation for the grid points $\{x_j\}_{j=1}^{N-1}$, which gives two points on each side for which the third derivative scheme is not valid. The Neumann condition on the right is implemented by requiring the central difference approximation to $u_x(l)$ to be zero. That is

$$\frac{v_{N+1}'' - v_{N-1}''}{2\Delta x} = 0. \quad (2.2)$$

This implies $v_{N-1}'' = v_{N+1}''$. Here, we have added a point outside the domain. This makes it possible to use the third derivative scheme (2.1) at the grid point x_{N-1} . As there is no Neumann condition at the left boundary, a similar treatment is not possible for the evaluation of the third derivative at x_1 . Instead, a different differentiation scheme is employed which uses only one point to the left of x_1 , and three points to the right:

$$\frac{-v_4 + 6v_3 - 12v_2 + 10v_1 - 3v_0}{2\Delta x^3}. \quad (2.3)$$

Applying these schemes and boundary conditions yields the following differentiation matrices for the first and third derivative operator, respectively:

$$D_{N,1} = \frac{1}{2\Delta x} \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ -1 & 0 & 1 & & \\ 0 & -1 & \ddots & & \vdots \\ \vdots & & & \ddots & 1 \\ 0 & & \cdots & -1 & 0 \end{pmatrix}, \quad (2.4)$$

and

$$D_{N,3} = \frac{1}{2\Delta x^3} \begin{pmatrix} 10 & -12 & 6 & -1 & 0 & \cdots & 0 \\ 2 & 0 & -2 & 1 & 0 & \cdots & 0 \\ -1 & 2 & 0 & -2 & 1 & & \vdots \\ 0 & -1 & 2 & & \ddots & & \\ \vdots & & & \ddots & & \ddots & 1 \\ & & & & \ddots & 0 & -2 \\ 0 & & \cdots & -1 & 2 & 1 & \end{pmatrix}. \quad (2.5)$$

For the linear term, a trapezoidal time integration scheme is used. This method has the advantage of being implicit and A-stable, even though it can be evaluated explicitly. To gain efficiency, the nonlinear term is treated explicitly, using a second-order Adams–Bashforth scheme. Combining these two methods yields the difference equation

$$\frac{v^{n+1} - v^n}{\Delta t} = -\frac{1}{2} \left(\frac{3}{2} D_{N,1}(v^n)^2 - \frac{1}{2} D_{N,1}(v^{n-1})^2 \right) - \frac{\Delta t}{2} (D_{N,3}v^{n+1} + D_{N,3}v^n) = 0, \quad (2.6)$$

where $v^n = (v_1^n, v_2^n, \dots, v_{N-1}^n)$. This can be solved with respect to v^{n+1} to advance the time step. Note that combining two second-order schemes does not automatically yield a second-order scheme for the full equation. However, numerical experiments suggest that the combined method is indeed second-order convergent in the size of the time step. A representative result is shown in Table 1, which shows that the temporal convergence rate in this case is nearly equal to 2.

Table 1

Convergence rates for the finite-difference method as $\Delta t \rightarrow 0$, using the single-soliton initial condition on $[-30, 30]$ with $c = 1$ and $N = 8192$. The codes were integrated to $t_1 = 1$. The errors and error rates are denoted by e and r , respectively, and the subscript denotes which error is used.

Δt	e_2	r_2	e_∞	r_∞
2^0	0.1485		0.1184	
2^{-1}	0.09225	1.61	0.06976	1.697
2^{-2}	0.02659	3.47	0.01874	3.722
2^{-3}	0.00612	4.344	0.004058	4.619
2^{-4}	0.001517	4.034	0.0009892	4.103
2^{-5}	0.0003799	3.993	0.0002463	4.017
2^{-6}	9.496e-05	4.001	6.151e-05	4.003
2^{-7}	2.351e-05	4.039	1.534e-05	4.01
2^{-8}	5.671e-06	4.146	3.829e-06	4.006
2^{-9}	1.481e-06	3.829	1.06e-06	3.613
2^{-10}	9.895e-07	1.497	6.274e-07	1.689

Since the difference Eq. (2.6) involves function values at three time steps, the first time step has to be treated in a different way. We decided to use a simple forward Euler method because the Euler method has local truncation error of order two, supplying a perfect match for the second-order convergence of the schemes used to construct (2.6). Moreover, instability is not likely to cause any problems during one single time step.

The stability of the scheme (2.6) is analyzed separately for the linear and nonlinear parts. Assuming that there are no boundary effects that affect the stability of the solution, and using the ansatz

$$v_j^n = \xi^n e^{ikj\Delta x}, \quad (2.7)$$

the stability criterion is $|\xi| \leq 1$. As stated above, the trapezoidal scheme is unconditionally stable for Airy's equation. This can be seen by inserting (2.7) into the difference equation (2.6). Disregarding the nonlinear terms, there appears the formula

$$\xi = \frac{1 - i(\Delta t/\Delta x^3)(\sin(2\kappa\Delta x) - \sin(\kappa\Delta x))}{1 + i(\Delta t/\Delta x^3)(\sin(2\kappa\Delta x) - \sin(\kappa\Delta x))}. \quad (2.8)$$

Since the nominator and denominator on the right hand side constitute a complex conjugate pair, we have $|\xi| = 1$. Hence the scheme is stable regardless of the choice of Δt and Δx .

For the nonlinear part, it will be assumed that the solution is bounded by a constant A . Thus there appears the transport equation

$$\eta_t + A\eta_x = 0. \quad (2.9)$$

Using the difference operator $D_{N,1}$ and the Adams–Bashforth scheme, the corresponding difference equation becomes

$$v_j^{n+1} = v_j^n - \frac{A\Delta t}{4\Delta x}(3v_{j+1}^n - 3v_{j-1}^n - v_{j+1}^{n-1} - v_{j-1}^{n-1}). \quad (2.10)$$

On inserting (2.7) and manipulating terms, we get a quadratic equation in ξ ,

$$\xi^2 - \left(1 - i\frac{3}{2}\frac{A\Delta t}{\Delta x}\sin(\kappa\Delta x)\right)\xi - i\frac{1}{2}\frac{A\Delta t}{\Delta x}\sin(\kappa\Delta x) = 0, \quad (2.11)$$

which can be solved using the quadratic formula, giving

$$\xi(\sigma) = \frac{1}{2} - i\frac{3}{4}\sigma \pm \frac{1}{2}\sqrt{1 - i\sigma - \frac{9}{4}\sigma^2}, \quad (2.12)$$

where $\sigma = (A\Delta t/\Delta x) > 0$. As it does not contribute to the growth, the factor $\sin(\kappa\Delta x)$ has been omitted.

The criterion for stability is $\xi_{\max} = \max(|\xi_+|, |\xi_-|) \leq 1$, where the subscripts denote the choice of sign of the square root. It can be seen by substitution that $\xi_+(0) = 1$ and $\xi_-(0) = 0$, although this is outside the domain we are considering. Fig. 1 shows the absolute values of the two modes of the growth factor as a function of σ . We see that $\xi_{\max} = |\xi_+|$.

Fig. 1. The two modes of the growth factor for the nonlinear term as functions of σ .

We also see that $|\xi_+|$ increases monotonically away from $\sigma = 0$, but that it approaches 1 as $\sigma \rightarrow 0$. Hence, there is in principle never stability, but the growth factor can be made as small as we please by choosing small enough Δt .

Next we provide some tables confirming that the second-order convergence is achieved in practice. Here, the L^2 -error and the L^∞ -error are used. They are defined respectively by

$$e_2(N, \Delta t, n) = \left(\Delta x \sum_{j=0}^N (u(x_j, n \Delta t) - v_j^n)^2 \right)^{1/2},$$

$$e_\infty(N, \Delta t, n) = \sup_{0 \leq j \leq N} |u(x_j, n \Delta t) - v_j^n|.$$

We will use both of these to determine the convergence rates. The procedure we have used for calculating spatial convergence rates, is to fix Δt , and start with a rather low value of N , integrate up to some $t = t_1$, measure the error, and do it again with N doubled, and so forth. The errors are then compared, and the ratio $r^{(i)} = e^{(i)}/e^{(i-1)}$ are calculated for each run. It is apparent from Table 2 that second-order convergence is achieved for the spatial discretization. The procedure is the same for the time convergence rates, only that the roles of N and Δt are reversed and Δt is halved, not doubled, in each run. To compute the error, we use a solitary wave solution [13,29] of (1.1), given by

$$u(x, t) = 12c^2 \operatorname{sech}(c(x - 4c^2t)). \quad (2.13)$$

Table 2

Convergence rates for the finite-difference method as $N \rightarrow \infty$, using the single-soliton initial condition on $[-30, 30]$ with $c = 1$ and $\Delta t = 10^{-5}$. The codes were integrated to $t_1 = 1$. The errors and error rates are denoted by e and r , respectively, and the subscript denotes which error is used.

N	e_2	r_2	e_∞	r_∞
32	0.1232		0.07264	
64	0.01963	6.275	0.01049	6.927
128	0.004439	4.421	0.00293	3.579
256	0.001095	4.055	0.000728	4.024
512	0.0002728	4.013	0.0001816	4.009
1024	6.815e-05	4.003	4.537e-05	4.003
2048	1.703e-05	4.001	1.134e-05	4.001
4096	4.258e-06	4	2.835e-06	4
8192	1.065e-06	4	7.086e-07	4

Since this function decays exponentially as $x \rightarrow \pm\infty$ it is an exact solution up to machine precision if the numerical domains $[-l, l]$ is large enough.

3. The Chebyshev-collocation method

A Chebyshev-collocation method for the approximation of solutions of the initial-boundary value problem (1.2) can be defined as follows. Recall that the Chebyshev polynomials are defined by $T_k(x) = \cos(k \arccos(x))$, for $k = 0, 1, 2, \dots$. Denoting the unknown by w_N , we use the representation

$$w_N(x, t) = \sum_{k=0}^N a_k(t) T_k(x). \quad (3.1)$$

The equation is enforced at the collocation points defined by

$$x_j = \cos\left(\frac{j\pi}{N}\right), \quad j = 0, 1, \dots, N. \quad (3.2)$$

The grid points being used, are the so-called Chebyshev Gauss–Lobatto points, which are the extremal points of the Chebyshev polynomial $T_N(x)$. The discrete collocation derivative is denoted by D_N . Let us briefly recall how to obtain the matrix D_N , as described in [5,28]. Given a set of function values $\{f_j\}_{j=0}^N$, there exists a unique polynomial p of degree $\leq N$, such that $f_j = p(x_j)$ for $j = 0, 1, \dots, N$. The linear map from $\{f_j\}_{j=0}^N$ to $\{p'(x_j)\}_{j=0}^N$ is represented by the matrix $D_N = (d_{ij})$ which is given explicitly by

$$\begin{aligned} d_{00} &= \frac{2N^2 + 1}{6}, & d_{NN} &= -\frac{2N^2 + 1}{6}, \\ d_{jj} &= \frac{-x_j}{2(1 - x_j^2)}, & j &= 1, \dots, N-1, \\ d_{ij} &= \frac{c_i(-1)^{i+j}}{c_j(x_i - x_j)}, & i \neq j, \quad i, j &= 1, \dots, N-1, \\ c_0 &= c_N = 2, & c_i &= 1 \quad \text{otherwise.} \end{aligned} \quad (3.3)$$

The differentiation matrix D_N is designed for the interval $[-1, 1]$ on the x -axis. In order to be able to use it on the interval $[-l, l]$, it needs to be scaled by $1/l$. For the third derivative operator, we simply use the cube D_N^3 . While this is not the most efficient way of computing the discrete third derivative, it has to be done only once at the beginning, so that it plays a minor role.

The boundary conditions require special care in the spectral method. This is especially true for the Neumann condition. Dirichlet conditions could be implemented by considering the trial functions $\phi(x) = T_k(x) - (-1)^n T_0(x)$ instead of $T_k(x)$ [17]. However, it appears more convenient to use the approach advocated by Trefethen [28] to simply truncate the Chebyshev differentiation matrices D_N and D_N^3 , by stripping the first and last columns and the uppermost and lowest rows. In this way, the boundary conditions $w(\pm l) = 0$ are enforced, and the remaining collocation points are in the interior of the interval.

To enforce the homogeneous Neumann condition, we use the following procedure. Instead of using an arbitrary polynomial satisfying $f_j = p(x_j)$, we require that $p(x) = (l - x)q(x)$ for some polynomial q of degree $\leq N$. Differentiating once gives $p_x = (l - x)q_x - q$, and the Neumann condition at $x = l$ is clearly satisfied. The third derivative is $p_{xxx} = (l - x)q_{xxx} - 3q_{xx}$. The procedure for the spectral differentiation is as follows. For a given set of values $\{f_j\}_{j=0}^N$, let q be the unique polynomial of degree $\leq N$ with $q(\pm 1) = 0$ and $q(x_j) = f_j/(l - x_j)$, for $1 \leq j \leq N-1$. Define the collocation derivatives by $(L_1 f)_j = (l - x_j)q_x(x_j) - q(x_j)$ and $(L_3 f)_j = (l - x_j)q_{xxx}(x_j) - 3q_{xx}(x_j)$. Thus the first derivative matrix becomes

$$L_1 = -\text{diag}\left(\frac{1}{l - x_j}\right) + \frac{1}{l} \text{diag}(l - x_j) \tilde{D}_N \text{diag}\left(\frac{1}{l - x_j}\right), \quad (3.4)$$

Table 3

Convergence rates for the spectral method as $\Delta t \rightarrow 0$, using the single-soliton initial condition on $[-30, 30]$ with $c=1$ and $N=1024$. The codes were integrated to $t_1=1$. The errors and error rates are denoted by e and r , respectively, and the subscript denotes which error is used.

Δt	e_2	r_2	e_∞	r_∞
2^{-5}	13.69		5.352	
2^{-6}	3.198	4.279	1.177	4.545
2^{-7}	0.6984	4.58	0.2217	5.312
2^{-8}	0.1286	5.431	0.05651	3.923
2^{-9}	0.03458	3.719	0.01874	3.015
2^{-10}	0.009485	3.646	0.006445	2.908
2^{-11}	0.00121	7.836	0.001354	4.761
2^{-12}	0.0002577	4.696	7.79e-05	17.37
2^{-13}	6.433e-05	4.006	2.665e-05	2.924
2^{-14}	1.605e-05	4.009	1.334e-05	1.998
2^{-15}	3.975e-06	4.036	6.652e-06	2.005

and the third derivative matrix becomes

$$L_3 = \frac{1}{l^3} \text{diag}(l - x_j) \tilde{D}_N^3 \text{diag} \left(\frac{1}{l - x_j} \right) - \frac{3}{l^2} \tilde{D}_N^2 \text{diag} \left(\frac{1}{l - x_j} \right). \quad (3.5)$$

Here, $\tilde{D}_N$, $\tilde{D}_N^2$, and $\tilde{D}_N^3$ are the first, second, and third powers of the Chebyshev differentiation matrix, with the first and last rows and columns removed.

For the Chebyshev method, the same method for time integration is used as for the finite-difference method. The resulting difference equation is

$$v^{n+1} = \left(I_N + \frac{\Delta t}{2} L_3 \right)^{-1} \left\{ \left(I_N - \frac{\Delta t}{2} L_3 \right) v^n - \frac{\Delta t}{4} (3L_1^2(v^n)^2 - L_1^2(v^{n-1})^2) \right\}. \quad (3.6)$$

Just like for the finite-difference method, the first time step needs special treatment. We still use the forward Euler scheme, because of its second-order local truncation error.

The scheme is validated by doing some numerical experiments with solitary waves. The procedure we have used for calculating the convergence rates is the same as in the previous section. However, in order to compute the L^2 -error, we first interpolate $\{w_j^n\}_{n=0}^N$ to a nonuniform grid using a cubic spline interpolation. This might not be optimal, and an interpolation using (3.1) might yield a somewhat better approximation. The results of the convergence study are shown in Tables 3 and 4.

As can be seen in Table 4, spectral convergence is achieved before the error starts to grow again. This divergence is most likely due to instability in the numerical scheme. Indeed, some of the eigenvalues of D_N , and therefore of L_1 and L_3 have positive real part. The linear part is treated with a trapezoidal method, and is thus susceptible for instability. However, the nonlinear term is computed using a second-order Adams–Bashforth method, which has a smaller stability domain than the trapezoidal method. Moreover, the stability of the nonlinear term is likely to suffer from the imposition of three boundary conditions. We therefore expect the nonlinear term to dominate any possible instability. Fig. 2 shows the eigenvalues λ of L_1 in the case that $N=256$. The eigenvalues are multiplied by the time step Δt , for four different

Table 4

Convergence rates for the spectral method as $N \rightarrow \infty$, using the single-soliton initial condition on $[-30, 30]$ with $c=1$ and $\Delta t=10^{-5}$. The codes were integrated to $t_1=1$. The errors and error rates are denoted by e and r , respectively, and the subscript denotes which error is used.

N	e_2	r_2	e_∞	r_∞
32	0.3982		0.1916	
64	0.0396	10.06	0.02101	9.121
128	1.503e-05	2635	7.871e-06	2669
256	4.994e-11	3.009e+05	3.488e-11	2.256e+05
512	4.208e-11	1.187	2.622e-09	0.01331
1024	3.875e-07	0.0001086	2.137e-06	0.001227
2048	0.003516	0.0001102	0.001066	0.002006

Fig. 2. An illustration of the behavior of $\lambda\Delta t$, where λ are the eigenvalues of L_1 , as $\Delta t \rightarrow 0$. The shaded area is the stability region of the Adams–Bashforth method. $N=256$ in all plots. Note that it is the scale of the axis that changes, not the size of the stability region.

values of Δt . As can be gleaned from Fig. 2, there will always be some eigenvalues with positive real part, but they approach zero as $\Delta t \rightarrow 0$. Thus, the eigenvalues come arbitrarily close to the boundary of the stability region as $\Delta t \rightarrow 0$ for some fixed N . It also turns out that the magnitudes of the eigenvalues increase as $N \rightarrow \infty$, reflecting the severity of the condition on the size of the time step which was already expected because for large N , the spatial grid size near the boundaries is extremely small.

For optimal results one might try to use a third-order Adams–Bashforth method. Implicit methods are probably not viable for the nonlinear term since our treatment of the boundary conditions prohibit the use of the fast Fourier transform.

4. Benchmarking

One of the main goals of this study is to determine whether the finite-difference or the Chebyshev method is more effective. In this section we will present a comparison of the numerical methods developed in the previous two sections. In particular, we will present results from a benchmarking test, contrasting the computational efficiency of the two methods. While there are some benchmarking results for numerical schemes for the KdV equation [4,27], there does not seem to exist a comparison of numerical projections of the initial-boundary value problem (1.2). In particular, we do not have any knowledge of previous studies that include benchmarking tests of Chebyshev methods versus finite-difference methods.

The benchmarking was carried out for a single solitary wave (2.13). Given a few different tolerance levels ε , we found combinations of N and Δt that gave results with error in the ℓ^∞ -norm below the tolerance level in the least amount of time for the two respective methods. As a general guide for finding the most advantageous combinations of N and Δt , we expect that they should be close to the values where errors due to the temporal discretization and errors due to the spatial discretization are approximately balanced. After obtaining these values for N and Δt , the run times for the two codes for each tolerance level were compared. Results from benchmarking are given in Tables 5–7. The spatial and temporal resolutions for the finite-difference and Chebyshev codes are given by $(N, \Delta t^{-1})_{FD}$ and $(N,$

Table 5

Benchmarking results, $t_1 = 1$. The tolerance level is ε . $(N, \Delta t^{-1})_{FD}$ and $(N, \Delta t^{-1})_C$ give the spatial and temporal resolutions for the finite-difference and Chebyshev codes, respectively. T_{FD} and T_C are the respective median run times in seconds.

ε	$(N, \Delta t^{-1})_{FD}$	$(N, \Delta t^{-1})_C$	T_{FD}	T_C
10^{-2}	(62, 5)	(73, 13)	0.001306	0.007201
10^{-3}	(200, 20)	(93, 30)	0.005312	0.013357
10^{-4}	(633, 64)	(111, 61)	0.038326	0.022427
10^{-5}	(1996, 201)	(129, 213)	0.35963	0.045229
10^{-6}	(6338, 629)	(157, 489)	4.2295	0.096428
10^{-7}		(173, 1669)		0.28512

Table 6

Benchmarking results, $t_1 = 2$. The tolerance level is ε . $(N, \Delta t^{-1})_{FD}$ and $(N, \Delta t^{-1})_C$ give the spatial and temporal resolutions for the finite-difference and Chebyshev codes, respectively. T_{FD} and T_C are the respective median run times in seconds.

ε	$(N, \Delta t^{-1})_{FD}$	$(N, \Delta t^{-1})_C$	T_{FD}	T_C
10^{-2}	(73, 5)	(75, 9)	0.001827	0.007855
10^{-3}	(226, 18)	(95, 29)	0.008604	0.015389
10^{-4}	(704, 57)	(109, 65)	0.063784	0.025376
10^{-5}	(2220, 180)	(129, 261)	0.63594	0.065847
10^{-6}	(7024, 565)	(151, 692)	7.1751	0.16266
10^{-7}		(173, 1971)		0.57488

$\Delta t^{-1})_C$, respectively. The respective median run times in seconds are T_{FD} and T_C . Figs. 3–5 plot the median run times against the negative logarithm of ε .

The benchmarking tests show that the spectral method performs much better than the finite-difference method, at least for ε less than 10^{-2} , which are the most relevant cases. We also see from Figs. 3–5 that the difference between the performances of the two methods increases as $\varepsilon \rightarrow 0$. For $\varepsilon = 10^{-4}$ the relation T_{FD}/T_C takes values between about 1.7 and 2.7. For $\varepsilon = 10^{-5}$, this has increased to between 7.8 and 10.9, and for $\varepsilon = 10^{-6}$ it is in the interval from 40 to 44. We would expect this trend to continue in a similar manner, at least for a while. However, in Figs. 3–5 we see that the slopes of the curves corresponding to the spectral method increase slightly as ε decreases. This could mean that for very low tolerances, the finite-difference method might pass the spectral method with respect to computational efficiency. If this scenario were to play out, it would likely happen only for tolerances below machine precision. On the other hand, for integration over long time scales, the finite-difference method may gain on the Chebyshev method because of possible instability due to the time integration scheme that was used. Note that the graph corresponding to the finite-difference method in Figs. 3–5 is approximately linear. This means that if the tolerance is lowered by a ratio r , then the computation time increased roughly by the same ratio. Consequently, the implementation of the finite-difference method was nearly optimal.

In conclusion, it can be said that the Chebyshev method is more efficient than the finite-difference method, at least for relatively short time integrations. The spectral properties of the Chebyshev projections probably make it not the best candidate for a stable discretization. A method based on Legendre polynomials is expected to perform at least

Table 7

Benchmarking results, $t_1 = 4$. The tolerance level is ε . $(N, \Delta t^{-1})_{FD}$ and $(N, \Delta t^{-1})_C$ give the spatial and temporal resolutions for the finite-difference and Chebyshev codes, respectively. T_{FD} and T_C are the respective median run times in seconds.

ε	$(N, \Delta t^{-1})_{FD}$	$(N, \Delta t^{-1})_C$	T_{FD}	T_C
10^{-2}	(76, 5)	(77, 9)	0.00268	0.008815
10^{-3}	(234, 17)	(97, 28)	0.014073	0.019224
10^{-4}	(741, 52)	(111, 89)	0.10772	0.039817
10^{-5}	(2312, 164)	(131, 228)	1.0909	0.1002
10^{-6}	(7244, 521)	(151, 755)	12.55	0.30579
10^{-7}		(173, 2607)		1.4442

Fig. 3. Comparison of median run times for the two models on different tolerance levels, $t_1 = 1$. The values on the x -axis are the negative logarithms of ϵ . The values on the y -axis are the run times for the codes in seconds. The error bars correspond to the 0.01 and 0.99 quantiles.

Fig. 4. Comparison of median run times for the two models on different tolerance levels, $t_1 = 2$. The values on the x -axis are the negative logarithms of ϵ . The values on the y -axis are the run times for the codes in seconds. The error bars correspond to the 0.01 and 0.99 quantiles.

Fig. 5. Comparison of median run times for the two models on different tolerance levels, $t_1 = 4$. The values on the x-axis are the negative logarithms of ϵ . The values on the y-axis are the run times for the codes in seconds. The error bars correspond to the 0.01 and 0.99 quantiles.

as well as the Chebyshev method. An interesting hybrid method was proposed by Ma and Sun [23], where the linear term was treated by a Legendre–Petrov method, and the nonlinear term was treated using a Chebyshev method. Ma and Sun were able to evaluate the nonlinear term pseudospectrally. It would be interesting to compare the efficiency of their scheme with the Chebyshev-collocation method used in the present article. We should also note the work of Shen [26] who has proposed a dual Petrov–Galerkin method with nearly optimal computational complexity.

Fig. 6. Two-soliton interaction: we see that the taller wave overtakes the shorter wave, and that both waves experience a phase shift in the interaction. No dispersive tail is observed. The initial profile was given by $u(x, 0) = 2((3 + 4 \cosh(2x - 8t_0)) + \cosh(4x - 64t_0)) / (3 \cosh(x - 28t_0) + \cosh(3x - 36t_0))^2$ for an appropriate value of t_0 .

Finally, we would like to mention that there are more advanced time integration methods available for the KdV and similar equations. We chose to use the same time discretization for both the finite-difference and the Chebyshev method in order to have a fair comparison between the two spatial discretizations. However, our choice of time integration may not be the most advantageous match for either the finite-difference or the Chebyshev discretization. The methods presented by Fornberg and Driscoll [15] and by de Frutos and Sanz-Serna [11] might be a better match for the Chebyshev method than the mixed Adams–Bashforth and Crank–Nicolson method we have used. However, better as they may be, we do not expect these methods to change our benchmarking results in a qualitative way.

Acknowledgments

This work was supported in part by the Research Council of Norway. The authors wish to thank Einar Rønquist for helpful comments.

References

- [1] J.L. Bona, V.A. Dougalis, O.A. Karakashian, W.R. McKinney, Conservative, high-order numerical schemes for the generalized Korteweg–de Vries equation, *Philos. Trans. R. Soc. Lond. Ser. A* 351 (1995) 107–164.
- [2] J.L. Bona, S.M. Sun, B.-Y. Zhang, A nonhomogeneous boundary-value problem for the Korteweg–de Vries equation posed on a finite domain, *Comm. Partial Diff. Eq.* 28 (2003) 1391–1436.
- [3] J. Boussinesq, Théorie des ondes et des remous qui se propagent le long d'un canal rectangulaire horizontal, en communiquant au liquide contenu dans ce canal des vitesses sensiblement pareilles de la surface au fond, *J. Math. Pures Appl.* 17 (1872) 55–108.
- [4] D.G. Cacuci, O.A. Karakashian, Benchmarking the propagator method for nonlinear systems: a Burgers–Korteweg–de Vries equation, *J. Comput. Phys.* 89 (1990) 63–79.
- [5] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, *Spectral Methods in Fluid Dynamics*, Springer, 1988.
- [6] G.F. Carey, Y. Shen, Approximations of the KdV equation by least squares finite elements, *Comput. Methods Appl. Mech. Eng.* 93 (1991) 1–11.
- [7] R.C. Cascaval, Variable coefficient KdV equations and waves in elastic tubes, *Evolution equations Lecture Notes in Pure and Appl. Math.*, vol. 234, Dekker, New York, 2003, pp. 57–69.
- [8] T. Colin, J.-M. Ghidaglia, An initial-boundary value problem for the Korteweg–de Vries equation posed on a finite interval, *Adv. Diff. Eq.* 6 (2001) 1463–1492.
- [9] J.E. Colliander, C.E. Kenig, The generalized Korteweg–de Vries equation on the half line, *Comm. Partial Diff. Eq.* 27 (2002) 2187–2266.
- [10] D.G. Crighton, Applications of KdV, *Acta Appl. Math.* 39 (1995) 39–67.
- [11] J. de Frutos, J.M. Sanz-Serna, An easily implementable fourth-order method for the time integration of wave problems, *J. Comput. Phys.* 103 (1992) 160–168.
- [12] K. Djidjeli, W.G. Price, E.H. Twizell, Y. Wang, Numerical methods for the solution of the third- and fifth-order dispersive Korteweg–de Vries equations, *J. Comput. Appl. Math.* 58 (1995) 307–336.
- [13] P.G. Drazin, R.S. Johnson, *Solitons: an introduction*, in: *Cambridge Texts in Applied Mathematics*, Cambridge University Press, Cambridge, 1989.
- [14] B. Fornberg, T. Driscoll, A fast spectral algorithm for nonlinear wave equations with linear dispersion, *J. Comput. Phys.* 155 (1999) 456–467.
- [15] B. Fornberg, G.B. Whitham, A numerical and theoretical study of certain nonlinear phenomena, *Philos. Trans. R. Soc. Lond. Ser. A* 289 (1978) 373–404.
- [16] D. Gottlieb, S. Orszag, *Numerical Analysis of Spectral Methods: Theory and Applications*, SIAM, Philadelphia, PA, 1977.
- [17] W. Heinrichs, Spectral approximation of third-order problems, *J. Sci. Comput.* 14 (1999) 275–289.
- [18] J. Holmer, The initial-boundary value problem for the Korteweg–de Vries equation, *Comm. Partial Diff. Eq.* 31 (2006) 1151–1190.
- [19] A. Jeffrey, T. Kakutani, Weak nonlinear dispersive waves: a discussion centered around the Korteweg–de Vries equation, *SIAM Rev.* 14 (1972) 582–643.
- [20] H. Kalisch, X. Raynaud, On the rate of convergence of a collocation projection of the KdV equation, *Math. Model. Numer. Anal.* 41 (2007) 95–110.
- [21] D.J. Korteweg, G. de Vries, On the change of form of long waves advancing in a rectangular channel and on a new type of long stationary wave, *Philos. Mag.* 39 (1895) 422–443.
- [22] H. Ma, W. Sun, A Legendre–Petrov–Galerkin method and Chebyshev collocation method for the third-order differential equations, *SIAM J. Numer. Anal.* 38 (2000) 1425–1438.
- [23] Y. Maday, A. Quarteroni, Error analysis for spectral approximation of the Korteweg–de Vries equation, *RAIRO Modl. Math. Anal. Num.* 22 (1988) 499–529.
- [24] D. Pavoni, Single and multidomain Chebyshev collocation methods for the Korteweg–de Vries equation, *Calcolo* 25 (1988) 311–346.
- [25] J. Shen, A new dual-Petrov–Galerkin method for third and higher odd-order differential equations: application to the KDV equation, *SIAM J. Numer. Anal.* 41 (2003) 1595–1619.
- [26] T. Taha, M. Ablowitz, Analytical and numerical aspects of certain nonlinear evolution equations. III. Numerical, Korteweg–de Vries equation, *J. Comput. Phys.* 55 (1984) 231–253.

- [28] L.N. Trefethen, *Spectral Methods in Matlab*, SIAM, Philadelphia, 2000.
- [29] G.B. Whitham, *Linear and Nonlinear Waves*, Wiley, New York, 1974.
- [30] N.J. Zabusky, M.D. Kruskal, Interaction of solutions in a collisionless plasma and the recurrence of initial states, *Phys. Rev. Lett.* 15 (1965) 240–243.