ANZIAM J. 47(2005), 131–142

NUMERICAL INTEGRATION OF THE RLW EQUATION USING CUBIC SPLINES

DURSUN IRK[✉1], İDRIS DAĞ[2] and ABDÜLKADIR DOĞAN[3]

(Received 16 July, 2003)

Abstract

A numerical solution of the RLW equation is presented using a cubic spline collocation method. Basic cubic spline relations are outlined and incorporated into the numerical solution procedure. Two test problems are studied to show the robustness of the proposed procedure.

1. Introduction and governing equation

A numerical solution of the regularised long wave (RLW) equation in the form

$$U_t + U_x + \varepsilon U U_x - \mu U_{xxt} = 0, \tag{1.1}$$

where the subscripts t and x denote differentiation and ε and μ are positive parameters, is studied with the artificial boundary conditions

$$\begin{aligned} U(a,t) &= \alpha_1, \qquad U(b,t) = \alpha_2, \\ U_x(a,t) &= 0, \qquad U_x(b,t) = 0, \end{aligned} \qquad a \le x \le b, \quad t \ge 0 \tag{1.2}$$

and initial conditions

$$U(x,0) = f(x), \tag{1.3}$$

where $f(x)$ will be chosen later. Peregrine [13] proposed the RLW equation to describe the development of the undular bore and wave motion. Mathematical theory

[1]Department of Mathematics, Osmangazi University, Eskişehir, Türkiye; email: ogumath@yahoo.com.
[2]Department of Computer Engineering, Osmangazi University, Eskişehir, Türkiye; e-mail: idag@ogu.edu.tr.
[3]Department of Mathematics, Niğde University, Niğde, Türkiye.
© Australian Mathematical Society 2005, Serial-fee code 1446-1811/05

Downloaded from https://www.cambridge.org/core. Abdullah Gul Universitesi, on 02 Apr 2020 at 12:10:21, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1446181100009822

of the equation was developed in the papers [2] and [4]. Exact solutions exist for the limited boundary and initial conditions. Therefore numerical studies of the equation are of interest in explaining physical phenomena such as shallow water waves and ion acoustic plasma waves.

The cubic spline collocation procedure and its application have been widely used in the numerical solution of partial differential equations (PDEs) [5, 14–16] because they possess some advantages over other numerical methods without the disadvantages of being computationally intensive and having a complex problem formulation. Various numerical schemes for obtaining the numerical solution of the RLW equation incorporating spline functions have been set up. Finite difference methods were proposed based on both cubic and quintic splines in the papers [3, 11]. Some variants of finite element and collocation methods were constructed for the RLW equation by using B-splines as weight and trial functions [6–9]. The use of cubic splines in numerical methods for finding solutions of PDEs leads to a matrix system which is tridiagonal, thus permitting the use of the Thomas algorithm. Evaluation of the first and second derivatives is obtained directly from the spline relations. Boundary conditions involving derivatives may be directly incorporated into the solution procedure.

In Section 2 of this paper, necessary basic spline formulations are given and then an application of the proposed scheme on the RLW equation is presented. So the RLW equation is turned into three separate scalar matrix systems of tridiagonal form consisting of unknown parameters U, first derivatives U' and second derivatives U'' respectively at the knots. In Section 3, the proposed method is tested for the model problems of solitary wave solution and undular bore development. The computed results are compared with the analytical solution for the solitary wave.

2. Numerical method

Consider a mesh with nodal points such that $a = x_0 < x_1 < \cdots < x_N = b$ and $h = x_i - x_{i-1} > 0$, $i = 1, 2, \ldots, N$.

We outline the basic cubic spline relations and use them in the solution of the RLW equation. The function $U(x, t)$, its first derivative U' and its second derivative U'' are defined on $[a, b]$ respectively, at the node point x_i by:

$$U(x_i, t) = U_i, \quad U'(x_i, t) = U_i' \quad \text{and} \quad U''(x_i, t) = U_i'',$$

where $'$ and $''$ denote differentiation with respect to x. Let $S_p(U, x) = S_p(x)$ denote the cubic spline function.

The cubic spline function and its first and second derivatives are continuous on the solution domain $[a, b]$ and satisfy

$$S_p(x_i) = U_i, \quad S_p'(x_i) = m_i \quad \text{and} \quad S_p''(x_i) = M_i.$$

Downloaded from https://www.cambridge.org/core. Abdullah Gul Universitesi, on 02 Apr 2020 at 12:10:21, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1446181100009822

In general, since $S_p(x)$ is a cubic spline function on each interval $[x_{i-1}, x_i]$, the function $S_p''(x)$ is linear on the interval and can be determined by two points $S_p''(x_{i-1}) = M_{i-1}$, $S_p''(x_i) = M_i$ from

$$S_p''(x) = \frac{(x_i - x)M_{i-1} + (x - x_{i-1})M_i}{h}.$$

We will form the second antiderivative of $S_p''(x)$ on $[x_{i-1}, x_i]$ and apply the interpolating conditions, which consist of continuity of the function and its first derivative values at the node points, to evaluate the integration constants. This results in the cubic spline interpolation polynomial:

$$S_p(x) = \frac{(x_i - x)^3 M_{i-1} + (x - x_{i-1})^3 M_i}{6h} + \frac{(x_i - x)U_{i-1} + (x - x_{i-1})U_i}{h} - \frac{h}{6}((x_i - x)M_{i-1} + (x - x_{i-1})M_i)$$

in the interval $[x_{i-1}, x_i]$. The main cubic spline relationships [1] are

$$M_{i-1} + 4M_i + M_{i+1} = 6(U_{i+1} - 2U_i + U_{i-1})/h^2, \tag{2.1}$$

$$m_{i-1} + 4m_i + m_{i+1} = 3(U_{i+1} - U_{i-1})/h, \tag{2.2}$$

$$m_{i+1} - m_i = h(M_i + M_{i+1})/2, \tag{2.3}$$

$$m_i = \begin{cases} M_i h/3 + M_{i-1}h/6 + (U_i - U_{i-1})/h & \text{or} \\ -M_i h/3 - M_{i+1}h/6 + (U_{i+1} - U_i)/h, \end{cases} \tag{2.4}$$

$$M_i = \begin{cases} 2m_{i-1}/h + 4m_i/h - 6(U_i - U_{i-1})/h^2 & \text{or} \\ -4m_i/h - 2m_{i+1}/h + 6(U_{i+1} - U_i)/h^2. \end{cases} \tag{2.5}$$

Details about these relationships can be found in Ahlberg *et al.* [1]. The RLW equation can be written in the form

$$U_t - \mu U_{xxt} = -U_x - \varepsilon U U_x = f(U). \tag{2.6}$$

An approximate solution for U_i can be obtained by considering the solution of

$$(U_t)_i - \mu(U_{xxt})_i = (1 - \theta)(f^n)_i + \theta(f^{n+1})_i, \tag{2.7}$$

where the time derivative is discretised in the usual finite-difference way:

$$(U_i^{n+1} - U_i^n) - \mu(M_i^{n+1} - M_i^n) = (1 - \theta)\Delta t(f^n)_i + \theta\Delta t(f^{n+1})_i, \tag{2.8}$$

where $(f^n)_i = -(m_i^n + \varepsilon U_i^n m_i^n)$. The nonlinear term in (2.6) may be linearised by using the following term [14, 15]:

$$(UU_x)_i^{n+1} = U_i^{n+1} m_i^n + U_i^n m_i^{n+1} - U_i^n m_i^n. \tag{2.9}$$

Downloaded from https://www.cambridge.org/core. Abdullah Gul Universitesi, on 02 Apr 2020 at 12:10:21, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1446181100009822

So, (2.8) can be rewritten as

$$\begin{aligned} U_i^{n+1}(1+\theta\Delta t\varepsilon m_i^n) = {} & U_i^n + \mu M_i^{n+1} - \mu M_i^n \\ & - (1-\theta)\Delta t\{m_i^n + \varepsilon(U_i^n m_i^{n-1} + U_i^{n-1} m_i^n - U_i^{n-1} m_i^{n-1})\} \\ & - \theta\Delta t\{m_i^{n+1} + \varepsilon(U_i^n m_i^{n+1} - U_i^n m_i^n)\}. \end{aligned} \tag{2.10}$$

Rearranging (2.10), we get

$$U_i^{n+1} = F_i + G_i m_i^{n+1} + S_i M_i^{n+1}, \tag{2.11}$$

where

$$\begin{aligned} F_i = {} & \frac{U_i^n - \mu M_i^n - (1-\theta)\,\Delta t(m_i^n + \varepsilon(U_i^n m_i^{n-1} + U_i^{n-1} m_i^n - U_i^{n-1} m_i^{n-1}))}{1+\theta\Delta t\varepsilon m_i^n} \\ & + \frac{\varepsilon\theta\Delta t\, U_i^n m_i^n}{1+\theta\Delta t\varepsilon m_i^n}, \\ G_i = {} & \frac{\theta\Delta t(-1-\varepsilon U_i^n)}{1+\theta\Delta t\varepsilon m_i^n} \quad \text{and} \quad S_i = \frac{\mu}{1+\theta\Delta t\varepsilon m_i^n}. \end{aligned}$$

It can be noted that $\theta = 0$ gives an explicit scheme, $\theta = 1$ gives a fully implicit scheme and $\theta = 1/2$ gives the Crank-Nicholson scheme. Equation (2.11) can be reduced to three separate tridiagonal equations for U_i, m_i and M_i using (2.1)–(2.5). Using (2.4), (2.11) can be transformed into a scalar set of equations involving U_{i-1}, U_i and U_{i+1}:

$$A_i U_{i-1}^{n+1} + B_i U_i^{n+1} + C_i U_{i+1}^{n+1} = D_i, \tag{2.12}$$

where

$$\begin{aligned} A_i = {} & 72S_{i-1}S_iS_{i+1} + 12h^2G_{i-1}G_iS_{i+1} - 36hG_{i-1}S_iS_{i+1} + 3h^3G_{i-1}G_iG_{i+1} \\ & - 12h^2G_{i-1}S_iG_{i+1} + h^4G_iG_{i+1} - 24hS_{i-1}G_iS_{i+1} - 6h^2S_{i-1}G_iG_{i+1} \\ & + 24hS_{i-1}S_iG_{i+1} + 4h^3G_iS_{i+1} - 12h^2S_iS_{i+1} - 4h^3S_iG_{i+1}, \\ B_i = {} & 60hG_{i-1}S_iS_{i+1} + 4h^4G_{i-1}G_{i+1} + 24h^2G_{i-1}S_iG_{i+1} - 6h^2G_{i-1}G_iS_{i+1} \\ & + 14h^3G_{i-1}S_{i+1} - 60hS_{i-1}S_iG_{i+1} - 14h^3S_{i-1}G_{i+1} - 144S_{i-1}S_iS_{i+1} \\ & - 48h^2S_{i-1}S_{i+1} - 6h^2S_{i-1}G_iG_{i+1}, \\ C_i = {} & -4h^3S_{i-1}G_i + 4h^3G_{i-1}S_i + h^4G_{i-1}G_i - 3h^3G_{i-1}G_iG_{i+1} \\ & - 12h^2G_{i-1}S_iG_{i+1} + 24hS_{i-1}G_iS_{i+1} + 12h^2S_{i-1}G_iG_{i+1} + 72S_{i-1}S_iS_{i+1} \\ & + 36hS_{i-1}S_iG_{i+1} - 12h^2S_{i-1}S_i - 6h^2G_{i-1}G_iS_{i+1} - 24hG_{i-1}S_iS_{i+1} \quad \text{and} \\ D_i = {} & -14h^3S_{i-1}F_iG_{i+1} + 4h^3F_{i-1}G_iS_{i+1} - 48h^2S_{i-1}F_iS_{i+1} + 4h^3G_{i-1}S_iF_{i+1} \\ & + 14h^3G_{i-1}F_iS_{i+1} + 4h^4G_{i-1}F_iG_{i+1} - 12h^2F_{i-1}S_iS_{i+1} + h^4F_{i-1}G_iG_{i+1} \\ & + h^4G_{i-1}G_iF_{i+1} - 12h^2S_{i-1}S_iF_{i+1} - 4h^3S_{i-1}G_iF_{i+1} - 4h^3F_{i-1}S_iG_{i+1}. \end{aligned}$$

Downloaded from https://www.cambridge.org/core. Abdullah Gul Universitesi, on 02 Apr 2020 at 12:10:21, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1446181100009822

Using basic cubic spline formulations (2.5) and eliminating U_i and M_i from (2.11), the following triangular form of matrix system is derived for m_i:

$$A_i m_{i-1}^{n+1} + B_i m_i^{n+1} + C_i m_{i+1}^{n+1} = D_i, \tag{2.13}$$

where

$$A_i = \frac{1}{3h} - \left(\frac{2S_i + 4S_{i-1} - hG_{i-1}}{h^3\Delta_i}\right),$$
$$B_i = \frac{4}{3h} - \left(\frac{2S_{i+1} + 4S_i - hG_i}{h^3\Delta_{i+1}} + \frac{2S_{i-1} + 4S_i - hG_i}{h^3\Delta_i}\right),$$
$$C_i = \frac{1}{3h} - \left(\frac{2S_i + 4S_{i+1} - hG_{i+1}}{h^3\Delta_{i+1}}\right),$$
$$D_i = \frac{F_{i+1} - F_i}{h^2\Delta_{i+1}} + \frac{F_i - F_{i-1}}{h^2\Delta_i} \quad \text{and} \quad \Delta_i = 1 + 6\left(\frac{S_i + S_{i-1}}{h^2}\right).$$

Using (2.2)–(2.4a) and solving (2.11) by eliminating m_i for M_i, the following single tridiagonal system may be obtained:

$$A_i M_{i-1}^{n+1} + B_i M_i^{n+1} + C_i M_{i+1}^{n+1} = D_i, \tag{2.14}$$

where

$$A_i = \frac{h}{6} + \frac{G_i + 2G_{i-1}}{6\Delta_i} - \frac{S_{i-1}}{h\Delta_i},$$
$$B_i = \frac{2h}{3} - \frac{G_{i+1} + 2G_i}{6\Delta_{i+1}} + \frac{2G_i + G_{i-1}}{6\Delta_i} + S_i\left(\frac{1}{h\Delta_{i+1}} + \frac{1}{h\Delta_i}\right),$$
$$C_i = \frac{h}{6} - \frac{2G_i + G_i}{6\Delta_{i+1}} - \frac{S_{i+1}}{h\Delta_{i+1}},$$
$$D_i = \frac{F_{i+1} - F_i}{h\Delta_{i+1}} - \frac{F_i - F_{i-1}}{h\Delta_i} \quad \text{and} \quad \Delta_i = 1 - \frac{G_i - G_{i-1}}{h}.$$

Approximate values U_i at the nodal points can be calculated using the resulting system (2.12). Direct calculation of the first derivatives m_i and second derivatives M_i can be obtained by working out the systems (2.13)–(2.14) respectively. Since those systems require the computation of U_i, m_i and M_i from the two previous time steps, the first time step is calculated by putting $\theta = 1$. This means that the formula only requires U_i, m_i and M_i from the previous time step. For the next time steps, any value of θ between 0 and 1 can be used. We used the fully implicit scheme in our computation.

Downloaded from https://www.cambridge.org/core. Abdullah Gul Universitesi, on 02 Apr 2020 at 12:10:21, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1446181100009822

TABLE 1. Single solitary wave, amplitude = 0.3, $h = 0.125$, $\Delta t = 0.1$, $-40 \leqslant x \leqslant 60$.

Time	$L_2 \times 10^3$	$L_\infty \times 10^3$	I_1	I_2	I_3
0	0.0	0.0	3.97992	0.810462	2.57900
4	0.191	0.083	3.97995	0.810276	2.57839
8	0.208	0.087	3.97997	0.810276	2.57839
12	0.234	0.095	3.97998	0.810276	2.57839
16	0.266	0.104	3.97998	0.810276	2.57839
20	0.301	0.114	3.97996	0.810276	2.57839

The local truncation error is computed as

$$\begin{aligned} & k\left[6\mu^2(U_{tt})_i^n + 12\mu^2\theta\varepsilon(U)_i^n(U_{xt})_i^n + 12\mu^2\theta(U_{xt})_i^n + 12\mu^2\theta\varepsilon(U_x)_i^n(U_t)_i^n\right] \\ & \quad + h^2[2\mu^2(U_{xxx})_i^n + 2\mu^2(U_{xxt})_i^n + 6\mu^2\varepsilon(U_x)_i^n(U_{xx})_i^n + 2\mu^2\varepsilon(U)_i^n(U_{xxx})_i^n] \\ & \quad + h^2k[2\mu^2\theta\varepsilon(U_{xxx})_i^n(U_t)_i^n + 6\mu^2\theta\varepsilon(U_{xx})_i^n(U_{xt})_i^n - 6\mu\theta\varepsilon^2(U)_i^n((U_x)^2)_i^n \\ & \quad + 12\mu^2\theta\varepsilon(U_x)_i^n(U_{xxt})_i^n - 6\mu\theta\varepsilon((U_x)^2)_i^n - 6\mu\theta\varepsilon(U_x)_i^n(U_t)_i^n] + \cdots \end{aligned} \tag{2.15}$$

by neglecting the terms of higher order. Therefore (2.15) gives a truncation error of $O(k + h^2 + kh^2)$ for $\theta \in [0, 1]$.

Applying the von Neuman stability analysis by taking U constant in the nonlinear term UU_x in (1.1), we find that the proposed technique is unconditionally stable for $\theta \in [1/2, 1]$.

3. Numerical computations

The numerical scheme has been assessed by using the maximum error L_∞,

$$L_\infty = \| U^{\text{exact}} - U_N \|_\infty = \max_j | U_j^{\text{exact}} - (U_N)_j |, \tag{3.1}$$

and the root mean square error L_2,

$$L_2 = \sqrt{h \sum_{j=1}^{N} | U_j^{\text{exact}} - (U_N)_j |^2}. \tag{3.2}$$

Olver [12] showed that the RLW equation has three conservation laws:

$$I_1 = \int_{-\infty}^{\infty} U\,dx, \quad I_2 = \int_{-\infty}^{\infty} (U^2 + \mu(U_x)^2)\,dx \quad \text{and} \quad I_3 = \int_{-\infty}^{\infty} (U^3 + 3U^2)\,dx.$$

Conservation quantities and error norms will be computed to show how well the numerical scheme behaviour models the test problems. Firstly we will study the

Downloaded from https://www.cambridge.org/core. Abdullah Gul Universitesi, on 02 Apr 2020 at 12:10:21, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1446181100009822

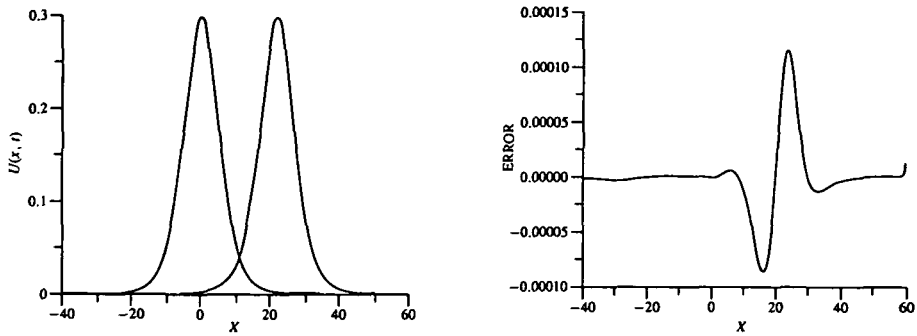


FIGURE 1. (a) Graphs of solutions at times $t = 0$ and $t = 20$. (b) Error = exact − numerical solution at time $t = 20$.

solitary wave solution of the RLW equation. The analytical solution of the solitary wave solution is computed from

$$U(x, t) = 3c \sec h^2(k[x - x_0 - (1 + \varepsilon c)t]), \tag{3.3}$$

which represents a single solitary wave of amplitude $3c$, velocity $v = 1 + \varepsilon c$ and $k = (\varepsilon c/\mu(1 + \varepsilon c))^{1/2}/2$ travelling across the interval $-40 \leq x \leq 60$ in the time period $0 \leq t \leq 20$ with parameters $\varepsilon = \mu = 1$ and $x_0 = 0$.

The initial condition is used from (3.3) by putting $x_0 = 0$ and $t = 0$. The parameters $c = 0.1$, $\Delta t = 0.1$ and $h = 0.125$ were chosen to coincide with those in papers [6, 7, 10]. The constants of motion may be worked out analytically for a solitary wave solution of amplitude 0.3 by

$$\begin{aligned} I_1 &= \frac{6c}{k} = 3.9799497, \\ I_2 &= \frac{12c^2}{k} + \frac{48kc^2\mu}{5} = 0.81046249 \quad \text{and} \\ I_3 &= \frac{36c^2}{k} + \frac{144c^3}{5k} = 2.579007. \end{aligned} \tag{3.4}$$

The program was run to the time $t = 20$ and the results of error norms and constants of motion I_1, I_2 and I_3 at time $t = 20$ are summarised in Table 1. During the run, numerical invariants remained almost the same when compared with the analytical values (3.4). In Figure 1 (a), the initial function and the numerical solution are graphed for visual representation of the solution at time $t = 20$. The difference between the exact and the numerical solution is depicted in Figure 1 (b) to observe the error distribution over the domain from which the maximum error occurred just around the peak position of wave amplitude.

Downloaded from https://www.cambridge.org/core. Abdullah Gul Universitesi, on 02 Apr 2020 at 12:10:21, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1446181100009822

TABLE 2. Invariants and error norms for single solitary wave, amplitude 0.09, $h = 0.125$, $\Delta t = 0.1$, $-40 \leqslant x \leqslant 60$.

Time	$L_2 \times 10^3$	$L_\infty \times 10^3$	I_1	I_2	I_3
0	0	0	2.10702	0.127302	0.388804
4	0.149	0.193	2.10838	0.127294	0.388781
8	0.280	0.141	2.10928	0.127294	0.388782
12	0.397	0.151	2.10981	0.127294	0.388782
16	0.479	0.213	2.10981	0.127294	0.388782
20	0.547	0.431	2.10900	0.127294	0.388781

TABLE 3.

Amplitude 0.3				Amplitude 0.09			
h	Δt	$L_2 \times 10^3$	$L_\infty \times 10^3$	h	Δt	$L_2 \times 10^3$	$L_\infty \times 10^3$
0.025	0.025	0.03133	0.01267	0.025	0.025	1.10984	0.43138
0.05	0.05	0.07552	0.02819	0.05	0.05	0.92754	0.43158
0.125	0.1	0.30141	0.11405	0.125	0.1	0.54726	0.43151
0.25	0.2	1.19736	0.45444	0.25	0.2	0.28658	0.43151
0.5	0.4	4.82429	1.82587	0.5	0.4	0.53640	0.43151
1.0	0.8	19.2379	7.15536	1.0	0.8	1.75726	0.54777
4.0	0.8	86.8661	30.0064	4.0	0.8	4.44220	1.27680

Corresponding simulation for the solitary wave with smaller amplitude 0.09 was carried out for further verification of accuracy of the numerical methods. Invariants and error norms are also recorded at the same selected times in Table 2. At time $t = 20$, the error norms are satisfactorily small at $L_\infty = 0.431 \times 10^{-3}$ and $L_2 = 0.547 \times 10^{-3}$. Changes of invariants I_1, I_2 and I_3 from time $t = 0$ to time $t = 20$ are calculated from the values in Table 2 and found less than 0.2, 0.0008 and 0.003 percent respectively. The solitary wave solution of amplitude 0.09 is graphed at time $t = 20$ in Figure 2 (a). As seen from the graph, the solitary wave migrated by keeping its shape during the computer run. The error distribution is illustrated in Figure 2 (b). The maximum error occurred at the right boundary of the domain at time $t = 20$. This error resulted from the right artificial boundary condition, which is taken as 0, whereas the analytical solution at the right boundary is 0.00043. Various time-step combinations are used for a further verification of the algorithm to show accuracy numerically in Table 3. We find that higher accuracy is obtained than the results in the paper [7, Table III] when smaller space-time steps were used. For instance, in paper [7], when $h = 0.025$ and $\Delta t = 0.025$, error norms $L_2 = 19.9 \times 10^{-3}$ and $L_\infty = 5.87 \times 10^{-3}$ were obtained at time $t = 40$ whereas we get $L_2 = 1.110 \times 10^{-3}$ and $L_\infty = 0.431 \times 10^{-3}$.

Downloaded from https://www.cambridge.org/core. Abdullah Gul Universitesi, on 02 Apr 2020 at 12:10:21, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1446181100009822

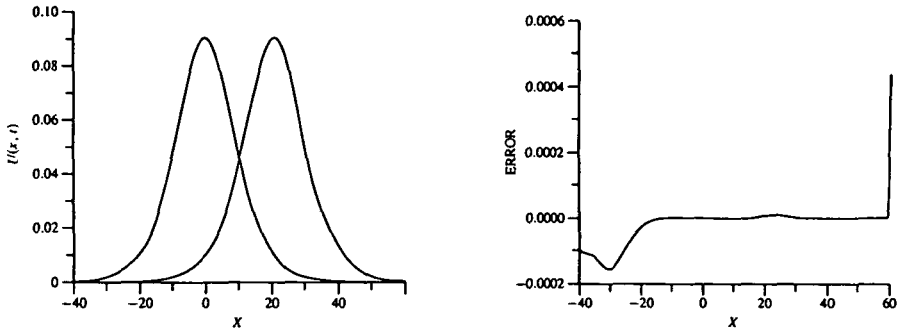


FIGURE 2. (a) Graphs of the initial solution at times $t = 0$ and $t = 20$. (b) Error = exact − numerical solution at time $t = 20$.

Secondly, we study the undular bore development using the initial condition

$$U(x, 0) = 0.5\,U_0\big[1 - \tanh\big((x - x_c)/d\big)\big]$$

and boundary conditions $U(a, t) = U_0$ and $U(b, t) = 0$, where $U(x, 0)$ denotes the elevation of the water above the equilibrium surface at time $t = 0$. Here d represents the slope between the still water and deep water. The change in water level of magnitude $U(x, 0)$ is centred on $x = x_c$. Physical boundary conditions

$$U \xrightarrow{x\to\infty} 0 \quad \text{and} \quad U \xrightarrow{x\to-\infty} U_0$$

are replaced by artificial boundary conditions so as to be able to apply the numerical methods.

We have chosen parameters $U_0 = 0.1$, $\varepsilon = 3/2$, $\mu = 0.16666667$, $h = 0.24$ and $\Delta t = 0.1$ over the domain $-60 \le x \le 540$ to compare the results with previous ones [6, 10]. The program is run with gentle profile $d = 5$ up to time $t = 400$. Numerical quantities and the magnitude of the leading undulation are demonstrated in Table 4. Here x shows the peak position of the leading undulation in the same table. Theoretical variations of the invariants I_1, I_2, I_3

$$\frac{d}{dt}I_1 = \frac{d}{dt}\int_{-\infty}^{\infty} U\,dx = U_0 + \frac{\varepsilon}{2}U_0^2 = 0.1075,$$

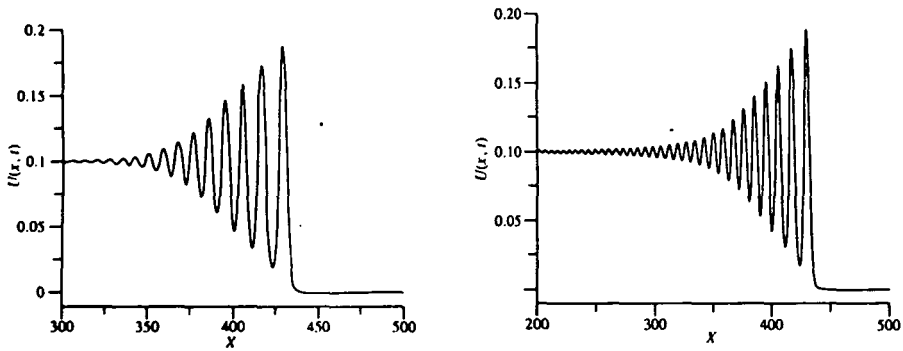
$$\frac{d}{dt}I_2 = \frac{d}{dt}\int_{-\infty}^{\infty} (U^2 + \mu(U_x)^2)\,dx = U_0^2 + \frac{2\varepsilon}{3}U_0^3 = 0.011 \quad \text{and}$$

$$\frac{d}{dt}I_3 = \frac{d}{dt}\int_{-\infty}^{\infty} (U^3 + 3U^2)\,dx = 3U_0^2 + (1 + 2\varepsilon)U_0^3 + \frac{3\varepsilon}{4}U_0^4 = 0.034113$$

Downloaded from https://www.cambridge.org/core. Abdullah Gul Universitesi, on 02 Apr 2020 at 12:10:21, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1446181100009822

TABLE 4. Results for an undular bore $U_0 = 0.1$.

Time	U_{max}	x	I_1	I_2	I_3
0	0.10000	−60	6.01198	0.57630	1.78496
50	0.11031	48.480	11.38698	1.12631	3.49045
100	0.13694	102.24	16.76198	1.67632	5.19570
150	0.15754	156.24	22.13698	2.22635	6.90068
200	0.17029	210.48	27.51198	2.77638	8.60551
250	0.17786	264.96	32.88698	3.32642	10.31029
300	0.18208	319.44	38.26198	3.87645	12.01504
350	0.18508	374.15	43.63698	4.42649	13.71979
400	0.18686	428.87	49.01198	4.97652	15.42453



(a) $d = 5$ (b) $d = 2$

FIGURE 3. Initial condition and undulation profile at $t = 400$, $U_0 = 0.1$, $h = 0.24$ and $\Delta t = 0.1$.

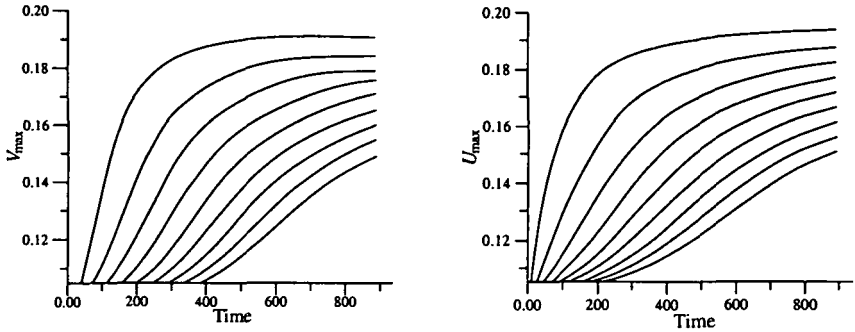
can be worked out to compare with the numerical values:

$$\frac{d}{dt}I_1 = 0.1075, \quad \frac{d}{dt}I_2 = 0.011001 \quad \text{and} \quad \frac{d}{dt}I_3 = 0.034099$$

so that the numerical change in the invariants is found to be very small. In Figure 3, development of undular bores for both the gentle slope $d = 5$ and the steep slope $d = 2$ are drawn at time $t = 400$.

The program is rerun by extending the domain to $-60 \leq x \leq 900$ to observe the behaviour of the undulations in a long runtime. The magnitudes of the successive nine leading undulations against time are depicted in Figure 4 from which the progress of the undular bores is smooth and stable for both the gentle and steep slopes up to time $t = 900$. Amplitudes of the successive nine leading undulations at time $t = 900$ are documented in Table 5. By that time, after a short incubation period, undulations begun to appear successively and have grown in time, then the principal undulation

Downloaded from https://www.cambridge.org/core. Abdullah Gul Universitesi, on 02 Apr 2020 at 12:10:21, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1446181100009822



(a) $d = 5$ (b) $d = 2$

FIGURE 4. Development of the undulations.

TABLE 5. Amplitudes of undulations at time $t = 900$.

Undulation	Amp.($d = 5$)	Amp.($d = 2$)
First	0.1907	0.1938
Second	0.1839	0.1882
Third	0.1788	0.1825
Fourth	0.1753	0.1774
Fifth	0.1708	0.1721
Sixth	0.1652	0.1668
Seventh	0.1599	0.1617
Eighth	0.1545	0.1565
Ninth	0.1491	0.1515

moves to right by attaining their magnitude, that is, the principal undulation shows solitary wave-like behaviour. This can be seen in Figure 4. For example, the first leading undulation attains an amplitude of 0.1907 for the gentle slope, and of 0.1938 for the steep slope at time $t = 900$. When $d = 5$, the magnitude of the first leading undulation remains almost the same after about the time $t = 600$.

A numerical solution algorithm for the RLW equation has been constructed using the collocation method with cubic splines as interpolation functions. Since error norms have been found to be small enough, single solitary wave motion is well presented. Instabilities are not observed for undular bore development simulation in a long run of the algorithm. Therefore the cubic spline collocation procedure is advisable for obtaining the numerical solution of the RLW equation.

References

[1] J. H. Ahlberg, E. N. Nilson and J. L. Walsh, *The theory of splines and their application* (Academic Press, New York, 1967).

Downloaded from https://www.cambridge.org/core. Abdullah Gul Universitesi, on 02 Apr 2020 at 12:10:21, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1446181100009822

[2] T. B. Benjamin, J. L. Bona and J. J. Mahony, "Model equations for long waves in non-linear dispersive systems", *Phil. Trans. Roy. Soc. London* **A272** (1972) 47–78.

[3] D. Bhardwaj and R. Shankar, "A computational method for regularised long wave equation", *Comp. Math. Appl.* **40** (2000) 1397–1404.

[4] J. L. Bona, W. G. Pritchard and L. R Scott, "An evaluation of a model equation for water waves", *Phil. Trans. Roy. Soc. London* **A302** (1981) 457–510.

[5] J. Caldwell, "Application of cubic splines to the nonlinear Burgers' equation", in *Numerical Methods for Nonlinear Problems*, Vol 3 (ed. E. Hinton *et al.*), (Pineridge Press, Swansea, 1989) 253–261.

[6] İ. Dağ, "Least squares quadratic B-spline finite element method for the regularized long wave equation", *Comp. Methods Appl. Mech. Eng* **182** (2000) 205–215.

[7] A. Doğan, "Numerical solution of regularized long wave equation using Petrov-Galerkin method", *Commum. Numer. Methods Eng.* **17** (2001) 485–494.

[8] L. R. T. Gardner and G. A. Gardner, "Solitary waves of the regularised long wave equation", *J. Comput. Phys.* **91** (1990) 441–459.

[9] L. R. T. Gardner, G. A. Gardner and İ. Dağ, "A B-spline finite element method for the regularized long wave equation", *Commum. Numer. Methods Eng.* **11** (1995) 59–68.

[10] L. R. T. Gardner, G. A. Gardner and A. Doğan, "A least squares finite element scheme for the RLW equation", *Commum. Numer. Methods Eng.* **12** (1996) 795–804.

[11] P. C. Jain, R. Shankar and T. V. Singh, "Numerical solution of regularized long-wave equation", *Commum. Numer. Methods Eng.* **9** (1993) 579–586.

[12] P. J. Olver, "Euler operators and conservation laws of the BBM equation", *Math. Proc. Camb. Phil. Soc.* **85** (1979) 143–160.

[13] D. H. Peregrine, "Calculations of the development of an undular bore", *J. Fluid. Mech.* **25** (1966) 321–330.

[14] S. G. Rubin and R. A. Graves, "Cubic spline approximation for problems in fluid mechanics", NASA TR R-436, Washington, DC, 1975.

[15] S. G. Rubin and P. K. Khosla, "Higher-order numerical solutions using cubic splines", *AIAA J.* **14** (1976) 851–858.

[16] P. Wang and R. Kahawita, "Numerical integration of partial differential equations using cubic splines", *Intern. J. Computer Math.* **13** (1983) 271–286.

Downloaded from https://www.cambridge.org/core. Abdullah Gul Universitesi, on 02 Apr 2020 at 12:10:21, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1446181100009822