Journal homepage www.jzs.univsul.edu.iq
Journal of Zankoy Sulaimani
Part-A- (Pure and Applied Sciences)

Numerical Solution of Nonlinear Whitham-Broer-Kaup Shallow Water Model Using Finite Difference Methods

Rostam K. Saeed[1] and Mohammed I. Sadeeq[2]

1 Salahaddin University/Erbil-College of Science-Department of Mathematics, Kurdistan Region, Erbil-Iraq,
2 Duhok University-College of Education/Akre-Department of Mathematics, Kurdistan Region, Duhok-Iraq.
E-mail: rostam.saeed@su.edu.krd, mohammedmath87@yahoo.com.

Article info

Original: 29 May 2016
Revised: 15 September 2016
Accepted: 16 October 2016

Key Words:
Finite difference method, Whitham-Broer-Kaup shallow water model.

Abstract

In this paper, we presented finite difference methods for solving nonlinear Whitham-Broer-Kaup (WBK) shallow water model numerically. We first subdivided the domain of the model by a net with a finite number of mesh points, and the derivative at each point replaced by explicit, Crank-Nicolson, and exponential finite difference approximations. The result is the system of algebraic equations which when solved, provide an approximation to the solutions of WBK model at the selected grid points. Also, a comparison has been made between the approximate solutions obtained by the proposed methods and the exact solutions. Numerical results represented in tables and figures with the help of MATLAB R2015a.

Introduction

Non-linear partial differential equations (PDEs) play an important role in various areas of modern physics and engineering. There are PDEs for which the solution concerning formulas is so complex that one often prefers to use a numerical method to solve such equations. Finite difference methods (FDMs) are numerical methods for approximating the solutions of PDEs, using the finite difference equations to approximate the derivatives, and they were found to be discrete techniques. Consequently, a FDM yields a solution only at discrete points in the domain of interest rather than, as we expect for an analytic calculation a formula or closed-form solution valid at all points [9, 10]. For more details about the numerical methods, see [11-16].

To understand finite difference techniques, it is first necessary to consider the nomenclature and fundamental concepts encountered in this form of approximation theory. The basic ideas are quite simple; a net first subdivides the domain of solution of the given PDE with a finite number of mesh points, and a finite difference approximation then replaces the derivative at each point. Alternatively, one can imagine this discretization procedure as the replacement of the solution of the PDE with an interpolating polynomial and the differentiation of this polynomial. Many option contract the values can obtain by solving PDEs with certain initial and boundary conditions, the most standard FDMs for solving PDE or system of PDEs are the explicit method, implicit method, and Crank-Nicolson method [8, 10].

The explicit method seems to be easier than Crank-Nicolson and implicit methods to solve problems in terms of time and effort spent on a computer. This satisfied, because in the explicit method, only one unknown involved in the use of the finite difference approximation. Implicit and Crank-Nicolson methods require an indirect calculation, usually involving a system of equations, which can be solved at each step and computationally severe on a fine mesh. In general, Crank-Nicolson and implicit methods may need more computation, but they have a significant advantage of being more stable than explicit method [8, 9].

Taylor series expansions play a significant role in the formulation and classification of finite difference schemes. The fundamental idea of FDM to solve problems numerically is to exchange the partial derivatives by approximations obtained by Taylor series expansions, for full details see [10, 18].

Here, we consider the nonlinear WBK model, which has been studied by Whitham, Broer, and Kaup [5], the equations describe the propagation of shallow water waves, with different dispersion relations. The nonlinear WBK model is as follows [5]:

$$\begin{aligned} &u_t + v_x + uu_x + \beta u_{xx} = 0, \\ &v_t + vu_x + uv_x - \beta v_{xx} + \alpha u_{xxx} = 0, \end{aligned} \tag{1}$$

where $u = u(x,t)$ is the horizontal velocity, $v = v(x,t)$ is the height that deviates from equilibrium position of the liquid, x is the scaled space, t is the scaled time, and α, β are constants which are represented in different diffusion powers [19,21]. If $\alpha = 0, \beta \neq 0$, model (1) reduces to the classical long-wave equations that describe shallow water waves with diffusion [7]. If $\alpha = 1, \beta = 0$, model (1) becomes the variant Boussinesq equation [1].

The exact solutions of WBK model (1), given by [21] as follows:

$$\begin{aligned} u(x,t) &= \lambda + \mu_1 \tanh(\omega\xi + \gamma), \\ v(x,t) &= \mu_2 - \mu_2 \tanh^2(\omega\xi + \gamma), \end{aligned} \tag{2}$$

with,

$$\begin{aligned} \mu_1 &= 2\eta\omega\sqrt{\alpha + \beta^2}\,, \\ \mu_2 &= 2\omega^2\left(\alpha + \beta^2 - \eta\beta\sqrt{\alpha + \beta^2}\,\right), \end{aligned}$$

where, $\xi = x - \lambda t, \alpha + \beta^2 > 0, \omega \neq 0, \lambda \neq 0, \eta = \pm 1$ and γ is an arbitrary real constant.

This work studies the numerical solutions of this model on the domain, $a \leq x \leq b$, and $0 \leq t \leq T$ employing FDM. We divide the (x,t)-plane into a network of rectangles of sides $\Delta x = h = (b-a)/n$, and $\Delta t = k = T/m$ by drawing the set of lines:

$$\begin{aligned} x_p &= a + ph, \qquad p = 0,1,\dots,n, \\ t_q &= qk, \qquad\quad\; q = 0,1,\dots,m. \end{aligned}$$

The points of intersection of these families of lines are called *Grid Points* (sometimes referred to as Lattice Points or Mesh Points). The general procedure to solve PDEs or system by finite difference approximation is to obtain the solution at these grid points. These grid points for two dimensions are of the type [18]: square, rectangular, triangular, and hexangular. The discussion of the problems involving triangular and hexangular is more complicated. Hence, we confine to either square or rectangular zones, in most of cases.

The initial conditions are taken from the exact solutions (2) at the value $t = 0$, as follows:

$$\begin{aligned} u(x,0) &= \lambda + \mu_1 \tanh(\omega x + \gamma), \\ v(x,0) &= \mu_2 - \mu_2 \tanh^2(\omega x + \gamma). \end{aligned} \tag{3}$$

Similarly, the Dirichlet boundary conditions at $x = a$, and $x = b$, as follows:

$$\begin{aligned} u(a,t) &= \lambda + \mu_1 \tanh(\omega(a - \lambda t) + \gamma), \\ u(b,t) &= \lambda + \mu_1 \tanh(\omega(b - \lambda t) + \gamma), \\ v(a,t) &= \mu_2 - \mu_2 \tanh^2(\omega(a - \lambda t) + \gamma), \\ v(a,t) &= \mu_2 - \mu_2 \tanh^2(\omega(b - \lambda t) + \gamma), \end{aligned} \tag{4}$$

and the Neumann boundary conditions:

$$\begin{aligned} \left.\frac{\partial u}{\partial x}\right|_{x=a} &= \omega\mu_1 \operatorname{sech}^2(\omega(a - \lambda t) + \gamma) = \check{u}(t), \\ \left.\frac{\partial u}{\partial x}\right|_{x=b} &= \omega\mu_1 \operatorname{sech}^2(\omega(b - \lambda t) + \gamma) = \hat{u}(t), \\ \left.\frac{\partial v}{\partial x}\right|_{x=a} &= -2\omega\mu_2 \tanh(\omega(a - \lambda t) + \gamma) \operatorname{sech}^2(\omega(a - \lambda t) + \gamma) = \check{v}(t), \\ \left.\frac{\partial v}{\partial x}\right|_{x=b} &= -2\omega\mu_2 \tanh(\omega(b - \lambda t) + \gamma) \operatorname{sech}^2(\omega(b - \lambda t) + \gamma) = \hat{v}(t). \end{aligned} \tag{5}$$

Explicit Method

In this section, the explicit method implemented to solve (1). To approximate the model by this approach, we divide the domain of the model by a set of (equally spaced) lines parallel to the x-axis and t-axis to form a grid. Define the constants h and k to be grid sizes in the x and t directions, respectively. The solution algorithm is simple to set up but requires many time steps to carry out the calculations over a given interval of t. The term "explicit" refers to the fact that only one value of u and v on the line $q+1$ occurs in the finite difference approximation [4, 8]. In contrast, there are some implicit formulations in which two or more values of u and v appear on the line $q+1$, as we shall see later.

We will denote the discrete approximation to the exact solutions $u(x_p, t_q)$ and $v(x_p, t_q)$ at the grid points (x_p, t_q) by $U_{p,q}$ and $V_{p,q}$, respectively. Then approximate the time derivatives u_t and v_t by forward-time difference, and the space derivatives u_x, v_x, u_{xx}, v_{xx}, and u_{xxx} by central-space difference approximation, is then follow that:

$$\frac{U_{p,q+1}-U_{p,q}}{k}+\frac{V_{p+1,q}-V_{p-1,q}}{2h}+U_{p,q}\left(\frac{U_{p+1,q}-U_{p-1,q}}{2h}\right)+\beta\left(\frac{U_{p+1,q}-2U_{p,q}+U_{p-1,q}}{h^2}\right)=0,$$

$$\frac{V_{p,q+1}-V_{p,q}}{k}+V_{p,q}\left(\frac{U_{p+1,q}-U_{p-1,q}}{2h}\right)+U_{p,q}\left(\frac{V_{p+1,q}-V_{p-1,q}}{2h}\right)-\beta\left(\frac{V_{p+1,q}-2V_{p,q}+V_{p-1,q}}{h^2}\right)$$
$$+\alpha\left(\frac{U_{p+2,q}-2U_{p+1,q}+2U_{p-1,q}-U_{p-2,q}}{2h^3}\right)=0.$$

Simplify the above equations, we get:

$$\begin{aligned}
U_{p,q+1} &= U_{p,q}-r_1\left[U_{p,q}\left(U_{p+1,q}-U_{p-1,q}\right)+V_{p+1,q}-V_{p-1,q}\right]-\ \beta r_2\left(U_{p+1,q}-2U_{p,q}+U_{p-1,q}\right),\\
V_{p,q+1} &= V_{p,q}-r_1\left[V_{p,q}\left(U_{p+1,q}-U_{p-1,q}\right)+U_{p,q}\left(V_{p+1,q}-V_{p-1,q)}\right]+\ \beta r_2\left(V_{p+1,q}-2V_{p,q}+V_{p-1,q}\right)\\
&\quad -\alpha r_3\left(U_{p+2,q}-2U_{p+1,q}+2U_{p-1,q}-U_{p-2,q}\right),
\end{aligned}\tag{6}$$

where, $r_1 = k/2h$, and $r_2 = k/h^2$, and $r_3 = k/2h^3$.

If we put $p = 1$ and $p = n-1$ into the second equation of (6), two fictitious terms $U_{-1,q}$ and $U_{n+1,q}$ are obtained respectively. Then using central difference equation for approximating the Neumann boundary conditions (5) at (x_0, t_q) to eliminate $U_{-1,q}$, and at (x_n, t_q) to eliminate $U_{n+1,q}$, as follows:

$$u_x(x_0, t_q) \approx \frac{U_{1,q}-U_{-1,q}}{2h} = \breve{u}(t_q),$$
$$u_x(x_n, t_q) \approx \frac{U_{n+1,q}-U_{n-1,q}}{2h} = \hat{u}(t_q).$$

Simplify the last two equations, yields:

$$U_{1,q}-U_{-1,q} = 2h\breve{u}(t_q),$$
$$U_{n+1,q}-U_{n-1,q} = 2h\hat{u}(t_q),$$

or,

(7)

Thus, the finite difference schemes described by (6) are known as the explicit schemes (explicit finite difference approximations) for (1), since it permits the direct or explicit calculation of numerical solutions $U_{p,q+1}$ and $V_{p,q+1}$ from the data at the preceding time step.

The numerical solutions U and V of (1) at the grid points (x_p, t_q) can obtained by using (6). This satisfied, by computing $U_{p,q+1}$ and $V_{p,q+1}$, $p = 1, \dots, n-1$, successively for each $q = 0, \dots, m-1$. The values on the side boundaries and lower boundary are given by boundary conditions (4) and initial conditions (3), respectively. Equivalently, we can find the unknown pivotal values of U and V along the first time row $t = k$, in terms of known boundary and initial values along $t = 0$. Then compute the unknown pivotal values along the second time row concerning the calculated pivotal values along the first, and so on.

Crank-Nicolson Method

Crank and Nicolson in (1947) invented the Crank-Nicolson finite difference scheme [10]. They initially applied it to the heat equation, and they approximated the derivatives space and time by finite difference approximation. Crank-Nicolson method has higher accuracy and better stability properties, but requires sophisticated algorithms for solving a system of equations at every time level [8, 9, 20].

In this section, we solve (1) by Crank-Nicolson method, to do this, replace the time derivatives u_t and v_t by forward-time difference and approximate $u_x, v_x, u_{xx}, v_{xx},$ and u_{xxx} by Crank-Nicolson expression. Likewise, approximate the nonlinear terms $uu_x, uv_x,$ and vu_x by central difference approximation at t_q and t_{q+1}, so that the scheme will remains linear at t_{q+1}. Then the discretization of (1) gives:

$$\begin{aligned}
\frac{U_{p,q+1}-U_{p,q}}{k} &+ \frac{1}{4h}\left[V_{p+1,q+1}-V_{p-1,q+1}+V_{p+1,q}-V_{p-1,q}\right]\\
&+\frac{1}{4h}\left[U_{p,q}\left(U_{p+1,q+1}-U_{p-1,q+1}\right)+U_{p,q+1}\left(U_{p+1,q}-U_{p-1,q}\right)\right]\\
&+\frac{\beta}{2h^2}\left[U_{p+1,q+1}-2U_{p,q+1}+U_{p-1,q+1}+U_{p+1,q}-2U_{p,q}+U_{p-1,q}\right]=0,\\
\frac{V_{p,q+1}-V_{p,q}}{k} &+ \frac{1}{4h}\left[V_{p,q}\left(U_{p+1,q+1}-U_{p-1,q+1}\right)+V_{p,q+1}\left(U_{p+1,q}-U_{p-1,q}\right)\right]\\
&+\frac{1}{4h}\left[U_{p,q}\left(V_{p+1,q+1}-V_{p-1,q+1}\right)+U_{p,q+1}\left(V_{p+1,q}-V_{p-1,q}\right)\right]\\
&-\frac{\beta}{2h^2}\left[V_{p+1,q+1}-2V_{p,q+1}+V_{p-1,q+1}+V_{p+1,q}-2V_{p,q}+V_{p-1,q}\right]\\
&+\frac{\alpha}{4h^3}\left[U_{p+2,q+1}-2U_{p+1,q+1}+2U_{p-1,q+1}-U_{p-2,q+1}+U_{p+2,q}\right.\\
&\left.-2U_{p+1,q}+2U_{p-1,q}-U_{p-2,q}\right]=0.
\end{aligned}\tag{8}$$

Rearranging (8), and then for simply assume: $s_1=\frac{k}{4h}$, $s_2=\frac{k}{2h^2}$, $s_3=\frac{k}{4h^3}$, $\varepsilon=\alpha s_3$, $A_p^q=\beta s_2-s_1U_{p,q}$, $B_p^q=1+s_1U_{p+1,q}-s_1U_{p-1,q}-2\beta s_2$, $C_p^q=s_1U_{p,q}+\beta s_2$, $D_p^q=2\varepsilon-s_1V_{p,q}$, $E_p^q=s_1V_{p,q}-2\varepsilon$, $F_p^q=-\left(\beta s_2+s_1U_{p,q}\right)$, $G_p^q=1+s_1U_{p+1,q}-s_1U_{p-1,q}+2\beta s_2$, $S_p^q=s_1U_{p,q}-\beta s_2$, $H_p^q=s_1V_{p+1,q}-s_1V_{p-1,q}$, $J_1^q=-\varepsilon+H_1^q$, and $J_{n-1}^q=\varepsilon+H_{n-1}^q$, we get the following difference equations:

$$\begin{aligned}
&A_p^qU_{p-1,q+1}+B_p^qU_{p,q+1}+C_p^qU_{p+1,q+1}-s_1V_{p-1,q+1}+s_1V_{p+1,q+1} = U_{p,q}-s_1\left(V_{p+1,q}-V_{p-1,q}\right)\\
&\qquad -\beta s_2\left(U_{p+1,q}-2U_{p,q}+U_{p-1,q}\right),\\
&-\varepsilon U_{p-2,q+1}+D_p^qU_{p-1,q+1}+H_p^qU_{p,q+1}+E_p^qU_{p+1,q+1}+\varepsilon U_{p+2,q+1}+F_p^qV_{p-1,q+1}+G_p^qV_{p,q+1}\\
&\qquad +S_p^q\,V_{p+1,q+1}=V_{p,q}+\beta s_2\left(V_{p+1,q}-2V_{p,q}+V_{p-1,q}\right)-\varepsilon\left(U_{p+2,q}-2U_{p+1,q}\right.\\
&\qquad \left.+2U_{p-1,q}-U_{p-2,q}\right).
\end{aligned}\tag{9}$$

Therefore, the finite difference schemes described by (9) represent the Crank-Nicolson schemes (Crank-Nicolson type schemes) for (1). The left sides of (9) contain unknown, and the right side known values.

If there are n internal mesh points along each time row, then for $q=0$ and $p=1,\dots,n-1$, the difference schemes (9) gives $2(n-1)$ simultaneous equations for $2(n-1)$ unknown pivotal values $(U_{1,q+1}, V_{1,q+1}, U_{2,q+1}, V_{2,q+1},\dots,U_{n-1,q+1}, V_{n-1,q+1})$ along the first time row in terms of known initial and boundary values. Similarly, $q=1$ expresses $2(n-1)$ unknown pivotal values along the second time row regarding the calculated values along the first time row, etc.

Now, to obtain linear simultaneous equations, put $p=1,\dots,n-1$, into (9), respectively. Then, we use the initial conditions (3) and boundary conditions (4) to find the initial values and boundary values, respectively. On the other hand, in similar manner of explicit method using the Neumann boundary conditions (5) to eliminate the fictitious terms $U_{-1,q}, U_{-1,q+1}, U_{n+1,q},$ and $U_{n+1,q+1}$ (see equation (7)).

Hence, the result is a linear system of equations and it can be written in the matrix form as a multi-diagonal linear system:

$$\boldsymbol{AX} = \boldsymbol{B}, \tag{10}$$

where, $\boldsymbol{A} =$

$$\begin{bmatrix}
B_1^q & 0 & C_1^q & s_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
J_1^q & G_1^q & E_1^q & S_1^q & \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
A_2^q & -s_1 & B_2^q & 0 & C_2^q & s_1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
D_2^q & F_2^q & H_2^q & G_2^q & E_2^q & S_2^q & \varepsilon & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & A_3^q & -s_1 & B_3^q & 0 & C_3^q & s_1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\varepsilon & 0 & D_3^q & F_3^q & H_3^q & G_3^q & E_3^q & S_3^q & \varepsilon & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & A_4^q & -s_1 & B_4^q & 0 & C_4^q & s_1 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -\varepsilon & 0 & D_4^q & F_4^q & H_4^q & G_4^q & E_4^q & S_4^q & \varepsilon & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & A_{n-2}^q & -s_1 & B_{n-2}^q & 0 & C_{n-2}^q & s_1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & -\varepsilon & 0 & D_{n-2}^q & F_{n-2}^q & H_{n-2}^q & G_{n-2}^q & E_{n-2}^q & S_{n-2}^q \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & A_{n-1}^q & -s_1 & B_{n-1}^q & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & -\varepsilon & 0 & D_{n-1}^q & F_{n-1}^q & J_{n-1}^q & G_{n-1}^q
\end{bmatrix},$$

$\boldsymbol{B} =$

$$\begin{bmatrix}
U_{1,q} - s_1\left(V_{2,q} - V_{0,q}\right) - \beta s_2\left(U_{2,q} - 2U_{1,q} + U_{0,q}\right) - A_1^q U_{0,q+1} + s_1 V_{0,q+1} \\
V_{1,q} + \beta s_2\left(V_{2,q} - 2V_{1,q} + V_{0,q}\right) - \varepsilon\left(U_{3,q} - 2U_{2,q} + 2U_{0,q} - U_{1,q} + 2h\breve{u}(t_q)\right) \\
\qquad\qquad -D_1^q U_{0,q+1} - F_1^q V_{0,q+1} - 2h\varepsilon\breve{u}(t_{q+1}) \\
U_{2,q} - s_1\left(V_{3,q} - V_{1,q}\right) - \beta s_2\left(U_{3,q} - 2U_{2,q} + U_{1,q}\right) \\
V_{2,q} + \beta s_2\left(V_{3,q} - 2V_{2,q} + V_{1,q}\right) - \varepsilon\left(U_{4,q} - 2U_{3,q} + 2U_{1,q} - U_{0,q}\right) + \varepsilon U_{0,q+1} \\
U_{3,q} - s_1\left(V_{4,q} - V_{2,q}\right) - \beta s_2\left(U_{4,q} - 2U_{3,q} + U_{2,q}\right) \\
V_{3,q} + \beta s_2\left(V_{4,q} - 2V_{3,q} + V_{2,q}\right) - \varepsilon\left(U_{5,q} - 2U_{4,q} + 2U_{2,q} - U_{1,q}\right) \\
U_{4,q} - s_1\left(V_{5,q} - V_{3,q}\right) - \beta s_2\left(U_{5,q} - 2U_{4,q} + U_{3,q}\right) \\
V_{4,q} + \beta s_2\left(V_{5,q} - 2V_{4,q} + V_{3,q}\right) - \varepsilon\left(U_{6,q} - 2U_{5,q} + 2U_{3,q} - U_{2,q}\right) \\
\vdots \\
U_{n-2,q} - s_1\left(V_{n-1,q} - V_{n-3,q}\right) - \beta s_2\left(U_{n-1,q} - 2U_{n-2,q} + U_{n-3,q}\right) \\
V_{n-2,q} + \beta s_2\left(V_{n-1,q} - 2V_{n-2,q} + V_{n-3,q}\right) - \varepsilon\left(U_{n,q} - 2U_{n-1,q} + 2U_{n-3,q} - U_{n-4,q}\right) - \varepsilon U_{n,q+1} \\
U_{n-1,q} - s_1\left(V_{n,q} - V_{n-2,q}\right) - \beta s_2\left(U_{n,q} - 2U_{n-1,q} + U_{n-2,q}\right) - s_1 V_{n,q+1} - C_{n-1}^q U_{n,q+1} \\
V_{n-1,q} + \beta s_2\left(V_{n,q} - 2V_{n-1,q} + V_{n-2,q}\right) - \varepsilon\left(U_{n-1,q} + 2h\hat{u}(t_q) - 2U_{n,q} + 2U_{n-2,q} - U_{n-3,q}\right) \\
\qquad\qquad -E_{n-1}^q U_{n,q+1} - S_{n-1}^q V_{n,q+1} - 2h\varepsilon\hat{u}(t_{q+1})
\end{bmatrix},$$

and $\boldsymbol{X} = \left[U_{1,q+1}, V_{1,q+1}, U_{2,q+1}, V_{2,q+1}, \dots, U_{n-1,q+1}, V_{n-1,q+1}\right]^T$.

Therefore, for each $q = 0, \dots, m-1$, successively, we must solve the diagonal linear system (10) by any suitable method to obtain numerical solutions U and V of (1), at the lattice (grid) points (x_p, t_q) on the qth time row.

Exponential Finite Difference Method

This section is devoted to solve (1), associated with initial conditions (3), boundary conditions (4), and Neumann conditions (5), by the exponential finite difference algorithm, was originally developed by Bhattachary [3]. He used the exponential finite difference technique to solve one-dimensional heat conduction in a solid slab. In [2], Bahadir applied this method to solve the Korteweg-de Vries (KdV) equation. In [6], some classes of nonlinear PDEs (Burger's equation and boundary layer equations) are solved by this approach.

For solving (1) by exponential finite difference technique. Multiply the first equation of (1) by the derivative of $\phi(u)$ and the second equation by the derivative of $\theta(v)$, where $\phi(u)$ and $\theta(v)$ denotes any continuous and differentiable functions, we get:

$$\begin{aligned}\frac{d\phi(u)}{du}\frac{\partial u}{\partial t} &= -\phi'(u)(v_x + uu_x + \beta u_{xx}),\\ \frac{d\theta(v)}{dv}\frac{\partial v}{\partial t} &= -\theta'(v)(vu_x + uv_x - \beta v_{xx} + \alpha u_{xxx}),\end{aligned}$$

or,

$$\begin{aligned}\frac{\partial\phi(u)}{\partial t} &= -\phi'(u)(v_x + uu_x + \beta u_{xx}),\\ \frac{\partial\theta(v)}{\partial t} &= -\theta'(v)(vu_x + uv_x - \beta v_{xx} + \alpha u_{xxx}).\end{aligned} \tag{11}$$

Next, we approximate the time derivatives in (11) by forward-time difference approximation, and using the central-space difference approximation for the space derivatives, as follows:

$$\begin{aligned}\frac{\phi(U_{p,q+1}) - \phi(U_{p,q})}{k} &= -\phi'(U_{p,q})\left[\frac{V_{p+1,q} - V_{p-1,q}}{2h} + U_{p,q}\left(\frac{U_{p+1,q} - U_{p-1,q}}{2h}\right)\right.\\ &\quad \left. +\beta\left(\frac{U_{p+1,q} - 2U_{p,q} + U_{p-1,q}}{h^2}\right)\right],\\ \frac{\theta(V_{p,q+1}) - \theta(V_{p,q})}{k} &= -\theta'(V_{p,q})\left[V_{p,q}\left(\frac{U_{p+1,q} - U_{p-1,q}}{2h}\right) + U_{p,q}\left(\frac{V_{p+1,q} - V_{p-1,q}}{2h}\right)\right.\\ &\quad \left. -\beta\left(\frac{V_{p+1,q} - 2V_{p,q} + V_{p-1,q}}{h^2}\right) + \alpha\left(\frac{U_{p+2,q} - 2U_{p+1,q} + 2U_{p-1,q} - U_{p-2,q}}{2h^3}\right)\right].\end{aligned} \tag{12}$$

Simplifying (12), yields:

$$\begin{aligned}\phi(U_{p,q+1}) &= \phi(U_{p,q}) - \phi'(U_{p,q})\big[r_1(U_{p,q}U_{p+1,q} - U_{p,q}U_{p-1,q} + V_{p+1,q} - V_{p-1,q})\\ &\quad + \beta r_2(U_{p+1,q} - 2U_{p,q} + U_{p-1,q})\big],\\ \theta(V_{p,q+1}) &= \theta(V_{p,q}) - \theta'(V_{p,q})\big[r_1(V_{p,q}U_{p+1,q} - V_{p,q}U_{p-1,q} + U_{p,q}V_{p+1,q} - U_{p,q}V_{p-1,q})\\ &\quad - \beta r_2(V_{p+1,q} - 2V_{p,q} + V_{p-1,q}) + \alpha r_3(U_{p+2,q} - 2U_{p+1,q} + 2U_{p-1,q} - U_{p-2,q})\big],\end{aligned} \tag{13}$$

where, r_1, r_2 and r_3 are as in (6).

To obtain the exponential finite difference scheme, consider [2]

$$\begin{aligned}\phi(u) &= \ln(u),\\ \theta(v) &= \ln(v).\end{aligned} \tag{14}$$

Using (14) in (13), we get:

$$\begin{aligned}\ln(U_{p,q+1}) &= \ln(U_{p,q}) - \frac{1}{U_{p,q}}\big[r_1(U_{p,q}U_{p+1,q} - U_{p,q}U_{p-1,q} + V_{p+1,q} - V_{p-1,q})\\ &\quad + \beta r_2(U_{p+1,q} - 2U_{p,q} + U_{p-1,q})\big],\\ \ln(V_{p,q+1}) &= \ln(V_{p,q}) - \frac{1}{V_{p,q}}\big[r_1(V_{p,q}U_{p+1,q} - V_{p,q}U_{p-1,q} + U_{p,q}V_{p+1,q} - U_{p,q}V_{p-1,q})\\ &\quad - \beta r_2(V_{p+1,q} - 2V_{p,q} + V_{p-1,q}) + \alpha r_3(U_{p+2,q} - 2U_{p+1,q} + 2U_{p-1,q} - U_{p-2,q})\big].\end{aligned} \tag{15}$$

Taking the inverse functions (we know that the inverse of ln(.) is exp(.)) to both sides of (15), is then follow that:

$$\begin{aligned} U_{p,q+1} &= U_{p,q}\exp\left\{\frac{1}{U_{p,q}}\left[r_1\left(U_{p,q}U_{p-1,q}-U_{p,q}U_{p+1,q}-V_{p+1,q}+V_{p-1,q}\right)-\beta r_2\left(U_{p+1,q}\right.\right.\right.\\ &\quad \left.\left.\left.-2U_{p,q}+U_{p-1,q}\right)\right]\right\},\\ V_{p,q+1} &= V_{p,q}\exp\left\{\frac{1}{V_{p,q}}\left[r_1\left(V_{p,q}U_{p-1,q}-V_{p,q}U_{p+1,q}-U_{p,q}V_{p+1,q}+U_{p,q}V_{p-1,q}\right)\right.\right.\\ &\quad \left.\left.+\beta r_2\left(V_{p+1,q}-2V_{p,q}+V_{p-1,q}\right)-\alpha r_3\left(U_{p+2,q}-2U_{p+1,q}+2U_{p-1,q}-U_{p-2,q}\right)\right]\right\}. \end{aligned} \tag{16}$$

Therefore, (16) represents the exponential finite difference schemes for (1). The boundary and initial values are given by Dirichlet boundary conditions (4) and initial conditions (3), respectively. Also, the values of $U_{-1,q}$ and $U_{n+1,q}$ are approximated by using Neumann boundary conditions (5). Thus, the numerical solutions can be obtained in (16), by computing $U_{p,q+1}$ and $V_{p,q+1}$, $p = 1, \dots, n-1$, successively for each $q = 0, \dots, m-1$.

Numerical Examples

In this section, we will apply the proposed methods to calculate the numerical solutions of (1) with the exact solutions (3), as well as with the given initial, Dirichlet and Neumann conditions. The accuracy of the method is tested by computing the absolute error and L_∞ error norm, which are defined as [17]:

$$L_{abs}(u) = \left|u(x_p,t) - U(x_p,t)\right|,\ \ p = 0,\dots,n,$$

$$L_{abs}(v) = \left|v(x_p,t) - V(x_p,t)\right|,\ \ p = 0,\dots,n,$$

$$L_\infty(u) = \max\left|u(x_p,t) - U(x_p,t)\right|,\ \ p = 0,\dots,n,$$

$$L_\infty(v) = \max\left|v(x_p,t) - V(x_p,t)\right|,\ \ p = 0,\dots,n,$$

where u, v are exact solutions, and U, V are numerical solutions of (1).

Example 1:

As a first example, we will solve (1) over the domain $-5 \le x \le 5$ and in the time period $0 \le t \le 1$, with $n = m = 40$, $\alpha = \beta = \lambda = 0.001$, $\omega = 0.4$, $\eta = 1$ and $\gamma = 0$. The space domain and time period are discretized with the values $\Delta x = 0.25$ and $\Delta t = 0.025$, respectively. Numerical results are listed in Tables 1-3, and plotted in Figures 1-5.

Example 2:

We solved (1) in this example over the domain $-10 \le x \le 10$ and in the time period $0 \le t \le 1.5$, with $n = m = 60$, $\alpha = \beta = 0.1$, $\lambda = \omega = 0.01$, $\eta = 1$ and taking the difference values of the parameter γ. The domain and time period are discretized with the values $\Delta x = 0.3$ and $\Delta t = 0.025$, respectively. Numerical results are listed in Tables 4-6.

Table 1: L_{abs} and L_∞ errors for the solutions obtained by the explicit method for Example 1.

	$t = 0.1$		$t = 0.5$		$t = 0.7$	
x	$L_{abs}(u)$	$L_{abs}(v)$	$L_{abs}(u)$	$L_{abs}(v)$	$L_{abs}(u)$	$L_{abs}(v)$
-5	0	0	0	0	0	0
-4.75	1.4719e-09	6.8197e-09	7.5479e-09	3.4107e-08	1.0827e-08	4.7881e-08
-4.5	2.7197e-09	3.0216e-11	2.7189e-08	8.9045e-11	4.7733e-08	3.0088e-10
-4.25	3.2741e-09	1.0717e-10	1.5879e-08	3.8454e-10	2.1660e-08	3.1264e-10
-4	4.7921e-09	2.1056e-10	2.3568e-08	1.1310e-09	3.2654e-08	1.7047e-09
-3.75	6.9143e-09	3.5649e-10	3.4081e-08	1.7826e-09	4.7405e-08	2.4940e-09

-3.5	9.8280e-09	5.5545e-10	4.8465e-08	2.7807e-09	6.7373e-08	3.8945e-09
-3.25	1.3741e-08	8.1291e-10	6.7859e-08	4.0721e-09	9.4408e-08	5.7060e-09
-3	1.8855e-08	1.1266e-09	9.3300e-08	5.6458e-09	1.2993e-07	7.9126E-09
-2.75	2.5323e-08	1.4796e-09	1.2561e-07	7.4183e-09	1.7514e-07	1.0399e-08
-2.5	3.3170e-08	1.8333e-09	1.6499e-07	9.1961e-09	2.3037e-07	1.2895e-08
-2.25	4.2193e-08	2.1210e-09	2.1052e-07	1.0644e-08	2.9442e-07	1.4930e-08
-2	5.1845e-08	2.2473e-09	2.5955e-07	1.1285e-08	3.6359e-07	1.5834e-08
-1.75	6.1145e-08	2.0994e-09	3.0716e-07	1.0551e-08	4.3104e-07	1.4810e-08
-1.5	6.8653e-08	1.5734e-09	3.4607e-07	7.9183e-09	4.8647e-07	1.1124e-08
-1.25	7.2607e-08	6.1674e-10	3.6721e-07	3.1214e-09	5.1702e-07	4.3994e-09
-1	7.1238e-08	7.2517e-10	3.6135e-07	3.6144e-09	5.0951e-07	5.0495e-09
-0.75	6.3252e-08	2.2804e-09	3.2164e-07	1.1427e-08	4.5407e-07	1.6014e-08
-0.5	4.8356e-08	3.7643e-09	2.4634e-07	1.8886e-08	3.4808e-07	2.6488e-08
-0.25	2.7604e-08	4.8486e-09	1.4077e-07	2.4341e-08	1.9901e-07	3.4150e-08
0	3.3724e-09	5.2667e-09	1.7162e-08	2.6450e-08	2.4243e-08	3.7115e-08
0.25	2.1125e-08	4.9129e-09	1.0781e-07	2.4677e-08	1.5247e-07	3.4631e-08
0.5	4.2629e-08	3.8826e-09	2.1726e-07	1.9503e-08	3.0705e-07	2.7370e-08
0.75	5.8648e-08	2.4344e-09	2.9832e-07	1.2228e-08	4.2122e-07	1.7158e-08
1	6.7961e-08	8.9384e-10	3.4484e-07	4.4877e-09	4.8631e-07	6.2932e-09
1.25	7.0695e-08	4.5314e-10	3.5767e-07	2.2794e-09	5.0368e-07	3.2036e-09
1.5	6.8000e-08	1.4298e-09	3.4294e-07	7.1844e-09	4.8218e-07	1.0085e-08
1.75	6.1550e-08	1.9845e-09	3.0938e-07	9.9689e-09	4.3429e-07	1.3990e-08
2	5.3059e-08	2.1641e-09	2.6584e-07	1.0868e-08	3.7256e-07	1.5249e-08
2.25	4.3962e-08	2.0677e-09	2.1959e-07	1.0382e-08	3.0727e-07	1.4565e-08
2.5	3.5266e-08	1.8057e-09	1.7567e-07	9.0644e-09	2.4547e-07	1.2714e-08
2.75	2.7559e-08	1.4722e-09	1.3696e-07	7.3882e-09	1.9115e-07	1.0362e-08
3	2.1089e-08	1.1337e-09	1.0461e-07	5.6881e-09	1.4586e-07	7.9765e-09
3.25	1.5875e-08	8.2964e-10	7.8640e-08	4.1614e-09	1.0958e-07	5.8351e-09
3.5	1.1801e-08	5.7773e-10	5.8415e-08	2.8968e-09	8.1361e-08	4.0602e-09
3.75	8.6943e-09	3.8129e-10	4.3043e-08	1.9101e-09	5.9995e-08	2.6745e-09
4	6.3672e-09	2.3576e-10	3.1490e-08	1.2600e-09	4.3777e-08	1.8881e-09
4.25	4.6469e-09	1.3138e-10	2.2747e-08	5.0802e-10	3.1252e-08	4.8767e-10
4.5	3.9023e-09	5.0555e-11	3.3151e-08	4.1613e-11	5.6131e-08	2.7388e-10
4.75	2.4817e-09	6.8426e-09	1.2668e-08	3.4296e-08	1.8069e-08	4.8196e-08
5	0	0	0	0	0	0
L_∞	7.2607e-08	6.8426e-09	3.6721e-07	3.4296e-08	5.1702e-07	4.8196e-08

Table 2: L_{abs} and L_∞ errors for the solutions obtained by the Crank-Nicolson method for Example 1.

	$t = 0.1$		$t = 0.5$		$t = 0.7$	
x	$L_{abs}(u)$	$L_{abs}(v)$	$L_{abs}(u)$	$L_{abs}(v)$	$L_{abs}(u)$	$L_{abs}(v)$
-5	0	0	0	0	0	0
-4.75	1.4741e-09	6.8188e-09	7.5818e-09	3.4125e-08	1.0892e-08	4.7922e-08
-4.5	2.8897e-09	2.7271e-11	2.8067e-08	1.0521e-10	4.8998e-08	3.2573e-10
-4.25	3.2707e-09	1.0624e-10	1.5822e-08	3.5849e-10	2.1545e-08	2.6088e-10
-4	4.7889e-09	2.1107e-10	2.3541e-08	1.1454e-09	3.2599e-08	1.7342e-09
-3.75	6.9097e-09	3.5653e-10	3.4064e-08	1.7825e-09	4.7391e-08	2.4934e-09
-3.5	9.8215e-09	5.5554e-10	4.8431e-08	2.7810e-09	6.7326e-08	3.8945e-09
-3.25	1.3732e-08	8.1305e-10	6.7817e-08	4.0727e-09	9.4348e-08	5.7070e-09
-3	1.8845e-08	1.1268e-09	9.3251e-08	5.6467e-09	1.2986e-07	7.9139e-09
-2.75	2.5313e-08	1.4799e-09	1.2556e-07	7.4197e-09	1.7507e-07	1.0401e-08
-2.5	3.3162e-08	1.8337e-09	1.6495e-07	9.1980e-09	2.3032e-07	1.2897e-08
-2.25	4.2191e-08	2.1215e-09	2.1051e-07	1.0647e-08	2.9440e-07	1.4933e-08
-2	5.1853e-08	2.2480e-09	2.5959e-07	1.1289e-08	3.6365e-07	1.5839e-08
-1.75	6.1166e-08	2.1001e-09	3.0727e-07	1.0555e-08	4.3119e-07	1.4816e-08
-1.5	6.8692e-08	1.5740e-09	3.4626e-07	7.9218e-09	4.8674e-07	1.1129e-08
-1.25	7.2663e-08	6.1716e-10	3.6749e-07	3.1240e-09	5.1741e-07	4.4034e-09
-1	7.1306e-08	7.2511e-10	3.6169e-07	3.6136e-09	5.0999e-07	5.0481e-09
-0.75	6.3322e-08	2.2808e-09	3.2199e-07	1.1429e-08	4.5456e-07	1.6016e-08

-0.5	4.8415e-08	3.7652e-09	2.4664e-07	1.8891e-08	3.4850e-07	2.6494e-08
-0.25	2.7639e-08	4.8499e-09	1.4095e-07	2.4348e-08	1.9926e-07	3.4160e-08
0	3.3761e-09	5.2683e-09	1.7182e-08	2.6458e-08	2.4271e-08	3.7128e-08
0.25	2.1153e-08	4.9144e-09	1.0795e-07	2.4686e-08	1.5267e-07	3.4643e-08
0.5	4.2683e-08	3.8838e-09	2.1753e-07	1.9510e-08	3.0743e-07	2.7379e-08
0.75	5.8714e-08	2.4352e-09	2.9865e-07	1.2232e-08	4.2169e-07	1.7163e-08
1	6.8027e-08	8.9415e-10	3.4517e-07	4.4888e-09	4.8678e-07	6.2945e-09
1.25	7.0751e-08	4.5325e-10	3.5795e-07	2.2805e-09	5.0408e-07	3.2055e-09
1.5	6.8041e-08	1.4302e-09	3.4314e-07	7.1869e-09	4.8247e-07	1.0089e-08
1.75	6.1574e-08	1.9851e-09	3.0951e-07	9.9720e-09	4.3447e-07	1.3995e-08
2	5.3070e-08	2.1647e-09	2.6589e-07	1.0871e-08	3.7264e-07	1.5254e-08
2.25	4.3962e-08	2.0683e-09	2.1959e-07	1.0385e-08	3.0728e-07	1.4569e-08
2.5	3.5261e-08	1.8062e-09	1.7564e-07	9.0667e-09	2.4544e-07	1.2718e-08
2.75	2.7551e-08	1.4726e-09	1.3692e-07	7.3900e-09	1.9110e-07	1.0364e-08
3	2.1081e-08	1.1340e-09	1.0457e-07	5.6894e-09	1.4581e-07	7.9783e-09
3.25	1.5868e-08	8.2985e-10	7.8605e-08	4.1624e-09	1.0953e-07	5.8366e-09
3.5	1.1796e-08	5.7788e-10	5.8387e-08	2.8973e-09	8.1321e-08	4.0607e-09
3.75	8.6905e-09	3.8138e-10	4.3030e-08	1.9101e-09	5.9986e-08	2.6741e-09
4	6.3646e-09	2.3630e-10	3.1465e-08	1.2747e-09	4.3726e-08	1.9182e-09
4.25	4.6438e-09	1.3048e-10	2.2687e-08	4.8218e-10	3.1129e-08	4.3627e-10
4.5	4.0728e-09	4.6940e-11	3.4034e-08	6.1378e-11	5.7406e-08	3.0409e-10
4.75	2.4846e-09	6.8427e-09	1.2710e-08	3.4318e-08	1.8148e-08	4.8245e-08
5	0	0	0	0	0	0
L_∞	7.2663e-08	6.8427e-09	3.6749e-07	3.4318e-08	5.1741e-07	4.8245e-08

Table 3: L_{abs} and L_∞ errors for the solutions obtained by the exponential method for Example 1.

	$t = 0.1$		$t = 0.5$		$t = 0.7$	
x	$L_{abs}(u)$	$L_{abs}(v)$	$L_{abs}(u)$	$L_{abs}(v)$	$L_{abs}(u)$	$L_{abs}(v)$
-5	0	0	0	0	0	0
-4.75	1.4718e-09	6.8198e-09	7.5476e-09	3.4108e-08	1.0827e-08	4.7882e-08
-4.5	2.7196e-09	3.0238e-11	2.7189e-08	8.8930e-11	4.7733e-08	3.0072e-10
-4.25	3.2740e-09	1.0720e-10	1.5879e-08	3.8467e-10	2.1659e-08	3.1282e-10
-4	4.7920e-09	2.1059e-10	2.3568e-08	1.1312e-09	3.2653e-08	1.7049e-09
-3.75	6.9142e-09	3.5652e-10	3.4080e-08	1.7828e-09	4.7403e-08	2.4942e-09
-3.5	9.8278e-09	5.5549e-10	4.8463e-08	2.7809e-09	6.7371e-08	3.8948e-09
-3.25	1.3740e-08	8.1295e-10	6.7857e-08	4.0723e-09	9.4405e-08	5.7063e-09
-3	1.8854e-08	1.1266e-09	9.3297e-08	5.6460e-09	1.2993e-07	7.9129e-09
-2.75	2.5322e-08	1.4796e-09	1.2560e-07	7.4185e-09	1.7513e-07	1.0400e-08
-2.5	3.3169e-08	1.8333e-09	1.6498e-07	9.1963e-09	2.3037e-07	1.2895e-08
-2.25	4.2191e-08	2.1210e-09	2.1051e-07	1.0645e-08	2.9440e-07	1.4930e-08
-2	5.1843e-08	2.2474e-09	2.5954e-07	1.1286e-08	3.6358e-07	1.5834e-08
-1.75	6.1142e-08	2.0995e-09	3.0715e-07	1.0551e-08	4.3102e-07	1.4811e-08
-1.5	6.8649e-08	1.5734e-09	3.4605e-07	7.9185e-09	4.8644e-07	1.1124e-08
-1.25	7.2601e-08	6.1678e-10	3.6717e-07	3.1216e-09	5.1698e-07	4.3997e-09
-1	7.1229e-08	7.2513e-10	3.6130e-07	3.6140e-09	5.0945e-07	5.0487e-09
-0.75	6.3238e-08	2.2803e-09	3.2157e-07	1.1425e-08	4.5397e-07	1.6011e-08
-0.5	4.8329e-08	3.7646e-09	2.4620e-07	1.8896e-08	3.4789e-07	2.6507e-08
-0.25	2.7526e-08	4.8482e-09	1.4037e-07	2.4331e-08	1.9844e-07	3.4130e-08
0	3.4999e-09	5.2665e-09	1.7810e-08	2.6443e-08	2.5163e-08	3.7101e-08
0.25	2.1087e-08	4.9133e-09	1.0762e-07	2.4689e-08	1.5221e-07	3.4654e-08
0.5	4.2608e-08	3.8825e-09	2.1715e-07	1.9501e-08	3.0691e-07	2.7365e-08
0.75	5.8633e-08	2.4344e-09	2.9825e-07	1.2228e-08	4.2111e-07	1.7157e-08
1	6.7951e-08	8.9382e-10	3.4479e-07	4.4875e-09	4.8624e-07	6.2930e-09
1.25	7.0688e-08	4.5318e-10	3.5763e-07	2.2796e-09	5.0363e-07	3.2040e-09
1.5	6.7995e-08	1.4298e-09	3.4291e-07	7.1847e-09	4.8214e-07	1.0086e-08
1.75	6.1546e-08	1.9846e-09	3.0936e-07	9.9693e-09	4.3427e-07	1.3991e-08
2	5.3056e-08	2.1641e-09	2.6583e-07	1.0869e-08	3.7255e-07	1.5250e-08
2.25	4.3960e-08	2.0678e-09	2.1958e-07	1.0382e-08	3.0726e-07	1.4565e-08

2.5	3.5264e-08	1.8058e-09	1.7566e-07	9.0648e-09	2.4546e-07	1.2715e-08
2.75	2.7558e-08	1.4723e-09	1.3695e-07	7.3886e-09	1.9114e-07	1.0362e-08
3	2.1088e-08	1.1338e-09	1.0461e-07	5.6884e-09	1.4586e-07	7.9770e-09
3.25	1.5874e-08	8.2970e-10	7.8638e-08	4.1617e-09	1.0957e-07	5.8356e-09
3.5	1.1801e-08	5.7779e-10	5.8413e-08	2.8970e-09	8.1359e-08	4.0606e-09
3.75	8.6941e-09	3.8134e-10	4.3042e-08	1.9103e-09	5.9993e-08	2.6748e-09
4	6.3670e-09	2.3580e-10	3.1489e-08	1.2602e-09	4.3776e-08	1.8884e-09
4.25	4.6468e-09	1.3141e-10	2.2747e-08	5.0816e-10	3.1252e-08	4.8785e-10
4.5	3.9022e-09	5.0579e-11	3.3151e-08	4.1510e-11	5.6132e-08	2.7374e-10
4.75	2.4817e-09	6.8429e-09	1.2668e-08	3.4298e-08	1.8068e-08	4.8199e-08
5	0	0	0	0	0	0
L_∞	7.2601e-08	6.8429e-09	3.6717e-07	3.4298e-08	5.1698e-07	4.8199e-08

Table 4: L_∞ errors for the solutions obtained by the explicit method for Example 2.

$t = 0.1$		$t = 0.5$		$t = 1.2$		
$L_\infty(u)$	$L_\infty(v)$	$L_\infty(u)$	$L_\infty(v)$	$L_\infty(u)$	$L_\infty(v)$	γ
1.3878e-17	2.1706e-18	7.2858e-17	6.7389e-17	1.8492e-15	1.6935e-15	10
1.3878e-17	3.4179e-18	6.9389e-17	6.8140e-17	1.5057e-15	1.4968e-15	9
1.3878e-17	2.2486e-17	8.3267e-17	1.0382e-16	1.6410e-15	1.4830e-15	8
1.0408e-17	1.6715e-16	3.5388e-16	7.0985e-16	6.8522e-15	4.2776e-15	7
6.5919e-17	1.2440e-15	2.8588e-15	5.3674e-15	5.4887e-14	3.3170e-14	6
4.7531e-16	9.1844e-15	2.1042e-14	3.9559e-14	4.0384e-13	2.4374e-13	5
3.4694e-15	6.7665e-14	1.5495e-13	2.9140e-13	2.9743e-12	1.7956e-12	4
2.5029e-14	4.8959e-13	1.1210e-12	2.1083e-12	2.1521e-11	1.2993e-11	3
1.5663e-13	3.0906e-12	7.0689e-12	1.3306e-11	1.3583e-10	8.2013e-11	2
4.9616e-13	6.8765e-12	1.6557e-11	3.0008e-11	3.0894e-10	1.8050e-10	1
1.4135e-12	1.9934e-11	4.7849e-11	8.6907e-11	8.9387e-10	5.2682e-10	0

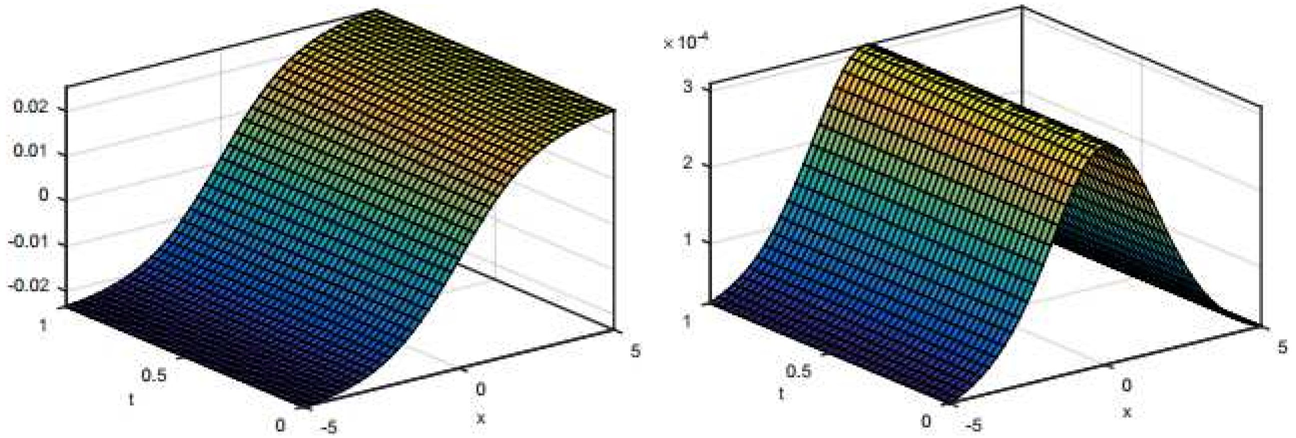
Table 5: L_∞ errors for the solutions obtained by the Crank-Nicolson method for Example 2.

$t = 0.1$		$t = 0.5$		$t = 1.2$		
$L_\infty(u)$	$L_\infty(v)$	$L_\infty(u)$	$L_\infty(v)$	$L_\infty(u)$	$L_\infty(v)$	γ
2.4286e-17	4.5297e-18	1.9776e-16	1.3522e-16	4.3507e-15	3.4418e-15	10
3.4694e-17	4.4291e-18	2.3939e-16	1.4407e-16	5.8530e-15	4.3006e-15	9
2.4286e-17	2.2566e-17	2.2551e-16	1.6442e-16	5.8183e-15	5.5751e-15	8
2.0817e-17	1.6606e-16	3.4001e-16	7.2287e-16	7.1297e-15	4.1290e-15	7
8.3267e-17	1.2223e-15	3.2231e-15	5.4549e-15	7.0319e-14	4.4692e-14	6
6.2103e-16	9.0342e-15	2.3630e-14	4.0292e-14	5.1723e-13	3.2886e-13	5
4.5970e-15	6.6570e-14	1.7428e-13	2.9695e-13	3.8130e-12	2.4202e-12	4
3.3282e-14	4.8167e-13	1.2614e-12	2.1487e-12	2.7598e-11	1.7520e-11	3
2.0898e-13	3.0404e-12	7.9496e-12	1.3558e-11	1.7416e-10	1.1059e-10	2
6.8839e-13	6.7795e-12	1.8993e-11	3.0849e-11	3.9889e-10	2.4270e-10	1
1.8152e-12	1.9633e-11	5.4022e-11	8.8875e-11	1.1491e-09	7.1019e-10	0

Table 6: L_∞ errors for the solutions obtained by the exponential method for Example 2.

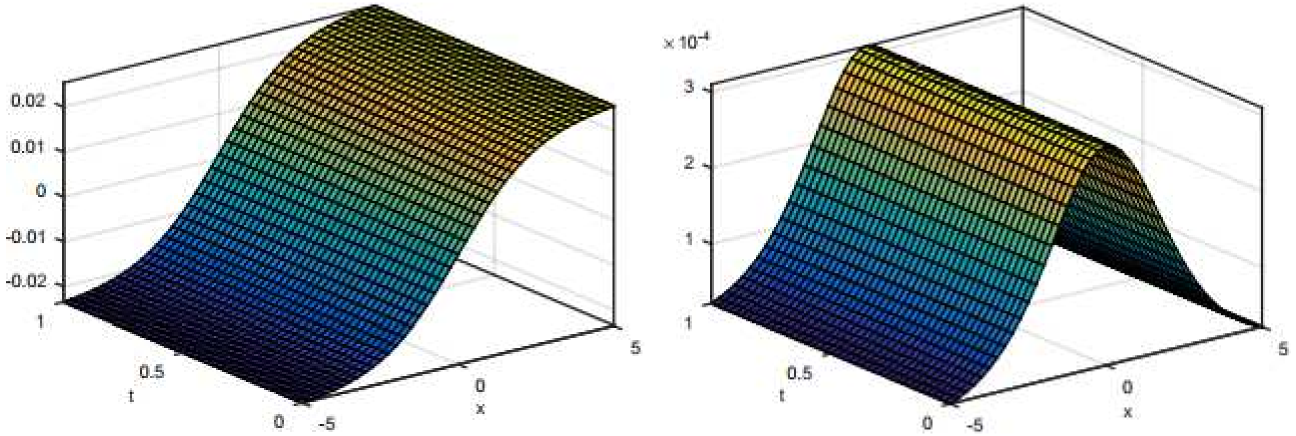
$t = 0.1$		$t = 0.5$		$t = 1.2$		
$L_\infty(u)$	$L_\infty(v)$	$L_\infty(u)$	$L_\infty(v)$	$L_\infty(u)$	$L_\infty(v)$	γ
1.0408e-17	1.9720e-18	4.8572e-17	3.3490e-17	9.9226e-16	7.5967e-16	10
1.0408e-17	3.7490e-18	5.8981e-17	3.4792e-17	1.0408e-15	8.9451e-16	9
1.3878e-17	2.2487e-17	7.2858e-17	9.1437e-17	1.4259e-15	1.0653e-15	8
1.0408e-17	1.6827e-16	3.8858e-16	7.2889e-16	7.7889e-15	5.1192e-15	7
6.2450e-17	1.2434e-15	2.8449e-15	5.3542e-15	5.4585e-14	3.3003e-14	6
4.6838e-16	9.1836e-15	2.1028e-14	3.9541e-14	4.0362e-13	2.4379e-13	5
3.4660e-15	6.7666e-14	1.5494e-13	2.9140e-13	2.9742e-12	1.7953e-12	4

2.5032e-14	4.8960e-13	1.1210e-12	2.1084e-12	2.1521e-11	1.2993e-11	3
1.5688e-13	3.0907e-12	7.0704e-12	1.3307e-11	1.3584e-10	8.2015e-11	2
4.9141e-13	6.8762e-12	1.6531e-11	2.9995e-11	3.0882e-10	1.8050e-10	1
1.4645e-12	1.9939e-11	4.8133e-11	8.7048e-11	8.9528e-10	5.2670e-10	0



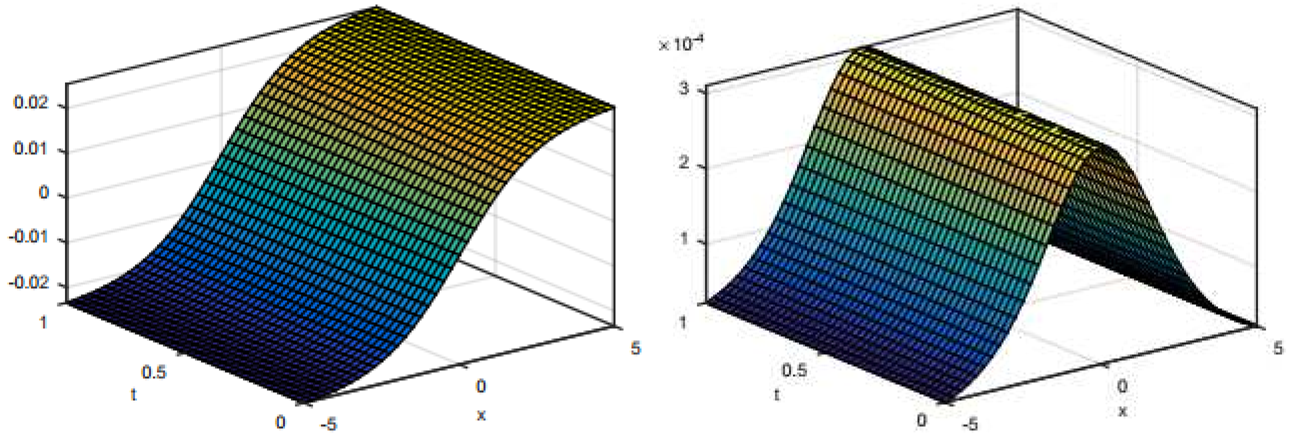
(a) $u(x,t)$ (b) $v(x,t)$

Figure 1: Exact solutions of WBK model for Example 1.



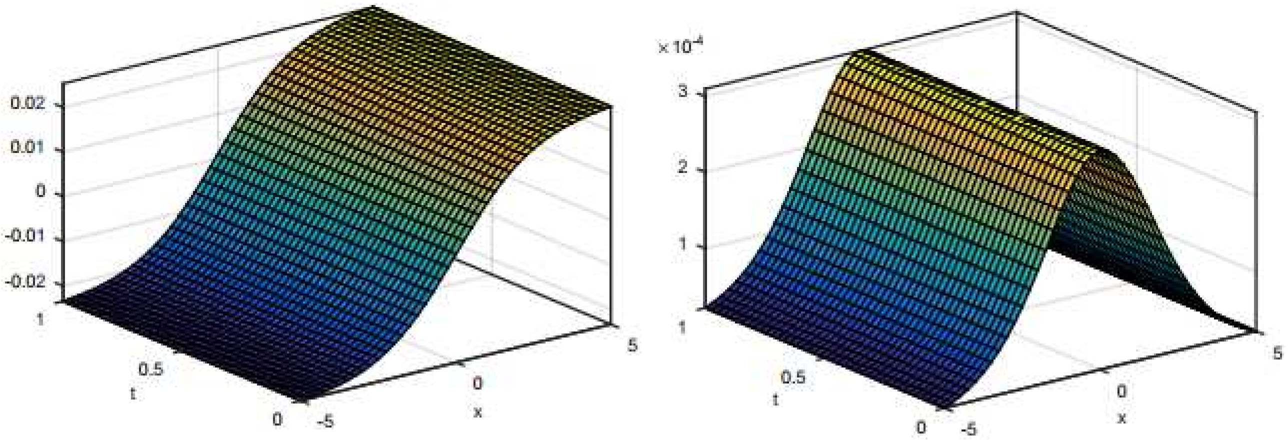
(a) $U(x,t)$ (b) $V(x,t)$

Figure 2: Numerical solutions obtained by the explicit method for Example 1.



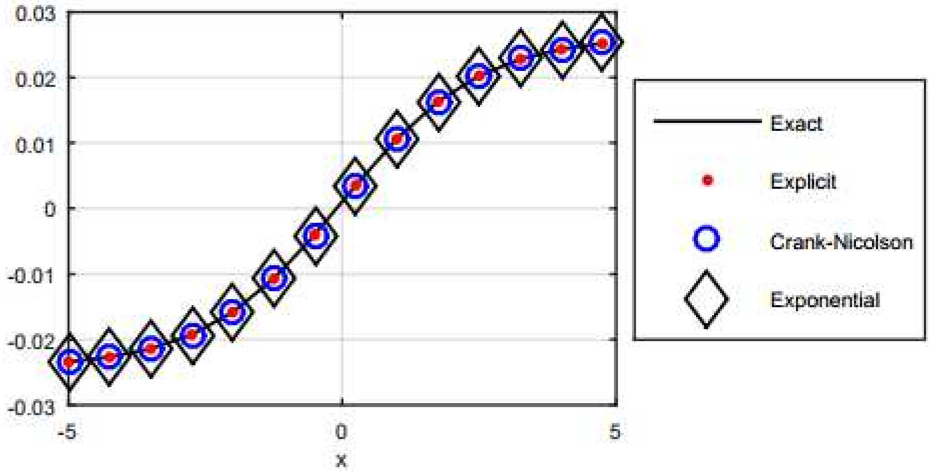
(a) $U(x,t)$ (b) $V(x,t)$

Figure 3: Numerical solutions obtained by the Crank-Nicolson method for Example 1.

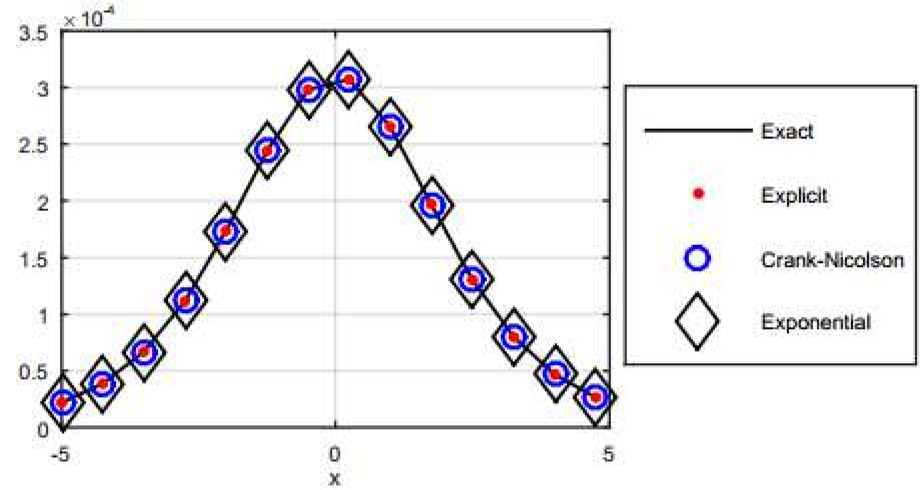


(a) $U(x,t)$ (b) $V(x,t)$

Figure 4: Numerical solutions obtained by the exponential method for Example 1.



(a) $U(x,0.5)$



(b) $V(x,0.5)$

Figure 5: Comparison of the exact solutions with the numerical solutions when $t = 0.5$ with some values of x in Example 1.

Conclusions

We have solved (1) numerically by using explicit, Crank-Nicolson and exponential finite difference methods. Two test problems demonstrated the efficiency and accuracy of the proposed methods. The numerical results show that these methods are a powerful and efficient technique for finding the approximate solutions of the model, and the results are in excellent agreement with the exact solutions. It is seen from the tables and figures that all methods are close to each other. In particular, these methods have optimal results for some values of given parameters.

Acknowledgement

I would like to acknowledge my supervisor Professor **Dr. Rostam K. Saeed**, for his assistance, many helpful comments, and suggestions that improved this paper and my M.Sc. thesis.

References

[1] Ablowitz, M. J. and Clarkson, P. A., *"Solitons, Nonlinear Evolution Equations and Inverse Scattering".* Cambridge University Press, (1991).

[2] Bahadir, A. R., *"Exponential finite-difference method applied to Kortewge-de Vries equation for small times".* Applied Mathematics and Computation, Vol. 160, No. 3, pp. 675- 682. (2005).

[3] Bhattacharya, M. C., "*A new improved finite difference equation for heat transfer during transient change".* Applied Mathematical Modelling, Vol. 10, No.1, pp. 68-70. (1986).

[4] Duffy, D. J.,*" Finite Difference Methods in Financial Engineering: A Partial Differential Equation Approach".* John Wiley & Sons, (2006).

[5] Engui, F. and Hongqing, Z., "*Backlund transformation and exact solutions for Whitham-Broer-kaup equations in shallow water".* Applied Mathematics and Mechanics, Vol. 19, No. 8, pp.713-716. (1998).

[6] Handschuh, R. F., *"An Exponential Finite Difference Technique for Solving Partial Differential Equations*". Master of Science Thesis, University of Toledo, (1987).

[7] Kupershmidt, B. A., *"Mathematics of dispersive water waves".* Communications in Mathematical Physics, Vol. 99, No. 1, pp.51-73. (1985).

[8] Lapidus, L. and Pinder, G. F.,"*Numerical Solution of Partial Differential Equations in Science and Engineering".* John Wiley & Sons, (1999).

[9] Logan, J. D., *"Applied Mathematics: A Contemporary Approach".* John Wiley & Sons, (1987).

[10] Smith, G. D., *"Numerical Solution of Partial Differential Equations: Finite Difference Methods".* Oxford University Press, (1985).

[11] Mishra, L. N., "*On Existence and Behavior of Solutions to Some Nonlinear Integral Eequations with Applications*". Ph.D. Thesis, National Institute of Technology, Silchar 788 010, Assam, India, (2016),

[12] Mishra, L. N.; Agarwal, R. P. and Sen, M., *"Solvability and asymptotic behavior for some nonlinear quadratic integral equation involving Erdèlyi-Kober fractional integrals on the unbounded interval*". Progress in Fractional Differentiation and Applications, Vol. 2, No. 3, pp.153-168. (2016).

[13] Mishra, L. N., Sen, M., "*On the concept of existence and local attractivity of solutions for some quadratic Volterra integral equation of fractional order*". Applied Mathematics and Computation, Vol. 285, pp.174-183. (2016).

[14] Mishra, L. N.; Sen, M. and Mohapatra, R. N., "*On existence theorems for some generalized nonlinear functional-integral equations with applications*". Filomat, accepted on March 21, 2016, in press.

[15] Mishra, L. N.; Srivastava, H. M. and Sen, M., "*On existence results for some nonlinear functional-integral equations in Banach algebra with applications*". Int. J. Anal. Appl., Vol. 11, No. 1, pp.1-10. (2016).

[16] H. S. Shukla, H. S.; Tamsir, M.; Srivastava, V. K. and Rashidi, M. M., *"Modified cubic B-spline differential quadrature method for numerical solution of three-dimensional coupled viscous Burger equation*", Modern Physics Letters B, Vol. 30, No. 11, pp. 1-17. (2016).

[17] Suarez, P. U. and Morales J. H., "*Numerical solutions of two-way propagation of nonlinear dispersive waves using radial basis functions"*. International Journal of Partial Differential Equations, Vol. (2014), pp.1-8. (2014).

[18] Thakumar, M. S., "*Computer Based Numerical Analysis*". Khanna Publishers, (1989).

[19] Wang, L.; Gao, Y. T.; Gai, X. L.; Yu, X.; and Sun, Z. Y., *"Vadermonde-type odd-soliton solutions for the Whitham–Broer–Kaup model in the shallow water small amplitude regime"*. Journal of Nonlinear Mathematical Physics, Vol. 17, *No.* 2, pp.197-211, (2010).

[20] Wani, S. S. and Thakar, S.H., *"Crank-Nicolson type method for Burgers equation*". International Journal of Applied Physics and Mathematics, Vol.3, No. 5, pp.324-328. (2013).

[21] Xu, T.; Li, J.; Zhang, H-Q.; Zhang, Y-X.; Yao, Z-Z. and Tian, B., New extension of the tanh-function method and application to the Whitham-Broer-Kaup shallow water model with symbolic computation. *Physics Letters A*, Vol. 369, No. 5, pp.458-463. (2007).