DOCTOR OF PHILOSOPHY

Finite element studies of the Korteweg-de Vries equation.

Ali, Ahmed Hassan Ahmed

Award date:
1989

Awarding institution:
Bangor University

FINITE ELEMENT STUDIES OF THE KORTEWEG-DE VRIES EQUATION

AHMED HASSAN AHMED ALI

School of Mathematics
University College of North Wales, Bangor

Thesis presented for the degree of
Philosophiae Doctor

in the
University of Wales

May, 1989

IN THE NAME OF

ALLAH

MOST GRACIOUS MOST MERCIFUL

THANKING

HIM

WITH A FULL HEART AND DEVOTED TONGUE

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor Dr. L.R.T Gardner for suggesting the topic of research and his help, enthusiastic interest, and encouragement throughout the period of this work and guidance during the preparation of this thesis. I am also deeply grateful to Dr. G. Gardner for her continual helpful advice, discussions and valuable time.

I would also like to extend my deepest gratitude to the Government of the Kingdom of Saudi Arabia. This research would not have been possible if not for the King Faisal Scholarship which the Government awarded me.

I am particularly grateful to my parents, brothers and sister. What I owe to my wife is not possible, within these formal limits, either to describe or to acknowledge.

I DEDICATE THIS WORK TO MY BELOVED CHILDREN

SUMMARY

The main aim of this study is the construction of new, efficient, and accurate numerical algorithms based on the finite element method, for the solution of the Korteweg-de Vries equation.

Firstly the theoretical background to the KdV equation is discussed, and existing numerical methods based mainly on finite differences are discussed.

In the following chapters finite element methods based on Bubnov-Galerkin approach are set up. Initially we used cubic Hermite interpolation functions, and in later methods cubic spline and quadratic spline shape functions. The appropriate element matrices were determined algebraically using the computer algebra package REDUCE. Finally we set up a method based on collocation using quintic spline interpolation functions.

The numerical algorithms have been validated by studying the motion, interaction and development of solitons. We have demonstrated that these algorithms can faithfully represent the amplitude of a single soliton over many time steps and predict the progress of the wave front with small error. In the interaction of two solitons the numerical algorithms faithfully reproduce the changes in amplitudes and phase shifts of the analytic solution.

We compare, in detail the L_2- and L_∞-error norms of the present algorithms with published results. The conservative properties of the algorithms are also examined in detail.

The modified and generalised Korteweg-de Vries equation have

also been solved using collocation method with quintic splines interpolation functions. Again, the solution method has been validated by studying the motion, interaction, and development of solitons.

We have concluded that all the new methods set up here are capable of reproducing the solutions to the **KdV** equation efficiently and accurately, the best amongst these methods are collocation with quintic splines or Galerkin with quadratic splines. The collocation method is also very efficient and accurate for solving the modified **KdV** equation.

CONTENTS

CHAPTER 1

INTRODUCTION

The first recorded observation of the solitary wave was made in 1834 by Scott Russell [1] when he saw a rounded, smooth, well defined heap of water detached itself from the prow of a stopped barge and proceeded without change of shape or diminution of speed over two miles along a channel [2]. The words "solitary wave" were coined by Scott Russell himself, mainly because this type of wave motion stands apart from the other types of oscillatory wave motion. There was subsequently a gap of more than sixty years between Scott Russell's observation of the shallow water solitary wave and any theoretical treatment of the phenomenon. Despite some attempts by Scott Russell to guess at the analytical formula for the wave profile, his observation went unexplained in his own life time. In the following decades after him, the solitary wave of translation was briefly mentioned by various mathematicians including Stokes [3] in 1847 and Boussinesq [4] in 1872. However, initial theoretical confirmation of Scott Russell's work had to wait until 1895 when Korteweg and de Vries [5] derived their now famous equation for the propagation of waves in one direction on the surface of a shallow canal. If the canal has normal depth ℓ and $\ell + \eta$ (η being small) represents the elevation of the surface above the bottom, the partial differential equation which governs the wave motion is:

$$\eta_t = \frac{3}{2}\sqrt{\frac{g}{\ell}}\,\frac{\partial}{\partial x}\left[\frac{2}{3}\alpha\,\eta + \frac{1}{2}\,\eta^2 + \frac{1}{3}\,\sigma\,\frac{\partial^2\eta}{\partial x^2}\right] \qquad (1.1)$$

where; α is a small constant related to the uniform motion of the liquid, $\sigma = \ell^3/3 - T\ell/\rho g$ is a parameter, T the surface tension and

ρ the density of the fluid. Appropriate scalings will transform it into a more manageable form. If we define:

$$\eta = \beta \alpha u \quad , \quad \xi = \sqrt{\frac{2\alpha\mu}{\sigma}}\, x \quad , \quad \tau = \sqrt{\frac{2g\mu\alpha}{\sigma\ell}}^{3} t$$

then the equation (1.1) can be written in the form of the Korteweg-de Vries (**KdV**) equation:

$$u_\tau + u_\xi + \varepsilon u u_\xi + \mu u_{\xi\xi\xi} = 0 \qquad (1.2)$$

where; $\varepsilon = \frac{3}{2}\beta$ and μ are given parameters. Using the variable transformation $x = \xi - \tau$ and writing t instead of τ, equation (1.2) becomes:

$$u_t + \varepsilon u u_x + \mu u_{xxx} = 0 \qquad (1.3)$$

A generalisation of the **KdV** equation is:

$$u_t + \varepsilon u^p u_x + \mu u_{xxx} = 0 \qquad (1.4)$$

where $p = 1, 2, \ldots$

The most simple generalisation of the **KdV** equation (1.4) is the modified Korteweg-de Vries (m**KdV**) equation:

$$u_t + \varepsilon u^2 u_x + \mu u_{xxx} = 0 \qquad (1.5)$$

This equation has been used to describe acoustic waves in certain anharmonic lattices [6] and Alfén waves in a collisionless plasma [7,8].

In spite of this early derivation of the Korteweg-de Vries (**KdV**) equation, it was not until 1960 that a new application of the equation was found in a study of collision free hydromagnetic waves by Gardner and Morikawa [9]. This is surprising because, in general, the **KdV** equation describes the unidirectional propagation of small but finite amplitude waves in a nonlinear dispersive medium. Gardner and Morikawa rederived the **KdV** equation and also proved that it was the limiting equation describing long wave

propagation perpendicular to a uniform magnetic field in a cold lossless (collisionless) plasma [10]. Since 1963, many researchers, e.g. Su and Gardner [11] have derived the **KdV** equation as the relevant long wave asymptotic description of a more complete set of model equations. Kruskal [12] and Zabusky [6,13,14] showed that the **KdV** equation governs longitudinal waves propagating in a one dimensional lattice of equal masses coupled by nonlinear springs, the Fermi Pasta Ulam problem. Other applications in plasma physics were given by Berezin and Karpman [15] and by Washimi and Taniuti [16] in their study of ion acoustic waves in a cold plasma. Wijngaaden [17] found that it described pressure waves in a liquid gas bubble mixture. Naraboli [18] proved it governed waves in elastic rods. Shen [19] derived the **KdV** equation in the study of three dimensional water waves and Leibovich [20] proved it described the axial component of velocity in a rotating fluid flow down a tube, and thermally excited phonon packets in low temperature nonlinear crystals [21].

The current interest in the **KdV** equation stems from the fact that it can be solved analytically by the inverse scattering method, but numerical methods for this pure initial value problem remain important. This is because the inverse scattering technique still requires the solution of the time independent Schrodinger equation, with a potential determined by the initial condition. Since the Schrodinger equation can only be solved analytically for a few special types of potential, the inverse scattering technique can thus not be used to obtain an explicit analytical solution of the **KdV** equation for arbitrary initial data [22]. The theoretical aspects of the solution of the **KdV** equation have attracted attention. In particular, the problem of existence and uniqueness of solutions for certain classes of initial conditions have been

studied by many authors including Lax [23], Sjoberg [24] and Gardner [25]. These authors have examined the existence of solitary wave or soliton solutions.

The physical models described by the **KdV** equation represent situations requiring large scale time calculations. Consequently, any numerical method proposed for determining the solution of the mathematical equation must possess at least two properties [26]:
(1) The method must represent faithfully amplitudes of the solution for many time steps in the calculation; it must be conservative, and
(2) The method must be capable of predicting such wave fronts with minimal error. Hence, the phase error of the method must be small.

The **KdV** equation was solved numerically first by Zabusky and Kruskal [27] using a finite difference method. In that study they discovered the properties of the interaction of two solitary waves. Zabusky and Kruskal [27] defined the concept of a soliton as a localised (solitary) wave that propagates at a uniform speed and preserves its shape and speed when it interacts with a second solitary wave but does suffer a phase shift. Greig and Morris [26] proposed a Hopscotch finite difference method and compared it with the original Zabusky and Kruskal [27] leap frog scheme and found that it gave better results [26].

The application of spectral, Pseudospectral and Fourier transform or series expansion methods to the **KdV** equation has been studied by Schamel [28], Abe and Inoue [29], Gazdag [30], and Canosa and Gazdag [31]. Fornberg and Whitham [32] have discussed the numerical solution of the **KdV** equation (1.3) using a pseudospectal method in the x variable together with a leap frog method in t. They have also studied the higher order generalised **KdV** equation (1.4) and numerical results show that with $p > 2$ the

soliton collision is inelastic. Numerical calculations for the generalised **KdV** equation (1.4) show that the solitons become taller and narrower during an interaction, unlike those of the **KdV** equation (1.3) which become wider and smaller during an interaction [33].

Finite element methods have also been used. The first of those proposed was due to Wahlbin [34], who suggested a dissipative Galerkin method in which the same trial and test functions are used. The basis functions are smoothed splines constructed from piecewise polynomials of order three or higher, and the elements are of equal length h. Numerical computations for this method were carried out by Alexander and Morris [35], who used cubic splines and a range of dissipation coefficients from zero to one. They studied the motion of a single soliton and a double soliton taking the initial condition in each case from the theoretical solution. The results particularly in the second problem were not good [33]. Sanz-Serna and Christie [36] proposed a modified Petrov-Galerkin method with piecewise linear trial and cubic spline test functions. They compared their method to some of those mentioned above, and showed that finite element methods for the **KdV** equation are well worth considering. Further schemes using Petrov-Galerkin methods have been given by Schoombie [22] which can be either dissipative or nondissipative, and which contain the Sanz-Serna and Christie method as a special case. The trial functions were chosen to be linear and the test functions to be B-splines of various orders. Higher accuracy is obtained by shifting the support of the test functions. This differs from the Wahlbin approach, where both trial and test functions have to be at least cubic splines, and where the test functions are modified rather than shifted to introduce dissipation into the numerical

method. The important advantage of the shift test functions is that piecewise linears can be used as both trial and test functions, in spite of the third derivative in the **KdV** equation (1.3), which would normally require at least the continuity of the first derivative for either the trial or the test functions. This method has the disadvantage that it is less accurate than the modified Petrov-Galerkin method (Sanz-Serna and Christie). However, it involves much less computational effort.

A different approach to the numerical study of the **KdV** equation has been adopted by Osborne and Provenzale [37]. In this study, direct use is made of the inverse scattering transformed to solve the initial value problem, with initial data being approximated by a piecewise constant function. This generalisation of the usual spectral methods promises to be a useful technique, with the only drawback being the fact that it is not applicable to equations for which no inverse scattering problem is known.

The Korteweg-de Vries equation is an important nonlinear partial differential equation which arises in the study of many different physical systems for which analytic solutions have only been found for a very restricted set of initial conditions. Thus numerical methods are very necessary to effect solutions for a wide range of initial conditions. In this thesis attempts are made to produce numerical methods based on the finite element method which are superior to those already being used. We expect these methods to have two advantages:

Firstly, the computed L_2- and L_∞-error norms might be smaller in comparison with those of earlier authors, so that the numerical solution will be more accurate.

Secondly, the computed values for the first three or four conservative quantities of the **KdV** equation should change as

little as possible during the computer run. Essentially the method should be conservative.

In chapter 2, we describe how the Korteweg-de Vries equation is set up for ion acoustic wave, and also how we derive the analytic solution for the **KdV** equation under restricted initial conditions. We give a review of the interaction of two solitary waves and also of the conservation laws.

Early attempts at numerical solutions of the **KdV** equation are outlined and discussed in chapter 3. These earlier methods were mainly of four types; the finite difference methods, both explicit and implicit, transform methods such as the pseudospectral or fourier transform, splitting method, fourier expansion methods and finite element methods such as a dissipative Galerkin method.

In chapter 4, we show how a finite element using the Galerkin method with trial and test functions as cubic Hermite polynomials can be set up. The element matrices are determined algebraically using REDUCE [38]. Assembling the element matrices together and using a Crank-Nicolson difference scheme for the time derivative leads to a 7-banded system of nonlinear algebraic equations which is solved by a septa-diagonal algorithm. The method is tested by calculating how the L_2- and L_∞-error norms vary during the motion of a single and double soliton and comparing this with the error obtained by earlier authors for a similar experiment. The first three conservative quantities are also computed for simulations using a single soliton, double soliton, and Gaussian initial condition.

In chapter 5, we set up a finite element using the Bubnov-Galerkin method in which trial and test functions are cubic spline polynomials. The element matrices are obtained analytically

using REDUCE. Assembling element matrices together and using a Crank-Nicolson difference scheme for the time derivative leads to a system of nonlinear algebraic equations which is solved using a septa-diagonal algorithm. A linear stability analysis is used to show that the scheme is unconditionally stable. Classical test problems, including collisionless shocks and soliton development and interaction, are used to prove the method. The L_2- and L_∞-error norms have been computed for single and double soliton. The first four conservative quantities have been computed. Two schemes have been discussed; one using integration by parts and the other without.

In chapter 6, a finite element method based on the Bubnov-Galerkin method in which the trial and test functions are quadratic spline polynomials is set up. The element matrices are determined algebraically using REDUCE. Assembling together the element matrices and again using a Crank-Nicolson difference scheme for the time derivative lead to a system of nonlinear algebraic equations which can be solved by a penta-diagonal algorithm. A linear stability analysis is used to show that the scheme is unconditionally stable. Classical problems concerning the development and interaction of solitons are used to test the method. The breakdown of a Gaussian initial condition into a train of solitons is considered. The L_2- and L_∞-error norms have been determined for single and double soliton. Also, the first three conservative quantities have been calculated.

In chapter 7, a finite element method based on collocation with quintic spline interpolation polynomials over the finite elements is set up. This leads to a nonlinear algebraic system with 5-banded matrices which can be solved using a penta-diagonal algorithm. A linear stability analysis is set up which shows that

the scheme is unconditionally stable. Classical initial conditions, which model soliton interaction and undular bores in shallow water, are used to evaluate the L_2- and L_∞-error norms for single and double soliton and the first fourth conservative quantities. The breakdown of a Gaussian initial condition into a train of solitons is observed.

In chapter 8, a numerical method to solve the generalised KdV equation (1.4) with $p = 1,2,3$ based on the collocation method with quintic spline interpolation polynomials over the finite elements is presented. The recurrence relationship obtained leads to a nonlinear algebraic system of 5-banded matrices which can be solved using a penta-diagonal algorithm. A linear stability analysis is investigated. Classical test problems, including collisionless shocks and soliton development, motion and interaction, are used to compute the first four (three if $p \geq 3$) conservative quantities. The L_2- and L_∞-error norms for a single and double (for $p = 1,2$) soliton solution are used to give an indication that as p (the power of u in the nonlinear term $u^p u_x$) increases, the error increases. The breakdown of a Gaussian initial condition into a train of solitons is demonstrated.

CHAPTER 2

PHYSICAL REVIEW FOR THE KORTEWEG-DE VRIES EQUATION

2.1 Introduction:

At the present time nonlinear wave phenomena are the subject of intense study in many branches of applied mathematics, physics, and engineering, e.g. in optics, plasma physics, radio physics, acoustics, hydrodynamics ...etc [39].

One of the most important nonlinear wave equations is the Korteweg-de Vries equation (**KdV**) which was originally derived in 1895 by Korteweg and de Vries [5] in order to describe the behaviour of one-dimensional shallow water waves with small but finite amplitude. More recently the **KdV** equation has also been found to describe various other kinds of phenomena such as acoustic waves in anharmonic crystals, waves in bubble-liquid mixtures, magnetohydrodynamics, waves in warm plasmas, and ion-acoustic waves.

2.2 Some Applications in which the **KdV** Equation Arises:

In this section, we will study some of the cases where the **KdV** equation arises as a realistic model governing the evolution of waves in media in which weak nonlinear effects are considered. We quote four examples: The first occurs in plasma physics where the **KdV** equation governs the evolution of long compressive waves in a plasma of cold ions and hot electrons; the second is the shallow water waves problem; the third case occurs in meteorology in studies of the propagation of nonlinear Rossby waves in a homogeneous rotating fluid. The latter case is slightly different than the earlier two in that a second space dimension (y) occurs in the original equations and the coefficients of the final **KdV**

equation are found to be integrals over y. The fourth example is from electric circuit theory in which a nonlinear capacitance is used. In this case a generalised pth order **KdV** equation with the nonlinearity depending on the capacitance is obtained. We have chosen this example to illustrate how the modified **KdV** equation can arise in certain circumstances. It is well known that the **KdV** equation is quite simple in structure as it is a single scalar equation involving one dependent variable and two independent variables. However, the original equations of motion of most physical systems are not simple and, generally, they contain several dependent variables, so that we need a procedure which reduces such sets of equations to simpler forms (perturbation procedure). To apply this method, we try scaling all variables in the problem to dimensionless form and we expand all the dependent variables in terms of a perturbation parameter ε [40]. To illustrate this approach we show in the next section how ion-acoustic waves are governed by the **KdV** equation

2.2.1 Ion-Acoustic Waves [40,41,42]:

Consider a one dimensional sea of electrons and ions, each of mass m_e, m_i and charge -e, +e with density n_e, n_i per unit volume respectively. This is technically known as a plasma of electrons and ions, as in [40]. Since the electron mass is much lighter than any ion mass, the electron inertia can be neglected but the electrostatic effect of the electron charge cannot be neglected. For this, the usual method is to treat the electrons as a gas [43]. An electron gas can be thought of as a gas problem and in an idealized situation may be described by the equation of the state:

$$P = K_B T_e n_e \qquad (2.2.1.1)$$

where K_B is known as Boltzmann's constant, P is the pressure and

T_e is the value of the electron temperature which gives a measure of how energetic (hot) the electrons in the gas are. Here we will take the ion temperature T_i as $T_i << T_e$. For the electron gas, the electrostatic force is related to the pressure gradient by an equation:

$$e\, n_e \frac{\partial \phi}{\partial x} = K_B\, T_e \frac{\partial n_e}{\partial x} \qquad (2.2.1.2)$$

where ϕ is the electrostatic potential.

Integrate this equation to obtain:

$$n_e = n_0 \exp\left(\frac{e}{K_B\, T_e}\, \phi\right) \qquad (2.2.1.3)$$

where n_0 is the equilibrium background density.

For the ions, the equations of conservation of mass and momentum are:

$$\frac{\partial n_i}{\partial t} + \frac{\partial}{\partial x}\,(n_i v_i) = 0 \qquad (2.2.1.4)$$

$$m_i \left(\frac{D}{Dt}\, v_i\right) = -e \frac{\partial \phi}{\partial x} \qquad (2.2.1.5)$$

where the total derivative is given by:

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + v_i \frac{\partial}{\partial x} \qquad (2.2.1.6)$$

For the electrostatic potential ϕ, Poisson's equation is:

$$\frac{\partial^2 \phi}{\partial x^2} = 4\, \pi\, e(n_e - n_i) \qquad (2.2.1.7)$$

Equations (2.2.1.3) and (2.2.1.7) indicate that ϕ and n_i can be rescaled as:

$$\Phi = e\, \phi/(K_B\, T_e) \quad ; \quad n = n_i/n_0 \qquad (2.2.1.8)$$

Substituting (2.2.1.3) into (2.2.1.7) and using (2.2.1.8), gives:

$$\left(\frac{K_B\, T_e}{4\pi n_0 e^2}\right) \Phi_{xx} = \exp(\Phi) - n \qquad (2.2.1.9)$$

Then, a new dimensionless x variable can be introduced as:

$$\bar{x} = \frac{x}{\lambda} \qquad ; \qquad \lambda = \left(\frac{K_B T_e}{4\pi n_0 e^2} \right)^{1/2} \tag{2.2.1.10}$$

where λ is known as the Debye length of the plasma. Using the expressions given above for Φ, n, and $\bar{x}$ one can define the non-dimensional variables:

$$\bar{t} = w_p t \qquad ; \qquad v = \frac{v_i}{\lambda w_p} \qquad ; \qquad w_p = \left(\frac{4\pi e^2 n_0}{m_i} \right)^{1/2} \tag{2.2.1.11}$$

where w_p is the plasma frequency, and λw_p is the ion sound speed. Using non-dimensional variables the equations (2.2.1.4), (2.2.1.5) and (2.2.1.7) can be written in the form:

$$\begin{aligned} n_{\bar{t}} + (nv)_{\bar{x}} &= 0 \\ v_{\bar{t}} + vv_{\bar{x}} &= -\Phi_{\bar{x}} \\ \Phi_{\bar{x}\bar{x}} &= \exp(\Phi) - n \end{aligned} \tag{2.2.1.12}$$

The boundary conditions are taken to be $n \longrightarrow 1$; $v, \Phi \longrightarrow 0$ as $|x| \longrightarrow \infty$, so that asymptotic expansions for n, Φ and v are:

$$\begin{aligned} n &= 1 + \varepsilon n^{(1)} + \varepsilon^2 n^{(2)} + \ldots & n^{(j)} &\longrightarrow 0 \text{ as } |x| \longrightarrow \infty \\ \Phi &= \varepsilon \Phi^{(1)} + \varepsilon^2 \Phi^{(2)} + \ldots & \Phi^{(j)} &\longrightarrow 0 \text{ as } |x| \longrightarrow \infty \\ v &= \varepsilon v^{(1)} + \varepsilon^2 v^{(2)} + \ldots & v^{(j)} &\longrightarrow 0 \text{ as } |x| \longrightarrow \infty \end{aligned}$$

$$j = 1, 2, \ldots \tag{2.2.1.13}$$

where ε is small parameter, the superscript denotes to the order of the perturbation.

Using (2.2.1.13) to linearise (2.2.1.12) and eliminating $n^{(1)}$ and $v^{(1)}$, we have:

$$\Phi^{(1)}_{\bar{x}\bar{x}\bar{t}\bar{t}} + \Phi^{(1)}_{\bar{x}\bar{x}} - \Phi^{(1)}_{\bar{t}\bar{t}} = 0 \tag{2.2.1.14}$$

which has the dispersion relation $w^2 = k^2(1+k^2)^{-1}$. Therefore, for small k ($k = \varepsilon^p k$, $p > 0$) the first two terms of w(k) are k and

k^3 terms. Then, we rescale $\bar{x}$ and $\bar{t}$ by defining ξ and τ to be:

$$\xi = \varepsilon^p(\bar{x} - a\bar{t}) \quad ; \qquad \tau = \varepsilon^{3p}\,\bar{t} \tag{2.2.1.15}$$

where a is the velocity of the frame of reference

After substitution into equation (2.2.1.12) we find that:

$$\left[\varepsilon^{3p}\frac{\partial}{\partial\tau} - a\varepsilon^p\frac{\partial}{\partial\xi}\right]\left[1 + \varepsilon n^{(1)} + \varepsilon^2 n^{(2)} + \ldots\right] +$$

$$\varepsilon^p\frac{\partial}{\partial\xi}\left[\varepsilon v^{(1)} + \varepsilon^2(v^{(2)} + n^{(1)}v^{(1)}) + \ldots\right] = 0 \tag{2.2.1.16}$$

$$\left[\varepsilon^{3p}\frac{\partial}{\partial\tau} - a\varepsilon^p\frac{\partial}{\partial\xi}\right]\left[\varepsilon v^{(1)} + \varepsilon^2 v^{(2)} + \ldots\right] +$$

$$\frac{1}{2}\varepsilon^p\frac{\partial}{\partial\xi}\left[\varepsilon^2(v^{(1)})^2 + \ldots\right] + \varepsilon^p\frac{\partial}{\partial\xi}\left[\varepsilon\Phi^{(1)} + \varepsilon^2\Phi^{(2)} + \ldots\right] = 0 \tag{2.2.1.17}$$

$$\varepsilon^{2p}\frac{\partial^2}{\partial\xi^2}\left[\varepsilon\Phi^{(1)} + \varepsilon^2\Phi^{(2)} + \ldots\right] - \varepsilon(\Phi^{(1)} - n^{(1)}) -$$

$$- \varepsilon^2\left[\Phi^{(2)} - n^{(2)} + \frac{1}{2}(\Phi^{(1)})^2\right] + \ldots = 0 \tag{2.2.1.18}$$

Collecting terms together we can evaluate the coefficients of each power of ε, we obtain for equation (2.2.1.16):

$$\varepsilon^{p+1} : \quad -an_\xi^{(1)} + v_\xi^{(1)} \tag{2.2.1.19}$$

$$\varepsilon^{p+2} : \quad -an_\xi^{(2)} + v_\xi^{(2)} + (n^{(1)}v^{(1)})_\xi \tag{2.2.1.20}$$

$$\vdots$$

$$\varepsilon^{3p+1} : \quad n_\tau^{(1)} \tag{2.2.1.21}$$

and for equation (2.2.1.17):

$$\varepsilon^{p+1} : \quad -av_{\xi}^{(1)} + \Phi_{\xi}^{(1)} \tag{2.2.1.22}$$

$$\varepsilon^{p+2} : \quad -av_{\xi}^{(2)} + \Phi_{\xi}^{(2)} + v^{(1)}v_{\xi}^{(1)} \tag{2.2.1.23}$$

$$\vdots$$

$$\varepsilon^{3p+1} : \quad v_{\tau}^{(1)} \tag{2.2.1.24}$$

while for equation (2.2.1.18), we have:

$$\varepsilon : \quad -(\Phi^{(1)} - n^{(1)}) \tag{2.2.1.25}$$

$$\varepsilon^{2} : \quad -(\Phi^{(2)} - n^{(2)}) - \frac{1}{2}(\Phi^{(1)})^{2} \tag{2.2.1.26}$$

$$\varepsilon^{2p+1} : \quad \Phi_{\xi\xi}^{(1)} \tag{2.2.1.27}$$

$$\varepsilon^{2p+2} : \quad \Phi_{\xi\xi}^{(2)} \tag{2.2.1.28}$$

Considering the lowest terms (i.e. ε^{p+1} and ε) and using the boundary condition, we have:

$$n^{(1)} = v^{(1)} = \Phi^{(1)} \tag{2.2.1.29}$$

for $a = 1$. To determine p, we go to the next order (i.e. ε^{p+2} and ε^{2}) where equations (2.2.1.19) - (2.2.1.24) indicate that if $3p+1 > p+2$ (i.e. $p > \frac{1}{2}$) then no τ-derivative occurs at order (ε^{p+2}) . This is unsatisfactory because $p > \frac{1}{2}$ means that the second derivative of $\Phi^{(1)}$ is an order higher than 2; and if we eliminate $\Phi^{(2)}$, $n^{(2)}$ and $v^{(2)}$ from the three equations, one can show that $n^{(1)} = 0$. Consequently, we must necessarily go to higher orders of perturbation theory to obtain an evolution equation for $n^{(1)}$. Therefore, setting $p = \frac{1}{2}$, and replacing $v^{(1)}$ and $\Phi^{(1)}$ by $n^{(1)}$, and equating terms of each order to zero, we obtain:

$$v_{\xi}^{(2)} - n_{\xi}^{(2)} + 2n^{(1)} n_{\xi}^{(1)} + n_{\tau}^{(1)} = 0 \qquad (2.2.1.30)$$

$$\Phi_{\xi}^{(2)} - v_{\xi}^{(2)} + n^{(1)} n_{\xi}^{(1)} + n_{\tau}^{(1)} = 0 \qquad (2.2.1.31)$$

$$n_{\xi\xi}^{(1)} - \frac{1}{2} (n^{(1)})^2 = \Phi^{(2)} - n^{(2)} \qquad (2.2.1.32)$$

Differentiating (2.2.1.32) w.r.t ξ and eliminating $v_{\xi}^{(2)}$, $n_{\xi}^{(2)}$, $\Phi_{\xi}^{(2)}$ from equations (2.2.1.30) - (2.2.1.32), we get:

$$\frac{1}{2} n_{\xi\xi\xi}^{(1)} + n^{(1)} n_{\xi}^{(1)} + n_{\tau}^{(1)} = 0 \qquad (2.2.1.33)$$

which is exactly the **KdV** equation, where the soliton velocities are positive

In the case a = -1 ,

$$v^{(1)} = - n^{(1)} = - \Phi^{(1)}$$

and the resulting **KdV** equation is:

$$\frac{1}{2} n_{\xi\xi\xi}^{(1)} + n^{(1)} n_{\xi}^{(1)} - n_{\tau}^{(1)} = 0$$

and the solitons move to the left as indicated by (2.2.1.15)

2.3 The Solution of the Korteweg-de Vries Equation

2.3.1 Introduction:

The **KdV** equation for long waves in shallow water [44] can be written as:

$$\mathbf{u}_t + \sqrt{gh_o}\left[1 + \frac{3}{2} (\mathbf{u}/ho) \right] \mathbf{u}_x + \frac{1}{6} \sqrt{gh_0}\, h_0^2\, \mathbf{u}_{xxx} = 0 \; ; \qquad (2.3.1.1)$$

Where; x denotes the coordinate along the horizontal bottom, t the time, $\mathbf{u}(x,t)$ the local wave-height above the undisturbed depth h_o, and g the acceleration of gravity and the subscripts x and t denote differentiation.

Let us define the non-dimensional parameters ε and μ to be:

$$\varepsilon = a/h_o \ , \qquad \mu = \frac{1}{6}(h_o/\lambda_o)^2$$

where a is the dominant amplitude and λ_o the wave-length. We introduce the dimensionless variables:

$$\bar{\xi} = x/\lambda_o \ , \quad \bar{t} = t\sqrt{gh_o}/\lambda_o \ , \quad \bar{u} = \frac{3}{2}\, u/(\varepsilon h_o)$$

Substitution of these new variables into equation (2.3.1.1) and omitting the bars gives:

$$u_t + u_\xi + \varepsilon\, u\, u_\xi + \mu\, u_{\xi\xi\xi} = 0 \qquad (2.3.1.2)$$

Let us now define the new independent variable x:

$$x = \xi - t$$

into equation (2.3.1.2) which is thus transformed into the well known **KdV** equation:

$$u_t + \varepsilon\, u\, u_x + \mu\, u_{xxx} = 0 \qquad (2.3.1.3)$$

Let us look at the travelling wave problems where the effects of the nonlinearity and dispersion balance and, result in stable solitary wave solutions called solitons. A soliton has the following remarkable properties:

(i) In a collision with another soliton it preserves its original shape and speed, although a phase shift exists after the collision, and

(ii) A general initial profile after a long time breaks up into a train of solitons together with a disturbance which disperses with time.

Let us look at the effect of the nonlinearity and the dispersion of the **KdV** equation (2.3.1.3):

(i) Linearity + no dispersion:

$$u_t + c\, u_x = 0 \ , \quad c \text{ is constant}$$

The initial profile is transmitted at constant speed without change of shape. Collisions cannot take place since all the

initial profile travels with the same velocity

(ii) Linearity + dispersion:

$$u_t + \mu u_{xxx} = 0$$

the solution can be expressed in the form:

$$u = a \exp(i(kx - \omega t)) = a \exp(i(x - ct)k)$$

where k is the wave number, ω is the frequency and $c = \omega/k$ is the speed of the travelling wave. This leads to the dispersion relation $\omega = -\mu k^3$, i.e. the group velocity depends on the wave number. The effect of the dispersion on a wave is to make a wave packet spread out as it travels. This dispersion rules out the possibility of solitary waves

(iii) Nonlinearity + no dispersion:

$$u_t + \varepsilon u u_x = 0$$

The term εu plays the role of a wave velocity. Since this velocity depends upon the solution itself, we may expect that portions of the wave profile at which **u** is large will move more rapidly than portions of the wave near the edge of the profile where **u** approaches zero. Thus the portion with large **u** will overtake the portion with smaller **u**.

(iv) Nonlinearity + dispersion:

$$u_t + \varepsilon u u_x + \mu u_{xxx} = 0$$

If there is a balance between nonlinearity and dispersion, then we obtain a soliton which travels without change of shape.

The generalised **KdV** equation has the form:

$$u_t + \varepsilon u^p u_x + \mu u_{xxx} = 0 \qquad (2.3.1.4)$$

where p is positive integer $p = 1, 2, 3, \ldots$ [32, 45, 46]. The most important case after $p = 1$, is $p = 2$, when the resulting equation has the form:

$$u_t + \varepsilon u^2 u_x + \mu u_{xxx} = 0 \qquad (2.3.1.5)$$

and is known as the modified Korteweg-de Vries (m**KdV**) equation. Moreover, the sign of the nonlinear term may be changed to obtain the non-trivial alternative equation:

$$u_t - \varepsilon u^2 u_x + \mu u_{xxx} = 0 \qquad (2.3.1.6)$$

Note that changing the sign of the nonlinear term in the **KdV** equation itself yields nothing new, since the resulting equation is reduced to (2.3.1.3) by changing the sign of **u** [45].

2.3.2 Single Soliton Solutions [40,41,42,47]:

The most interesting feature of the **KdV** equation is its ability to produce steady travelling wave solutions. These can be either solitary waves called solitons or the cnoidal wave, which is a generalisation of the sinusoidal wave. These are obtained by putting:

$$u(x,t) = U(X) \quad ; \quad X = x - ct \qquad (2.3.2.1)$$

where c represents the constant velocity of a wave travelling in the positive direction of the x-axis. Substitution of (2.3.2.1) into (2.3.1.4) leads to the ordinary differential equation:

$$-c U' + \varepsilon U^p U' + \mu U''' = 0 \qquad (2.3.2.2)$$

where a prime denotes differentiation with respect to X. Equation (2.3.2.2) can be integrated once immediately, to give:

$$\mu U'' = c U - \frac{\varepsilon}{p+1} U^{P+1} + a_1 \qquad (2.3.2.3)$$

where a_1 is the constant of integration

Multiplying (2.3.2.3) by U' and integrating, we obtain:

$$\frac{\mu}{2}\left[U'\right]^2 = \frac{c}{2} U^2 - \frac{\varepsilon}{(p+1)(p+2)} U^{P+2} + a_1 U + a_2 \qquad (2.3.2.4)$$

where a_2 is a constant of integration. For a real solution of (2.3.2.4), the right hand side must be non-negative, and so:

$$\int \frac{\left(\frac{\mu}{2}\right)^{1/2} dU}{\left[\frac{c}{2}U^2 - \frac{\varepsilon}{(p+1)(p+2)}U^{p+2} + a_1 U + a_2\right]^{1/2}} = \pm(x - ct) \qquad (2.3.2.5)$$

Two types of solution of (2.3.2.5) are cnoidal waves, which are expressed as Jacobi elliptic functions (see [21] for more details and the exact form), and solitary waves [7,33].

Now we are going to derive the solution of the equation (2.3.2.5) for the solitary waves. To do this, let U'', U' and $U \longrightarrow 0$ as $|x| \longrightarrow \infty$. Then the constants of integration a_1 and a_2 are zero, i.e.:

$$U' = U\left[\frac{1}{\mu}\left(c - \frac{2\varepsilon}{(p+1)(p+2)}U^p\right)\right]^{1/2} \qquad (2.3.2.6)$$

Let

$$y = \left[1 - \beta U^p\right]^{1/2}, \text{ where } \beta = \frac{2\varepsilon}{c(p+1)(p+2)}, \text{ then:}$$

$$U = \left[\frac{1 - y^2}{\beta}\right]^{1/p} \quad \text{and} \quad dU = -\frac{2y}{p\beta}\left[\frac{1 - y^2}{\beta}\right]^{\frac{1-p}{p}} \qquad (2.3.2.7)$$

Substitution of (2.3.2.7) into (2.3.2.6) leads to:

$$-\frac{2}{p}\left(\frac{\mu}{c}\right)^{\frac{1}{2}} \int \frac{dy}{1 - y^2} = \int dX \qquad (2.3.2.8)$$

and after integration of this equation, we obtain:

$$\ln\left(\frac{1 - y}{1 + y}\right) = p\left(\frac{c}{\mu}\right)^{\frac{1}{2}} X + c_1 \qquad (2.3.2.9)$$

Applying the initial condition leads to $c_1 = -p\left(\frac{c}{\mu}\right)^{1/2} x_0$ so that equation (2.3.2.9) can be written as:

$$\frac{1 - y}{1 + y} = \exp\left[p \left[\frac{c}{\mu} \right]^{\frac{1}{2}} (X - x_0) \right] \qquad (2.3.2.10)$$

If we let $X' = X - x_0$, $p = p \sqrt{c/\mu}$, we get:

$$y = \frac{1 - \exp(pX')}{1 + \exp(pX')} \qquad (2.3.2.11)$$

Since $U^p = (1 - y^2)/\beta$ we find that:

$$U^p = \frac{1}{\beta} \left[\frac{(1 + \exp(pX'))^2 - (1 - \exp(pX'))^2}{(1 + \exp(pX'))^2} \right]$$

$$= \frac{1}{\beta} \frac{4 \exp(pX')}{(1 + \exp(pX'))^2} \qquad (2.3.2.12)$$

or

$$U^p = \frac{1}{\beta} \frac{4}{(\exp(pX'/2) + \exp(-pX'/2))^2}$$

$$= \frac{1}{\beta} \operatorname{sech}^2(pX'/2) \qquad (2.3.2.13)$$

which simplifies to:

$$u^p(x, t) = \frac{c(p+1)(p+2)}{2\varepsilon} \operatorname{sech}^2\left[\frac{p}{2} \sqrt{c/\mu} (x - ct - x_0) \right] \qquad (2.3.2.14)$$

For $p = 1$ we have the well known solution:

$$u(x, t) = \frac{3c}{\varepsilon} \operatorname{sech}^2\left[\frac{1}{2} \sqrt{c/\mu} (x - ct - x_0) \right] \qquad (2.3.2.15)$$

Equation (2.3.2.14) describes a soliton with amplitude $3c/\varepsilon$, which is proportional to its velocity. Hence, a larger soliton moves faster than a smaller one. The solitons width is proportional to $\sqrt{\mu/c}$ and the constant x_0 plays the role of a phase

shift.

If the coefficient of the nonlinear term in equation (2.3.1.4) has a negative sign and p is odd then the solution is negative, that is:

$$u^p(x,t) = - \frac{c(p+1)(p+2)}{2\varepsilon} \operatorname{sech}^2\left[\frac{p}{2} \sqrt{c/\mu} \, (x - ct - x_0)\right] \quad (2.3.2.16)$$

If n is even, the solution is not a solitary wave [32,41,45].

Chen [48] has used Galerkin's method to obtain analytic solutions of the strongly nonlinear **KdV** equation:

$$u_t + \varepsilon u^3 u_x + \mu u_{xxx} = 0 \quad . \quad (2.3.2.17)$$

2.3.3 Linear Bargmann Method [42]:

The Bargmann method is based on the assumption that there exists a potential for the Schrodinger equation:

$$y'' + (k^2 - u)y = 0 \quad (2.3.3.1)$$

where k^2 is an eigenvalue parameter which remains constant as t varies and **u** satisfies the **KdV** equation:

$$u_t - 6 u u_x + u_{xxx} = 0 \quad (2.3.3.2)$$

such that the solution of the equation (2.3.3.1) can be expressed in the form:

$$y = \exp(ikx) F(k,x) \quad (2.3.3.3)$$

where F(k,x) is a polynomial in k, whose degree depends on the case under consideration. If F(k,x) is constant, then (2.3.3.1) implies that **u** = 0 which is the trivial solution. For the nontrivial solution let $F(k,x) = 2k + ia(x)$, then equation (2.3.3.1) implies that:

$$a' = - u \quad (2.3.3.4)$$

and $$a'' = u\, a \quad (2.3.3.5)$$

Eliminating **u** from (2.3.3.4) and (2.3.3.5) and integrating we

obtain:

$$a' + \frac{1}{2} a^2 = 2\mu^2 \qquad (2.3.3.6)$$

where $2\mu^2$ is the constant of integration. The substitution:

$$a = (2w')/w \qquad (2.3.3.7)$$

leads to the linear equation:

$$w'' - \mu^2 w = 0 \qquad (2.3.3.8)$$

which is a linear homogeneous equation of second order, whose solution is:

$$w = \alpha \exp(\mu x) + \beta \exp(-\mu x) . \qquad (2.3.3.9)$$

From equations (2.3.3.4) and (2.3.3.7) we get:

$$u = -2 (\ln(w))'' \qquad (2.3.3.10)$$

Using equation (2.3.3.9) the solution (2.3.3.10) becomes:

$$u = -2\mu^2 \operatorname{sech}^2(\mu x - \phi) \qquad (2.3.3.11)$$

where $\phi = \frac{1}{2} \ln(\beta/\alpha)$ which we take as a function of t. Substitution of (2.3.3.11) into the **KdV** equation (2.3.3.2) leads to:

$$\phi'(t) = 4\mu^3 \qquad (2.3.3.12)$$

Integrating this equation gives:

$$\phi(t) = 4\mu^3 t + d \qquad (2.3.3.13)$$

and hence we obtain:

$$u(x,t) = -2\mu^2 \operatorname{sech}^2(\mu x - 4\mu^3 t - d) \qquad (2.3.3.14)$$

If we set $\mu = \frac{1}{2}\sqrt{c}$, this form of $u(x,t)$ agrees with the steady state solution given in (2.3.3.10). This linear Bargmann's procedure thus yields the single soliton solution of the **KdV** equation.

A wide class of exact solutions to the **KdV** equation have been found, notably in recent times using the Inverse Scattering method. This method generates the well known N-soliton solutions possessing the property that amplitudes and velocities, as well as the shapes, of individual solitons are preserved in a (nonlinear) interaction [21,25,26,42,49].

2.3.4 Interaction of two Solitons [42,47,49,50,51,52,53]:

Consider two solitary waves as initial condition placed on the real line with the taller one to the left of the shorter one. As the time increases, the greater speed of the taller wave means that it eventually catches up with the shorter one and they undergo a nonlinear interaction according to the **KdV** equation. The surprising result is that they emerge from the interaction completely preserved in shape and speed with only a shift in positions relative to where they would have been had no interaction taken place [21]. This phenomenon was observed first, experimentally by Russell [1] and numerically by Zabusky and Kruskal [27]. Because of their preservation of form during nonlinear interactions and their resemblance to particles, Zabusky and Kruskal [27] coined the name soliton for such waves. Zabusky [6] showed the exact interaction of two solitons numerically and Lax [23] gave the analytic proof of the soliton properties. Dodd [41], Lamb [42], Whitham [49], and Wadati [51] have derived an analytic solution for the **KdV** equation with $\varepsilon = 6.0$, $\mu = 1.0$ when the initial condition is two solitary waves. This solution has the form:

$$\mathbf{u}(x,t) = 2(\ln(F))_{xx} \qquad (2.3.4.1)$$

where:

$$F = 1 + \exp(\eta_1) + \exp(\eta_2) + \beta \exp(\eta_1 + \eta_2)$$

$$\beta = \left(\frac{\alpha_1 - \alpha_2}{\alpha_1 + \alpha_2} \right)^2 \qquad (2.3.4.2)$$

$$\eta_i = \alpha_i x - \alpha_i^3 t + d_i \qquad i = 1,2 \qquad (2.3.4.3)$$

Before the interaction the solution will be:

$$\mathbf{u}(x,t) = \frac{1}{2} \alpha_1^2 \operatorname{sech}^2(\eta_1) + \frac{1}{2} \alpha_2^2 \operatorname{sech}^2(\eta_2 - \Delta) \qquad (2.3.4.4)$$

where:

$$\Delta = \ln(1 / \beta) \quad . \tag{2.3.4.5}$$

After the interaction the solution becomes [41,49]:

$$u(x,t) = \frac{1}{2} \alpha_1^2 \, \mathrm{sech}^2(\eta_1 - \Delta) + \frac{1}{2} \alpha_2^2 \, \mathrm{sech}^2(\eta_2) \quad . \tag{2.3.4.6}$$

The location of the solitary waves α_1 and α_2 are:

(i) before the interaction:

$$\text{Solitary wave } \alpha_1 \text{ on } x = \alpha_1^2 t - d_1/\alpha_1$$

$$\text{Solitary wave } \alpha_2 \text{ on } x = \alpha_2^2 t - (d_2 - \Delta)/\alpha_2 \tag{2.3.4.6a}$$

(ii) after the interaction:

$$\text{Solitary wave } \alpha_1 \text{ on } x = \alpha_1^2 t - (d_1 - \Delta)/\alpha_1$$

$$\text{Solitary wave } \alpha_2 \text{ on } x = \alpha_2^2 t - d_2/\alpha_2 \quad . \tag{2.3.4.6b}$$

The interaction occurs in the neighbourhood of:

$$t = - \frac{s_1 - s_2}{\alpha_1^2 - \alpha_2^2} \qquad x = \frac{\alpha_1^2 s_2 - \alpha_2^2 s_1}{\alpha_1^2 - \alpha_2^2} \tag{2.3.4.7}$$

where: $s_i = d_i/\alpha_i \quad i = 1,2$

The forward and the backward phase shifts are defined respectively as:

$$\Delta_1 = \Delta/\alpha_1 \, , \qquad \Delta_2 = \Delta/\alpha_2 \quad \text{for } p = 1,2 \tag{2.3.4.8}$$

For the initial conditions, equation (2.3.4.1) is used at $t = 0$.

Similarly the exact solution of the **mKdV** equation (2.3.1.5) with $\varepsilon = 6.0$, $\mu = 1.0$ for two solitary waves has been found by Taha and Ablowitz [52] as:

$$u(x,t) = i(\ln(f^*/ f))_x \tag{2.3.4.9}$$

where * denotes a complex conjugate, and

$$f = 1 + i \exp(\eta_1) + i \exp(\eta_2) - \beta \exp(\eta_1 + \eta_2) \, ,$$

where β and η_j $(j=1,2)$ are defined by equations (2.3.4.2) and (2.3.4.3) respectively.

Before the interaction the solution will be:

$$u(x,t) = \alpha_1 \text{ sech } (\eta_1) + \alpha_2 \text{ sech } (\eta_2 - \Delta) \qquad (2.3.4.10)$$

After the interaction the solution becomes:

$$u(x,t) = \alpha_1 \text{ sech } (\eta_1 - \Delta) + \alpha_2 \text{ sech } (\eta_2) \quad . \qquad (2.3.4.11)$$

For the initial conditions, equation (2.3.4.9) is used at t = 0.0.

For the case of N-solitons, an analytic proof that they are unchanged after interaction has been given by using the inverse scattering method [45].

More generally, arbitrary initial conditions used with the **KdV** equation will evolve into a number of solitons moving off to the right and an oscillatory dispersing state moving off to the left. Because of the dependence of the soliton speed on its amplitude, the solitons will sort themselves out, eventually ending up as a parade of solitons moving to the right with monotonically increasing amplitudes from left to right. Those solutions involving only solitons, and showing no oscillatory behaviour, are called pure soliton solutions or N-soliton solutions [21].

2.4 Conservation Laws for the Korteweg-de Vries Equation [6,10,54]:

The **KdV** equation can be written in the divergence form:

$$u_t + (\varepsilon \frac{u^2}{2} + \mu u_{xx})_x = 0 \qquad (2.4.1)$$

which has the form of a conservation law for the momentum $I_1 = \int_{-\infty}^{\infty} u(x,t)dx$. Multiplying both sides of equation (2.4.1) by u and u^2 , we obtain after simple calculation two more conservation laws, of which the first reflects the energy conservation:

$$\left(\frac{u^2}{2} \right)_t + \left[\varepsilon \frac{u^3}{3} + \mu \left(u\, u_{xx} - \frac{u_x^2}{2} \right) \right]_x = 0 \qquad (2.4.2)$$

$$\left(\frac{u^3}{3} - \frac{\mu}{\varepsilon} u_x^2 \right)_t + \left[\varepsilon \frac{u^4}{4} + \mu \left(u^2 u_{xx} + \frac{2}{\varepsilon} u_t u_x \right) + \frac{\mu^2}{\varepsilon} u_{xx}^2 \right]_x \qquad (2.4.3)$$

This is not the complete number of conservation laws. It is shown in [48] that there exist an infinite number of conservation quantities (invariants) I_m corresponding to the **KdV** equation, given by:

$$I_m = \int_{-\infty}^{\infty} Q_m(x,t)\, dx \qquad (2.4.4)$$

whose densities $Q_m(x,t)$ satisfy the relations of the form:

$$\frac{\partial Q_m(x,t)}{\partial t} + \frac{\partial P_m(x,t)}{\partial x} = 0 \qquad (2.4.5)$$

$$m = 1, 2, 3, \ldots$$

where Q_m and P_m (fluxes) are functions of **u** and its spatial derivatives. Such relationships imply essentially that the integral of Q_m over all x remains constant in time. That is Q_m is a conserved quantity [55].

Conservation laws can be used in deriving a priori estimates and to obtain integrals of motion. For example, if the flux P_m is zero as $|x| \longrightarrow \infty$, then:

$$\int_{-\infty}^{\infty} Q_m(x,t)\, dx = \text{constant} \qquad (2.4.6)$$

Furthermore, the existence of infinitely many conservation laws certainly indicates that the **KdV** equation is of immense physical interest.

As examples, we present the first four densities of the conserved quantities [6, 10, 39, 54]:

$$Q_1 = u\,, \qquad Q_2 = \frac{u^2}{2}\,,$$

$$Q_3 = \frac{u^3}{3} - \frac{\mu}{\varepsilon}\,u_x^2\,, \qquad Q_4 = \frac{u^4}{4} - 3\,\frac{\mu}{\varepsilon}\,u\,u_x^2 + \frac{9}{5}\,\frac{\mu^2}{\varepsilon^2}\,u_{xx}^2\,. \qquad (2.4.7)$$

Hence the quantities I_i $(i = 1,\ldots,4)$ can be written as:

$$I_1 = \int_{-\infty}^{\infty} u\,dx \qquad (2.4.8)$$

$$I_2 = \int_{-\infty}^{\infty} u^2\,dx \qquad (2.4.9)$$

$$I_3 = \int_{-\infty}^{\infty} \left(u^3 - \frac{3}{\varepsilon}\,\mu\,u_x^2\right) dx \qquad (2.4.10)$$

$$I_4 = \int_{-\infty}^{\infty} \left[u^4 - \frac{12}{\varepsilon}\,\mu\,u\,u_x^2 + \frac{36}{5\varepsilon^2}\mu^2 u_{xx}^2\right] dx\,. \qquad (2.4.11)$$

For a modified **KdV** equation (2.3.1.4) there are also many polynomial conservation laws. The first four conservative quantities have been found by [21,54] as:

$$I_1 = \int_{-\infty}^{\infty} u\,dx \qquad (2.4.12)$$

$$I_2 = \int_{-\infty}^{\infty} u^2\,dx \qquad (2.4.13)$$

$$I_3 = \int_{-\infty}^{\infty} \left(u^4 - \frac{6}{\varepsilon}\,\mu\,u_x^2\right) dx \qquad (2.4.14)$$

$$I_4 = \int_{-\infty}^{\infty} \left[u^6 - \frac{30}{\varepsilon}\,\mu\,u^2 u_x^2 + \frac{18}{\varepsilon^2}\mu^2 u_{xx}^2\right] dx\,. \qquad (2.4.15)$$

For $p > 2$ there are only three conservation laws, (the first three conservative quantities) [6,10,21,32] which can be written as:

$$I_1 = \int_{-\infty}^{\infty} u\,dx \qquad (2.4.16)$$

$$I_2 = \int_{-\infty}^{\infty} u^2 \, dx \qquad (2.4.17)$$

$$I_3 = \int_{-\infty}^{\infty} \left(u^{p+2} - \frac{(p+1)(p+2)}{2\varepsilon} \mu u_x^2 \right) dx \, . \qquad (2.4.18)$$

CHAPTER 3

A REVIEW OF NUMERICAL METHODS FOR SOLVING THE KORTEWEG-DE VRIES EQUATION

3.1 Introduction:

The study of numerical methods for the solution of partial differential equations has enjoyed an intense period of activity over the last thirty years from both theoretical and practical points of view. Improvements in numerical techniques, together with the rapid advance in computer technology, have meant that many of the partial differential equations arising from engineering and scientific applications which were previously intractable, can now be routinely solved [56].

In this chapter, we shall focus our attention on making a survey of the numerical methods used by earlier authors for solving the Korteweg-de Vries (KdV) equation:

$$u_t + \varepsilon u u_x + \mu u_{xxx} = 0 \qquad (3.1.1)$$

where; ε and μ are positive parameters, u_t and u_x are the first derivative of u with respect to time and space respectively and u_{xxx} is the third derivative of u with respect to space.

We shall give a brief discussion of these numerical methods under 4 headings:

(1) Finite difference methods,

(2) Finite Fourier transform or pseudospectral methods,

(3) Fourier expansion methods, and

(4) Finite element methods.

3.2 Finite Difference Methods.

3.2.1 Introduction:

The finite difference methods are the most frequently used

and universally applicable. These methods are approximate in the sense that the derivatives at a point are approximated by difference quotients over a small interval [57].

There are two types of finite difference method:

(1) Explicit methods which are conditionally stable,

(i) Zabusky and Kruskal difference scheme,

(2) Implicit methods which are sometimes unconditionally stable,

(i) Goda difference scheme,

(ii) Hopscotch difference scheme, and

(iii) Kruskal difference scheme.

Taha and Ablowitz [58] have proposed a local difference scheme which is based on the inverse scattering transform, and a global difference scheme, both of which have a truncation error of order $O[(\Delta t)^2] + O[(\Delta x)^2]$ [58].

In order to obtain a finite difference replacement of the **KdV** equation (3.1.1) the region to be examined is divided into equal rectangular meshes with sides Δx and Δt parallel to the x- and t- axes respectively (see Figure 3.1). The function $u(x,t)$ is approximated by $u_j^n = u(j\Delta x, n\Delta t)$ where j and n are integers and $j = n = 0$ is the origin.

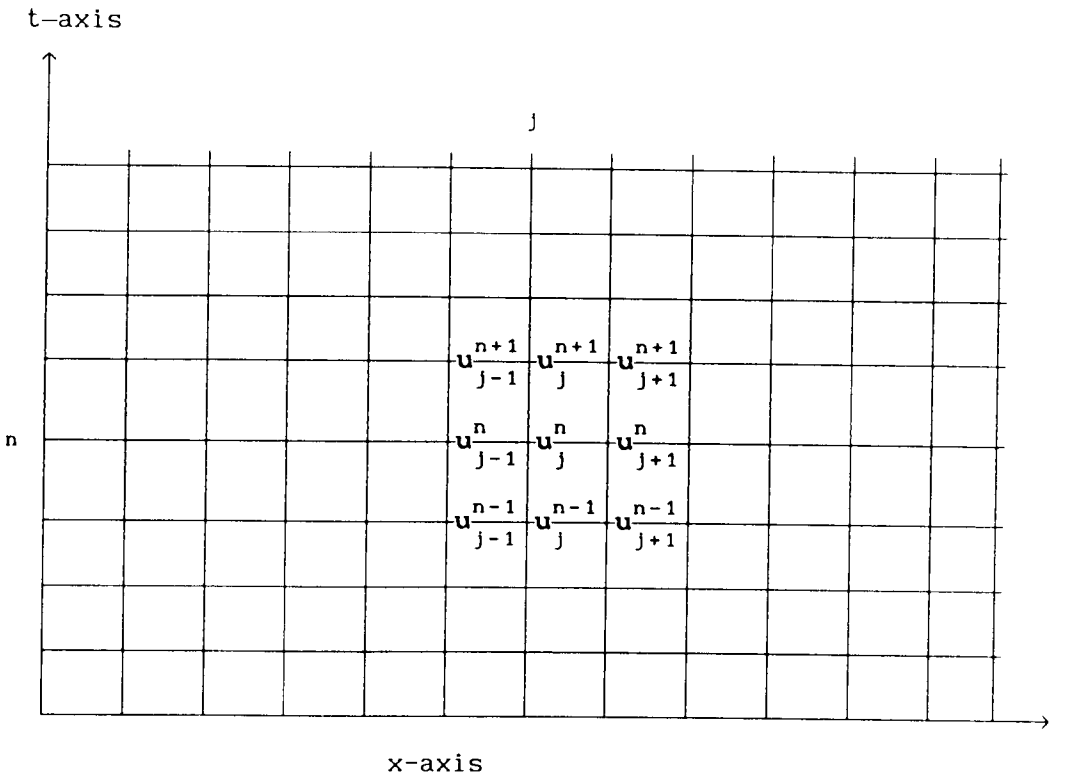

Figure 3.1

Let us define

$$u_t|_{(j,n)} = \frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} \qquad (3.2.1.1)$$

$$u_x|_{(j,n)} = \frac{u_{j+1}^{n} - u_{j-1}^{n}}{2\Delta x} \qquad (3.2.1.2)$$

$$u_{xxx}|_{(j,n)} = \frac{1}{2(\Delta x)^3}\left[u_{j+2}^{n} - 2u_{j+1}^{n} + 2u_{j-1}^{n} - u_{j-2}^{n}\right] \qquad (3.2.1.3)$$

3.2.2 Explicit Scheme:

The explicit scheme computes the value of the numerical solution at the forward time step in terms of known values at the previous time step.

An explicit scheme for solving the **KdV** equation produced originally by Zabusky and Kruskal [27,44] is centred in time and space. Substituting (3.2.1.1), (3.2.1.2) and (3.2.1.3) into

(3.1.1) with $u|_{(j,n)} = \frac{1}{3}(u^n_{j-1} + u^n_j + u^n_{j+1})$ leads to:

$$u^{n+1}_j = u^{n-1}_j - \frac{\varepsilon\Delta t}{3\Delta x}(u^n_{j-1} + u^n_j + u^n_{j+1})(u^n_{j+1} - u^n_{j-1}) - \frac{\mu\Delta t}{(\Delta x)^3}\left[u^n_{j+2} - 2u^n_{j+1} + 2u^n_{j-1} - u^n_{j-2}\right] \quad (3.2.2.1)$$

For the initial step, we use a scheme which is forward in time and centred in space:

$$u^1_j = u^0_j - \frac{\varepsilon\Delta t}{6\Delta x}(u^0_{j-1} + u^0_j + u^0_{j+1})(u^0_{j+1} - u^0_{j-1}) - \frac{\mu\Delta t}{2(\Delta x)^3}\left[u^0_{j+2} - 2u^0_{j+1} + 2u^0_{j-1} - u^0_{j-2}\right] \quad (3.2.2.2)$$

It is clear that equation (3.2.2.1) is a three-level scheme of time, i.e. in order to obtain u_j at the time level $n+1$, we need the following values of u_{j-2}, u_{j-1}, u_{j+1} and u_{j+2} at the previous time level n in addition to the value of u_j at the time level $n-1$.

The explicit difference scheme (3.2.2.1) has second order accuracy in Δt and Δx as the truncation error is $O[(\Delta t)^2] + O[(\Delta x)^2]$ and is also consistent with equation (3.1.1).

A stability analysis of the nonlinear numerical scheme (3.2.2.1) using the Fourier mode method is not easy to handle unless it is assumed that u, in the nonlinear term, is locally constant. This is equivalent to replacing the term $\frac{1}{3}(u^n_{j-1} + u^n_j + u^n_{j+1})$ in equation (3.2.2.1) by $\hat{u}$. This linearised scheme for the KdV equation has stability condition [44,58,59]:

$$\frac{\Delta t}{\Delta x}\left[\varepsilon\,|\hat{u}| + \frac{4\mu}{(\Delta x)^2}\right] \leq 1 \quad (3.2.2.3)$$

Peregrine [60] has produced another finite difference scheme

for the KdV equation which has only first order accuracy.

3.2.3 Implicit Methods:

Although the explicit method is computationally simple, it has one serious drawback. The time step is necessarily very small to satisfy the stability condition (3.2.2.3) and the space step must be kept small in order to attain reasonable accuracy.

We are going to give a brief discussion of the following implicit methods:

(i) Goda scheme,

(ii) Hopscotch method, and

(iii) Kruskal scheme.

3.2.3.1 Goda Scheme:

Goda [59] has proposed an unconditionally stable implicit scheme for approximating the KdV equation (3.1.1), namely:

$$\frac{1}{\Delta t}\left(u_j^{n+1} - u_j^n\right) + \frac{\varepsilon}{6\Delta x}\left[(u_j^n + u_{j+1}^n)u_{j+1}^{n+1} - (u_{j-1}^n + u_j^n)u_{j-1}^{n+1}\right] + \frac{\mu}{2(\Delta x)^3}\left[u_{j+2}^{n+1} - 2u_{j+1}^{n+1} + 2u_{j-1}^{n+1} - u_{j-2}^{n+1}\right] = 0 \qquad (3.2.3.1.1)$$

where n , j are integers n , $j = 0,1,\ldots,N$. We can rearrange this equation as:

$$-\alpha\, u_{j-2}^{n+1} + z_{j-1}u_{j-1}^{n+1} + u_j^{n+1} + y_{j+1}u_{j+1}^{n+1} + \alpha\, u_{j+2}^{n+1} = u_j^n \qquad (3.2.3.1.2)$$

where:

$$\alpha = \frac{\mu\Delta t}{2(\Delta x)^3} \quad , \quad \beta = \frac{\varepsilon\Delta t}{6\Delta x} \, ,$$

$$z_{j-1} = 2\alpha - \beta(u_{j-1}^n + u_j^n) \, , \quad y_{j+1} = -2\alpha + \beta(u_j^n + u_{j+1}^n) \qquad (3.2.3.1.3)$$

If there are $N+1$ internal mesh points along each row, then

for $n = 0$ and $j = 0,1,\ldots,N$, equation (3.2.3.1.2) gives $N+1$ simultaneous equations for $N+1$ unknown values of $\mathbf{u}$ along the first time row in terms of known initial and boundary values. Similarly, $n = 1$ and $j = 0,1,\ldots,N$ gives us $N+1$ unknown values of $\mathbf{u}$ along the second time row in terms of the calculated values along the first row, and so on. Generally, the scheme (3.2.3.1.2) can be expressed in terms of the matrix form:

$$A \underset{\sim}{u}^{n+1} = \underset{\sim}{u}^{n} \qquad (3.2.3.1.4)$$

where A is a 5-banded matrix of order $(N+1)x(N+5)$:

$$A = \begin{bmatrix} -\alpha & z_{-1} & 1 & y_0 & \alpha & & & & \\ & -\alpha & z_0 & 1 & y_1 & \alpha & & & \\ & & \cdot & \cdot & \cdot & \cdot & \cdot & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & -\alpha & z_{N-2} & 1 & y_{N-1} & \alpha \\ & & & & & & & -\alpha & z_{N-1} & 1 & y_N & \alpha \end{bmatrix}$$

$\underset{\sim}{u}^{n+1}$ is an unknown vector:

$$\underset{\sim}{u}^{n+1} = \left(u_{-2}^{n+1}, u_{-1}^{n+1}, \ldots, u_{N+1}^{n+1}, u_{N+2}^{n+1} \right)^T \qquad (3.2.3.1.5)$$

and $\underset{\sim}{u}^{n}$ is the known vector:

$$\underset{\sim}{u}^{n} = \left(u_0^n, u_1^n, \ldots, u_{N-1}^n, u_N^n \right)^T \qquad (3.2.3.1.6)$$

The 5-banded matrix can be reduced to penta-diagonal form by applying the boundary conditions at the end points, which means reducing the system from $N+1$ simultaneous equations in $N+5$ unknowns to $N+1$ equations in $N+1$ unknowns. Hence, the system can be solved by a penta-diagonal algorithm (see Appendix A2).

The implicit numerical scheme (3.2.3.1.2) is consistent and

is first order accurate in Δt and second order accurate in Δx, as the truncation error is $O[(\Delta t)] + O[(\Delta x)^2]$.

The stability analysis of the implicit numerical scheme (3.2.3.1.2) has been discussed by Goda [59] who proved that $\|u^{n+1}\| \leq \|u^n\|$, which implies that the scheme (3.2.3.1.2) is unconditionally stable.

3.2.3.2 Hopscotch Method:

In 1976, Greig and Morris [26] proposed a Hopscotch scheme for the KdV equation (3.1.1). To describe the method, discretize the space variable x into steps of size Δx and let $x = j\Delta x$, $j = 0, 1, \ldots, N$. Let $u = u(j\Delta x, n\Delta t)$ be the difference solution at the mesh point (j, n) and denote $f(u) = \frac{1}{2} u^2$ at mesh point (j, n) by $f_j^n = f(u_j^n)$. Their scheme is then:

$$u_j^{n+1} = u_j^n - \frac{\varepsilon \Delta t}{2\Delta x} (f_{j+1}^n - f_{j-1}^n) - \frac{\mu \Delta t}{2(\Delta x)^3} \left[u_{j+2}^n - 2u_{j+1}^n + 2u_{j-1}^n - u_{j-2}^n \right] \quad (3.2.3.2.1)$$

$$u_j^{n+1} = u_j^n - \frac{\varepsilon \Delta t}{2\Delta x} (f_{j+1}^{n+1} - f_{j-1}^{n+1}) - \frac{\mu \Delta t}{2(\Delta x)^3} \left[u_{j+2}^{n+1} - 2u_{j+1}^{n+1} + 2u_{j-1}^{n+1} - u_{j-2}^{n+1} \right] \quad (3.2.3.2.2)$$

To implement the scheme, we employ (3.2.3.2.1) for those mesh points for which $j+n$ is even and (3.2.3.2.2) for those for which $j+n$ is odd.

In their paper, Greig and Morris [26] have assumed that $u_{-1} = u_0 = u_N = u_{N+1} = 0$ for all t. The values obtained from equation (3.2.3.2.1) are now used in equation (3.2.3.2.2). Hence, rearranging (3.2.3.2.2), we have:

$$u_j^{n+1} + \frac{\mu\Delta t}{2(\Delta x)^3}\left[u_{j+2}^{n+1} - u_{j-2}^{n+1}\right] = u_j^n - \frac{\varepsilon\Delta t}{2\Delta x}\,(f_{j+1}^{n+1} - f_{j-1}^{n+1})$$

$$+ \frac{\mu\Delta t}{(\Delta x)^3}\left[u_{j+1}^{n+1} - u_{j-1}^{n+1}\right] \qquad (3.2.3.2.3)$$

This algorithm, under the present assumption that n is even, is to be applied for $j = 1,3,5,\ldots,N-2$. All the entries on the right hand side of (3.2.3.2.3) are known, hence it can be written as K_j^n . So (3.2.3.2.3) becomes:

$$u_j^{n+1} + \frac{\mu\Delta t}{2(\Delta x)^3}\left[u_{j+2}^{n+1} - u_{j-2}^{n+1}\right] = K_j^n \qquad (3.2.3.2.4)$$

This equation can be written in matrix form as:

$$A\,\underset{\sim}{u}^{n+1} = \underset{\sim}{K} \qquad (3.2.3.2.5)$$

where

$$A = \begin{bmatrix} 1 & \alpha^* & & & & & \\ -\alpha^* & 1 & \alpha^* & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & & & -\alpha^* & 1 & \alpha^* \\ & & & & & -\alpha^* & 1 \end{bmatrix}$$

$$\alpha^* = \frac{\mu\Delta t}{2(\Delta x)^3} \quad ,$$

$$\underset{\sim}{u}^{n+1} = \left[u_1^{n+1}, u_2^{n+1}, \ldots, u_{N-3}^{n+1}, u_{N-2}^{n+1} \right]^T ,$$

$$\underset{\sim}{K} = \left[k_1 , k_2 , \ldots , k_{N-3} , k_{N-2} \right]^T .$$

For n odd, we will obtain the obvious change of superscripts in the vector $\underset{\sim}{u}$ and $\underset{\sim}{K}$ in equation (3.2.3.2.5), but the coefficient

matrix A will remain unchanged.

The advantage of this scheme is that A is a tridiagonal matrix which means the system is also tridiagonal and so can be solved by the Thomas algorthim (see Appendix A1).

The stability of this Hopscotch method has been discussed by Greig and Morris [26] and they proved that it is stable if:

$$\frac{\Delta t}{\Delta x} \left| \varepsilon \hat{u} - \frac{2\mu}{(\Delta x)^2} \right| \leq 1 \qquad (3.2.3.2.6)$$

where $\hat{u}$ is the maximum value of u over the range of interest. Note that the condition of stability (3.2.3.2.6), is considerably less stringent than the stability condition for the Zabusky and Kruskal method (3.2.2.3)

The Hopscotch method has a truncation error of order $O[(\Delta t)^2] + O[(\Delta x)^2]$ and is consistent with equation (3.1.1)

3.2.3.3 Kruskal Method:

Kruskal [61] has suggested the numerical scheme:

$$\frac{1}{\Delta t}\left(u_j^{n+1} - u_j^n\right) + \frac{\mu}{2(\Delta x)^3}\left[u_{j+2}^{n+1} - 3u_{j+1}^{n+1} + 3u_j^{n+1} - u_{j-1}^{n+1}\right]$$

$$+ \frac{\mu}{2(\Delta x)^3}\left[u_{j+1}^{n} - 3u_j^{n} + 3u_{j-1}^{n} - u_{j-2}^{n}\right] = 0$$

(3.2.3.3.1)

for solving the linear differential equation:

$$u_t + \mu u_{xxx} = 0 \quad . \qquad (3.2.3.3.2)$$

Kruskal did not suggest any particular numerical scheme for the nonlinear term of the KdV equation (3.1.1). Taha and Ablowitz [58] proposed a numerical scheme to solve the KdV equation based on equation (3.2.3.3.1) and their own scheme for the nonlinear term. This leads to:

$$\frac{1}{\Delta t}\left(u_j^{n+1} - u_j^n\right) + \frac{\mu}{2(\Delta x)^3}\left[u_{j+2}^{n+1} - 3u_{j+1}^{n+1} + 3u_j^{n+1} - u_{j-1}^{n+1}\right]$$

$$+ \frac{\mu\Delta t}{2(\Delta x)^3}\left[u_{j+1}^{n} - 3u_j^{n} + 3u_{j-1}^{n} - u_{j-2}^{n}\right]$$

$$+ \frac{\epsilon}{4(\Delta x)}\left[\frac{\theta}{2}\left[(u^2)_{j+1}^{n+1} - (u^2)_{j-1}^{n+1} + (u^2)_{j+1}^{n} - (u^2)_{j-1}^{n}\right]\right.$$

$$\left. + (1-\theta)\left[(u_{j+1}^{n+1} - u_{j-1}^{n+1})u_j^{n+1} + (u_{j+1}^{n} - u_{j-1}^{n})u_j^{n}\right]\right] = 0$$

(3.2.3.3.3)

Several values of θ were employed and experimentally it was found that $\theta = \frac{2}{3}$ gave the best result.

This scheme is unconditionally stable according to linear stability theory, and has a truncation error of order $O[(\Delta t)^2] + O[(\Delta x)^2]$.

3.3 Finite Fourier Transform or Pseudospectral Methods.

3.3.1 Split Step Fourier Method by F. Tappert [58,62]:

For convenience, the spatial period 2p was normalised to $[0, 2\pi]$. Then the **KdV** equation becomes:

$$u_t + \frac{\varepsilon\pi}{p} u\, u_X + \mu\left(\frac{\pi}{p}\right)^3 u_{XXX} = 0 \qquad (3.3.1.1)$$

with $X = \frac{(x+p)\pi}{p}$

The essence of the solution method is to alternate between two steps:

(1) Advance the solution using only the nonlinear term by means of an implicit finite difference approximation, and

(2) Advance the solution using only the linear term by means of the discrete fast Fourier transform (FFT).

To implement this method to solve the **KdV** equation (3.3.1.1)

as the first step, we first approximate:

$$u_t + \frac{\varepsilon\pi}{p} u\, u_x = 0 \qquad (3.3.1.2)$$

The discretisation of this equation can be written as:

$$\tilde{u}_j^{n+1} = u_j^n - \frac{\varepsilon\pi\Delta t}{8p\Delta x}\left(\left[\tilde{u}^2 \right]_{j+1}^{n+1} - \left[\tilde{u}^2 \right]_{j-1}^{n+1} + \left[u^2 \right]_{j+1}^{n} - \left[u^2 \right]_{j-1}^{n} \right)$$

(3.3.1.3)

where $\tilde{u}$ is a solution of equation (3.3.1.2) and u is the solution of equation (3.3.1.1). For the second step, we would take:

$$u(X_j, t+\Delta t) = F^{-1}\left(e^{ik^3\pi^3/p^3\Delta t}\, F(\tilde{u}(X_j, t)) \right) \qquad (3.3.1.4)$$

where F denotes the discrete Fourier transform and F^{-1} its inverse. This scheme is unconditionally stable according to linear stability analysis, and has a truncation error of order $O[(\Delta t)^2] + O[(\Delta x)^2]$. In order to find $F(\tilde{u})$ and F^{-1}, the FFT technique is used.

Taha and Ablowitz [58] have found that an improved discretisation of (3.3.1.2) works considerably better. Specifically, the truncation error of the split step Fourier method is improved to the order of $O[(\Delta t)^2] + O[(\Delta x)^4]$ instead of $O[(\Delta t)^2] + O[(\Delta x)^2]$ by approximating equation (3.3.1.2) according to:

$$\tilde{u}_j^{n+1} = u_j^n - \frac{\varepsilon\pi\Delta t}{48p\Delta x}\left\{ 8\left[\tilde{u}^2 \right]_{j+1}^{n+1} - 8\left[\tilde{u}^2 \right]_{j-1}^{n+1} - \left[\tilde{u}^2 \right]_{j+2}^{n+1} + \left[\tilde{u}^2 \right]_{j-2}^{n+1} \right.$$

$$\left. + 8\left[u^2 \right]_{j+1}^{n} - 8\left[u^2 \right]_{j-1}^{n} - \left[u^2 \right]_{j+2}^{n} + \left[u^2 \right]_{j-2}^{n} \right\}$$

(3.3.1.5)

3.3.2 Pseudospectral Method by Fornberg and Whitham [32,58]:

This method is a Fourier method in which $u(x,t)$ is transformed into Fourier space with respect to x [63,64]. For convenience, the spatial period is normalised to $[0,2\pi]$. This interval is discretised by N equidistant points, with spacing $\Delta X = \frac{2\pi}{N}$. The function $u(X,t)$, numerically defined only on these points, can be transformed to the discrete Fourier space by:

$$\hat{u}(k,t) = Fu = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} u(j\Delta X, t)\, e^{-2\pi i jk/N}$$

$$k = -\frac{N}{2}, \ldots, -1, 0, 1, \ldots, \frac{N}{2} - 1 \qquad (3.3.2.1)$$

The inverse formula is:

$$u(j\Delta X, t) = F^{-1}\hat{u} = \frac{1}{\sqrt{N}} \sum_{k} \hat{u}(k,t)\, e^{2\pi i jk/N}$$

$$k = -\frac{N}{2}, \ldots, -1, 0, 1, \ldots, \frac{N}{2} - 1 \qquad (3.3.2.2)$$

These transformations can be performed efficiently with the fast Fourier transform algorithm [65,66,67]. With this scheme, u_X can be evaluated as $F^{-1}\{ikFu\}$, u_{XXX} as $F^{-1}\{(ik)^3 Fu\}$ and so on. Combined with a leap-frog time step, the KdV equation (3.3.1.1) would then be approximated by:

$$u(X,t+\Delta t) - u(X,t-\Delta t) + \frac{2i\varepsilon\pi\Delta t}{p} u(X,t)\, F^{-1}(kF(u)) - 2i\mu\Delta t \left(\frac{\pi}{p}\right)^3 F^{-1}(k^3 F(u)) = 0 \qquad (3.3.2.3)$$

Fornberg and Whitham [32] make a modification in the last term, and take:

$$u(X,t+\Delta t) - u(X,t-\Delta t) + \frac{2i\varepsilon\pi\Delta t}{p} u(X,t)\, F^{-1}(kF(u)) - 2i\mu\, F^{-1}\left\{ \sin\left(\left(\frac{\pi}{p}\right)^3 k^3 \Delta t\right) F(u) \right\} = 0 \qquad (3.3.2.4)$$

The difference between equation (3.3.2.3) and equation (3.3.2.4) is in the approximation of the linear equation:

$$u_t + \mu\left(\frac{\pi}{p}\right)^3 u_{XXX} = 0 \qquad (3.3.2.5)$$

The linear part of equation (3.3.2.4) is exactly satisfied for any solution of equation (3.3.2.5). It is found that the linearised stability condition for the scheme (3.3.2.4) is less restrictive than the scheme (3.3.2.3) in that we need only ensure

$$\frac{\Delta t}{(\Delta x)^3} < \frac{3}{2\pi^3} \simeq 0.1520 \text{ compared to } \frac{\Delta t}{(\Delta x)^3} < \frac{1}{\pi^3} \simeq 0.0323 \quad [32].$$

Since the Fornberg and Whitham [32] scheme is explicit, Taha and Ablowitz [58] consider a Crank-Nicolson implicit version e.g.:

$$u(X,t+\Delta t) - u(X,t) + \frac{i\varepsilon\pi\Delta t}{2p}\left\{u(X,t+\Delta t)\, F^{-1}(kF(u(X,t+\Delta t))) + u(X,t)\, F^{-1}(kF(u(X,t)))\right\} - \frac{i\mu\Delta t}{2}\frac{\pi^3}{p^3}\left\{F^{-1}(k^3F(u(X,t+\Delta t))) + F^{-1}(k^3F(u(X,t)))\right\} = 0 \qquad (3.3.2.6)$$

This scheme is unconditionally stable according to linear stability theory.

Taha and Ablowitz [58] have tested various numerical methods for solving the **KdV** equation (3.1.1), namely (i) Zabusky and Kruskal's scheme, (ii) Goda's scheme, (iii) the Hopscotch method, (iv) Kruskal's scheme, (v) the split step Fourier method of Tappert, (vi) the pseudospectral method of Fornberg and Whitham, (vii) Taha and Ablowitz's local scheme, and (viii) Taha and Ablowitz's global scheme. Two sets of initial conditions were used: (a) 1-soliton with various amplitudes, (b) the collision of two solitons with different values of the parameters μ and ε. From their numerical computations, they have drawn the following

conclusions:

(1) The scheme of Goda required more CPU time compared with the other schemes((i),(iii),(iv),(v),(vi),(vii)).

(2) Zabusky and Kruskal's scheme was good for small amplitudes but it needed more computing time than the other remaining methods ((iii),(iv),(v),(vi),(vii)) when amplitudes were large.

(3) The calculations for the above two methods, Goda's and Zabusky and Kruskal's, were not carried out for the 1-soliton case with an amplitude of 4. The CPU time required was too large.

(4) The Tappert and Hopscotch schemes took less computing time than the previous two schemes. For small amplitudes, Hopscotch was more efficient than Tappert, and they behave in almost the same way for medium amplitudes. On the other hand, for relatively large amplitudes, the Tappert scheme turned out to be better.

(5) Kruskal's scheme is in general faster than the schemes ((i),(ii),(iii),(v),(viii)).

(6) The Fornberg and Whitham method is much faster than the Kruskal scheme. It is roughly three times faster for small amplitudes and six times faster for large amplitudes.

(7) Taha and Ablowitz's local scheme is the best amongst all the described schemes above. It was roughly eight times faster than the Kruskal scheme and also it was roughly one and a half times faster than Fornberg and Whitham's scheme. Taha and Ablowitz's global scheme was faster than some of the utilized schemes, but much slower than its local version.

3.4 Fourier Expansion Method:

There are several methods for solving the **KdV** equation numerically, some of them are based on finite difference methods where we approximate all of the differentials by appropriate finite differences and reduce the partial differential equation to a set of algebraic equations. A numerical procedure competitive with the finite difference method is the Fourier expansion method. In this method, the unknown function is expanded in terms of a Fourier series and the original partial differential equation is reduced to a set of ordinary differential equations for the Fourier coefficients. We are going to give a summary of the Fourier expansion method.

Consider the **KdV** equation (3.1.1), with $\varepsilon = 1$, $\mu = 0.000484$, as followed in [29], where the initial periodic condition is:

$$u(x,0) = \cos(\pi x) \qquad 0 \le x \le 2 \tag{3.4.1}$$

The Fourier expansion corresponding to this is:

$$u(x,t) = \sum_{k=-\infty}^{\infty} a_k(t) \exp(i\pi k x) \tag{3.4.2}$$

with initial condition:

$$a_k(0) = \delta_{k,\pm 1}/2 \tag{3.4.3}$$

where $\delta_{k,m}$ is the Kronecker delta. Substituting equation (3.4.2) into equation (3.1.1), we obtain:

$$\frac{da_k}{dt} = -i\pi \sum_{m=-\infty}^{\infty} m\, a_{k-m}\, a_m + i\pi^3 \mu\, k^3 a_k$$

$$= -\frac{i\pi k}{2} \sum_{m=-\infty}^{\infty} a_{k-m}\, a_m + i\pi^3 \mu\, k^3 a_k \tag{3.4.4}$$

Using equations (3.4.3) and (3.4.4) leads to $a_0(t) = 0$ for all t.

Abe and Inoue [29] used the Runge-Kutta-Gill method for solving the set of ordinary differential equations (3.4.4). They found that the Fourier expansion method is the most accurate and effective method in comparison with the other methods [26,27,30].

In addition to the schemes noted above, there are other numerical schemes due to Gazdag [30], and a Taylor Fourier expansion method proposed by Canosa and Gazdag [31].

3.5 Finite Element Methods [68,69,70,71,72].

3.5.1 Introduction:

The term finite element was first used by Clough [73] in 1960. Since its inception, the literature on finite element applications has grown exponentially, and today there are numerous journals which are primarily devoted to the theory and applications of the finite element method [74].

The finite element method is now widely accepted as the first choice numerical method in all kinds of structural engineering applications in aerospace, naval architecture and the nuclear power industry. Applications to fluid mechanics are currently being developed for the study of tidal motion, thermal and chemical transport and diffusion problems, as well as for fluid-structure interactions.

During the nineteen-sixties, research on the finite element method was widely pursued simultaneously in various parts of the world, particulary in the following directions:

1) The method was reformulated as a special case of the weighted residual method,

2) A wide variety of elements were developed including bending elements, curved elements and the isoparametric concept was introduced, and

3) The method was recognised as a general method for the solution of partial differential equations. Its applicability to the solution of nonlinear and dynamic problems of structures was amply demonstrated as was its extension into other domains such as soil mechanics, fluid mechanics and thermodynamics. Solutions were obtained to engineering problems hitherto thought intractable [75].

In the finite difference approximation of a differential equation, the derivatives in the equation are replaced by difference quotients which involve the values of the solution at discrete mesh points of the domain. The resulting discrete equations are solved, after imposing the boundary conditions, for the values of the solution at the mesh points. Although the finite difference method is simple in concept, it suffers from several disadvantages. The most notable are the inaccuracy of the derivatives of the approximated solution, the difficulty in imposing the boundary conditions along nonstraight boundaries, the difficulty in accurately representing geometrically complex domains, and the inability to employ nonuniform and nonrectangular meshes.

The finite element method overcomes some of the difficulties of the finite difference method because it is based on integral formulations. The geometrical domain of the problem is represented as a collection of finite elements and can be divided into nonuniform and nonrectangular elements if the need arises [74].

Modern finite element integral formulations are mainly obtained by two different procedures: variational formulations and weighted residual formulations [76].

Variational models usually involve finding the nodal parameters that yield a stationary (maximum or minimum) value of a

specific integral relation known as a functional. It is well known that the solution that yields a stationary value of the functional and satisfies the boundary conditions, is equivalent to the solution of an associated differential equation known as the Euler equation. If the functional is known, then it is relatively easy to find the corresponding Euler equation.

Most engineering and physical problems are initially defined in terms of a differential equation. The finite element method requires an integral formulation so that one must search for the functional whose Euler equation has been given. Unfortunately, this is a difficult and sometimes impossible task, therefore there is an increasing emphasis on the various weighted residual techniques that can generate an integral formulation directly from the original differential equations.

The generation of finite element models by weighted residual techniques is a relatively recent development. However, these methods are increasingly important in the solution of differential equations.

Let us begin with finding an unknown function **u** which satisfies a certain operator equation:

$$A\,\mathbf{u} = f \qquad \text{in } \Omega = (a,b) \qquad (3.5.1.1)$$

where f is a known function and Ω is the domain of interest. A is a real differential operator of order 2m (m is postive). The differential operator A is linear if **u** and its derivatives appear linearly in A . Otherwise A is nonlinear.

The boundary conditions can contain the derivatives up to 2m - 1 and at each boundary point there are m boundary conditions. If the boundary conditions involve **u** and derivatives of order less than m then they are called essential. Otherwise they are natural.

In the weighted residual method the solution $\mathbf{u}$ is approximated by the interpolation functions ϕ_j through:

$$\mathbf{u}_N = \sum_{j=1}^{N} c_j \phi_j \qquad (3.5.1.2)$$

where c_j are unknown parameters to be determined.

The best choice of the approximated functions ϕ_j are polynomials because polynomials are easy to manipulate, both algebraically and computationally. Polynomials are also attractive from the point of view of the Weierstrass approximation theorem which states that any continuous function may be approximated, arbitrarly closely, by a suitable polynomial.

The choice of the approximation ϕ_j is required to satisfy the following conditions: The approximation must

(1) have geometrical invariance,

(2) contain a complete polynomial which includes all the lower terms, and

(3) have sufficient continuity and parameters to represent the solution.

Substitute the approximate solution (3.5.1.2) into the operator equation (3.5.1.1). This operation defines a residual R_N:

$$R_N = A\,\mathbf{u} - f \qquad (3.5.1.3)$$

where R_N is a function of the chosen independent functions ϕ_N and the unknown parameters c_j . To determine the unknown parameters c_j using the weighted residual method one can set the integral, over the domain Ω , of the product of the residual and some weight functions ψ_j to be zero:

$$\int_{\Omega} \psi_j R_N \, dx = 0 \qquad j = 1, \ldots, N \qquad (3.5.1.4)$$

where the weight functions, in general, are not the same as the approximation functions ϕ_j .

The equation (3.5.1.4) can be simplified to the form:

$$\sum_{j=1}^{N} \left(\int_{\Omega} \psi_i \, A \, \phi_j \, dx \right) c_j = \int_{\Omega} \psi_i \, f \, dx$$

or

$$\sum_{j=1}^{N} A_{ij} \, c_j = f_i \qquad (3.5.1.5)$$

where:

$$A_{ij} = \int_{\Omega} \psi_i \, A \, \phi_j \, dx$$

$$f_i = \int_{\Omega} \psi_i \, f \, dx$$

For different choices of the weight functions we obtain different types of the weighted residual technique (3.5.1.4).

For $\psi_i = \phi_i$, the weighted residual method (3.5.1.4) is called the Galerkin method while the weighted residual approach is called the Petrov-Galerkin method, if $\psi_i \neq \phi_i$.

To obtain the Least square method one determines the parameters c_i by minimising the integral of the square of the residual (3.5.1.4):

$$\frac{\partial}{\partial c_i} \int R_N^2 \, dx = 0$$

or

$$\int \frac{\partial R_N}{\partial c_i} R_N \, dx = 0 \qquad (3.5.1.6)$$

The equation (3.5.1.6) can be written in simplified form:

$$\sum_{j=1}^{N} \left(\int_{\Omega} A \, \phi_i \, A \, \phi_j \, dx \right) c_j = \int_{\Omega} (A \, \phi_i) \, f \, dx$$

or

$$\sum_{j=1}^{N} A_{ij} \, c_j = f_i \qquad (3.5.1.7)$$

where

$$A_{ij} = \int_{\Omega} (A\ \phi_i)\ (A\ \phi_j)\ dx$$

$$f_i = \int_{\Omega} (A\ \phi_i)\ f\ dx$$

Another popular method for solving the boundary value problem is the collocation method. The idea behind this approach is to make the residual in equation (3.5.1.3) identically zero at N selected points in the domain Ω:

$$R_N \equiv 0 . \qquad (3.5.1.8)$$

or

$$\sum_{j=1}^{N} c_j\ A\ \phi_j(x_i) = f(x_i) \qquad i=1,\ldots,N \qquad (3.5.1.9)$$

The equation (3.5.1.9) gives a system of N equations in the N unknown parameters c_j which can be solved numerically.

For both variational and weighted residual formulations, the following restrictions are generally accepted as a means of establishing convergence of the finite element model as the mesh is refined [76]:

(1) - (A necessary criterion) the element interpolation functions must be capable of modelling any constant values of the dependent variable, or its derivatives, to the order present in the defining integral statement, in the limit as the element size decreases.

(2) - (A sufficient criterion) the element shape functions should be chosen so that at element interfaces the dependent variable and its derivatives, of up to one order less than those occurring in the defining integral, statement, are continuous.

Let us introduce the basic terms which are used in the finite element analysis of any problem [74]:

1. Finite element discretisation. The continuous domain is

represented as a collection of a finite number N of subdomains, say segments. Each of these segments is called an element. The collection of elements is called the finite element mesh. If all the elements are of the same size, the mesh is said to be uniform. One can discretise the domain, depending on the shape of the domain, into a mesh of more than one type of element.

2. Error estimate. There are three sources of errors in a finite element solution:

(i) errors due to the approximation of the domain,

(ii) errors due to the approximation of the solution, and

(iii) errors due to numerical computation.

The estimation of these errors, in general, is not a simple task.

3. Number and location of the nodes. The number of the location of the nodes in an element depends on

(a) the geometry of the element,

(b) the degree of the approximation, and

(c) the variational form of the equation.

4. Assembly of elements. The assembly of elements, in a general case, is based on the idea that the solution and possibly its derivatives are continuous at the interelement boundaries.

5. Accuracy and convergence. The accuracy and convergence of the finite element solution depends on the differential equation solved and the elements used. The accuracy is the difference between the exact solution and the finite element solution, and the convergence is the accuracy as the number of elements in the mesh is increased. The convergence depends on the governing differential equation.

6. The time dependent problems. For time dependent problems, there are two steps to be followed:

Firstly, the differential equations are approximated by the finite

element method to obtain a set of ordinary differential equations in time.

Secondly, the differential equations in time are solved approximately by finite difference methods to obtain algebraic equations, which are then solved for the nodal values.

The basic steps for the solution of a differential equation using the finite element method is as follows [74]:

(1) Divide the given domain into a finite elements. Number the nodes (the points of subdomains where the function is evaluated) and the elements. Generate the geometric properties(e.g. coordinates, cross-sectional area ,...) needed for the problem.

(2) Evaluate the element equations by constructing a suitable weighted residual formula of the given differential equation using:

$$u = \sum_{i=1}^{N} u_i \psi_i \qquad (3.5.1.10)$$

where; ψ_i are the chosen interpolation functions.

If we substitute the equation (3.5.1.10) in the chosen weighted residual formula, we will obtain the formula:

$$\{ K^e \} \{ u^e \} = \{ F^e \} \qquad (3.5.1.11)$$

(3) Assemble the element contributions to obtain the equation for the whole problem.

(4) Impose the boundary conditions of the problem.

(5) Solve the overall system of equations.

(6) Compute the solution and represent the results in tabular and/or graphical form.

In the following sections we will give a brief discussion about the finite element approach by using the Galerkin method to solve the Korteweg-de-Vries equation.

3.5.2. Alexander and Morris Galerkin Method:

Alexander and Morris [35] proposed the Galerkin method, in which the trial and test functions were cubic splines for solving the KdV equation:

$$u_t + u\,u_x + \mu\,u_{xxx} = 0 \qquad (3.5.2.1)$$

Let S^m denote the space of smoothest splines, defined piecewise on intervals of length h (mesh size) as polynomials of order m (degree $m-1$). These spline functions can be constructed in the usual way as an $(m-1)$-fold convolution: Let:

$$M_1(X) = \begin{cases} 1 & -\frac{1}{2} \le X \le \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \qquad (3.5.2.2)$$

then

$$M_m(X) = M_1 * M_1 * M_1 * \ldots * M_1 \quad (m-1) \qquad (3.5.2.3)$$

where:

$$(f*g)X = \int_{-\infty}^{\infty} f(X)\,g(X-Y)dY \qquad (3.5.2.4)$$

Using (3.5.2.2), we see that

$$M_m(X) = \int_{X-\frac{1}{2}}^{X+\frac{1}{2}} M_{m-1}(Y)dY \qquad (3.5.2.5)$$

The basis functions $\phi_i(x)$ of S^m are then defined by:

$$\phi_i(x) = M_m\left(\frac{x}{h} - i\right) \qquad i \text{ is integer} \qquad (3.5.2.6)$$

Alexander and Morris proposed the generalised Galerkin method for solving the **KdV** equation (3.5.2.1): if $V \in S^m$, and U denotes the Galerkin solution, then:

$$\int_a^b \left[U_t + U\,U_x + \mu\,U_{xxx}\right]\left[V + q\,\mu\,h^3 V_{xxx}\right]dx = 0 \qquad (3.5.2.7)$$

where; q is an arbitrary parameter determining the amount of

dissipation in the scheme (q = 0 corresponds to nondissipative scheme).

In order to undertake a Fourier analysis of the accuracy and stability of (3.5.2.7), Alexander and Morris set $U = d$ in the nonlinear term $u\,u_x$ in the equation (3.5.2.1). Then they rescale x , t and u to remove the constant μ and obtain the linearised form:

$$\int_a^b \left[U_t + d\,U_x + U_{xxx}\right]\left[V + q\,h^3 V_{xxx}\right]dx = 0 \qquad (3.5.2.8)$$

The Galerkin solution may be expressed as:

$$U(x,t) = \sum_i \delta_i(t)\,\phi_i(x) \qquad (3.5.2.9)$$

where; $\phi_i(x)$ are cubic spline basis functions defined [35] by:

$$\phi_i(x) = M_4\left(\frac{x}{h} - i\right) \qquad (3.5.2.10)$$

where:

$$M_4(y) = \frac{1}{6}\begin{cases} 0 & y \le -2 \\ (y+2)^3 & -2 < y \le -1 \\ (y+2)^3 - 4(y+1)^3 & -1 < y \le 0 \\ (-y+2)^3 - 4(-y+1)^3 & 0 < y \le 1 \\ (-y+2)^3 & 1 < y \le 2 \\ 0 & 2 < y \end{cases} \qquad (3.5.2.11)$$

$\delta_i(t)$ are the unknown parameters to be determined, and y is a local variable. In addition to these cubic splines, quintic boundary polynomials are were specially constructed to maintain the continuity of the second derivative.

Substituting (3.5.2.9) into (3.5.2.8), setting $V = \phi_i$ and using Fourier analysis, Alexander and Morris proved that the scheme (3.5.2.8) is conditionally stable if $d > 0.0$; and unconditionally stable if $d < 0.0$. In the former case, they can make the scheme stable by making the value of h sufficiently small. This scheme is second order accurate.

A numerical computational procedure used for implementing the scheme (3.5.2.7) leads to the system:

$$A\dot{\underset{\sim}{\delta}} + B\underset{\sim}{\delta} = 0 \qquad (3.5.2.12)$$

where A and B are 7-banded matrices. This set of differential equations was solved using the IMSL library routine DREBS.

This system has been discussed by Alexander and Morris [35] for different values of q with initial conditions one soliton and two solitons and boundary conditions in which the solutions have their zeroth, first and second derivatives equal to zero on the boundaries.

Alexander and Morris computed the maximum error for a single soliton and they found that with $h = 0.05$ and $t = 0.39$, and exact time integration, a maximum error ranging between 0.025 and 0.059 according to the chosen value of the dissipation parameter; for $h = 0.033$ and $t = 0.046$ the error presented is of the order 0.015 . Also they conclude that a comparison with the Hopscotch method [26] shows that the Galerkin method has the advantage of smaller errors, for the same mesh size.

3.5.3 Petrov-Galerkin Method:

The Petrov-Galerkin method is a Galerkin method in which the test and trial functions are not the same. Sanz-Serna and Christie [36] in 1981 proposed a Petrov-Galerkin method in which the trial and test functions were chosen to be piecewise linear and cubic splines respectively.

We attempt to solve the **KdV** equation (3.5.2.1) together with the initial condition:

$$\mathbf{u}(x,0) = f(x) \qquad a \le x \le b \qquad (3.5.3.1)$$

Assume that the problem has a unique solution such that, for fixed

t , $u(x,t)$, together with all its x derivatives, tends to zero as $|x| \longrightarrow \infty$. Conditions on $f(x)$ guaranteeing existence and uniqueness are given by Lax [23] and Sjoberg [24].

If we employ the Galerkin method with weight function $v(x)$ and integrate by parts, we obtain:

$$\int_a^b \left[(u_t + u u_x) v + \mu u_x v_{xx} \right] dx = 0 \qquad (3.5.3.2)$$

We introduce finite elements in space in (3.5.3.2) and approximate the exact solution by the interpolation functions:

$$U(x,t) = \sum_{i=0}^{N} U_i(t) \, \phi_i(x) \qquad (3.5.3.3)$$

where $\phi_i(x)$, $i=0,1,\ldots,N$, are piecewise linear trial functions and $U_i(t)$ are unknown parameters to be determined from the system of ordinary differential equations:

$$\int_a^b \left[(U_t + U U_x) \psi_j + \mu U_x (\psi_j)_{xx} \right] dx = 0 \qquad (3.5.3.4)$$

where; ψ_j are piecewise cubic spline test functions defined [36] by:

$$\psi_j(x) = \psi((x - x_0)/h - j) \quad , \; j = 0,1, \ldots, N \qquad (3.5.3.5)$$

then:

$$\psi(x) = \frac{1}{6}\left[\sigma(x+1) + 4\sigma(x) + \sigma(x-1) + 3\rho(x+1) - 3\rho(x-1) \right] \qquad (3.5.3.6)$$

$$\sigma(x) = \begin{cases} (|x| - 1)^2 (2|x| + 1) & |x| \le 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\rho(x) = \begin{cases} x(|x| - 1)^2 & |x| \le 1 \\ 0 & \text{otherwise} \end{cases}$$

and the trial functions $\phi_i(x)$ are defined [36] as a piecewise linear hat function:

$$\phi_i(x_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

Sanz-Serna and Christie have discussed the following two cases:

(1) Linearised case:

Consider the linearised **KdV** equation:

$$u_t + d\,u_x + \mu\,u_{xxx} = 0 \qquad (3.5.3.7)$$

where; d is constant. With the test and trial functions defined previously, the linear analogue of equation (3.5.3.4) is:

$$A\,\dot{\underset{\sim}{U}} + B\,\underset{\sim}{U} = 0 \qquad (3.5.3.8)$$

where; $\underset{\sim}{U} = \left[U_0(t),\ U_1(t),\ \ldots,\ U_N(t)\right]^T$ and A, B are 5-band matrices of order $(N+1)\times(N+1)$. The system (3.5.3.8) is given explicitly by:

$$\frac{1}{60}\left[\alpha_1\dot{U}_{i+2} + \alpha_2\dot{U}_{i+1} + \alpha_3\dot{U}_i + \alpha_2\dot{U}_{i-1} + \alpha_1\dot{U}_{i-2}\right]$$

$$+ \frac{d}{24h}\left[\beta_1 U_{i+2} + \beta_2 U_{i+1} - \beta_2 U_{i-1} - \beta_1 U_{i-2}\right]$$

$$+ \frac{\mu}{2h^3}\left[U_{i+2} - 2U_{i+1} + 2U_{i-1} - U_{i-2}\right] = 0 \qquad (3.5.3.9)$$

where:

$\alpha_1 = 9\alpha - 1$, $\alpha_2 = 9 + 24\alpha$, $\alpha_3 = 44 - 66\alpha$,

$\beta_1 = 12\alpha - 1$, $\beta_2 = 14 - 24\alpha$,

$i = 0, 1, \ldots, N$ and we set $U_{-2} = U_{-1} = U_{N+1} = U_{N+2} = 0$

For general α , using Taylor expansion the explicit scheme (3.5.3.9) is second order accurate. But if $\alpha = \frac{1}{6}$ its accuracy becomes fourth order [36].

The stability analysis for the system (3.5.3.8) has been discussed using the von Neuman stability test and it is found that the system is unconditionally stable [36].

(2) Nonlinear case:

Returning to the **KdV** equation (3.5.2.1), the system (3.5.3.8) can be written as:

$$A \underset{\sim}{\dot{U}} + B(\underset{\sim}{U}) = 0 \qquad (3.5.3.10)$$

where; A and U are as defined in the linear case and B(U) is a nonlinear vector function. The ith component of equation (3.5.3.10) is of the form (3.5.3.9) except for the fact that the term involving d is replaced by:

$$\frac{1}{120h}\left[2\alpha_1 U_{i+2}^2 + \gamma_1 U_{i+2}U_{i+1} + \gamma_2 U_{i+1}^2 + \gamma_3 U_{i+1}U_i - \gamma_3 U_i U_{i-1} - \gamma_2 U_{i-1}^2 - \gamma_1 U_{i-1}U_{i-2} - 2\alpha_1 U_{i-2}^2 \right] \qquad (3.5.3.11)$$

where:

$$\gamma_1 = 24\alpha - 1 \quad , \quad \gamma_2 = 24 - 36\alpha \quad , \quad \gamma_3 = 23 - 72\alpha$$

Taylor expansion for any α renders the method second order accurate [33,36]

To increase the accuracy of this method Sanz-Serna and Christie have proposed an approximation to the nonlinear term:

$$U^2(x,t) = \sum_{i=0}^{N} U_i^2(t)\, \phi_i(x) \qquad (3.5.3.12)$$

using this definition the nonlinear term $\mathbf{u}\,\mathbf{u}_x$ can be replaced by:

$$\frac{1}{48h}\left[U_{i+2}^2 + 10U_{i+1}^2 - 10U_{i-1}^2 - U_{i-2}^2 \right]$$

So the ith component of the equation (3.5.3.10) becomes:

$$\frac{1}{120}\left[\dot{U}_{i+2} + 26\dot{U}_{i+1} + 66\dot{U}_{i} + 26\dot{U}_{i-1} + \dot{U}_{i-2}\right]$$

$$+ \frac{1}{48h}\left[U^2_{i+2} + 10U^2_{i+1} - 10U^2_{i-1} - U^2_{i-2}\right]$$

$$+ \frac{\mu}{2h^3}\left[U_{i+2} - 2U_{i+1} + 2U_{i-1} - U_{i-2}\right] = 0 \qquad (3.5.3.13)$$

Using Taylor expansion, the truncation error of the method is fourth order accurate [33,36].

The Sanz-Serna and Christie method has the disadvantage of a slight background noise in the form of a short wave length ripple of very small amplitude. It is only noticeable in those regions where the solution itself is very small, and would often not be considered to be a problem at all. In some applications, however, it may be desirable to have a smoother solution. In these cases a method which is slightly dissipative could be used [22,77,78].

The methods we propose in chapters four to eight are also finite element methods with some similarties to those discussed here. The results we obtain will be compared in detail with those obtained by previous authors.

CHAPTER 4

GALERKIN METHOD WITH HERMITE CUBIC FUNCTIONS

4.1 Introduction:

The cubic Hermite interpolation functions have continuous first derivatives, and for this reason one can chose them as basis functions to approximate the solution of the **KdV** equation. Another advantage of choosing the cubic Hermite functions as trial and test functions is that the first derivative is automatically computed at each mesh point.

The present chapter is devoted to solving the **KdV** equation using Galerkin's method with piecewise cubic Hermite trial and test functions.

4.2 The Governing Equation [77,79]:

The Korteweg-de-Vries equation is:

$$u_t + \varepsilon\, u\, u_x + \mu\, u_{xxx} = 0 \qquad a \le x \le b \tag{4.2.1}$$

where; ε and μ are positive parameters. The boundary conditions will be chosen from:

$$\left.\begin{aligned} u(a,t) &= \beta \\ u(b,t) &= 0 \\ u_x(a,t) &= u_x(b,t) = 0 \\ u_{xx}(a,t) &= u_{xx}(b,t) = 0 \end{aligned}\right\} \tag{4.2.2}$$

and the initial conditions on $u(x,t)$ will be prescribed in a later section.

Applying Galerkin's method with weight function $v(x)$, which is also assumed to satisfy the boundary conditions (4.2.2), gives:

$$\int_a^b v(u_t + \varepsilon u u_x + \mu u_{xxx})dx = 0. \qquad (4.2.3)$$

Integrating by parts, the term involving the third derivative and using the boundary conditions (4.2.2) leads to the equation:

$$\int_a^b [v(u_t + \varepsilon u u_x) - \mu v_x u_{xx}]dx = 0 \qquad (4.2.4)$$

The condition the integral imposes upon the interpolation functions, is now that those functions and their first derivatives must be continuous throughout the region. To satisfy this requirement we have chosen Hermite cubic polynomials as shape functions.

4.3 The Finite Element Solution:

Divide the region [a,b] into N equal finite elements of length h by the points x_i where:

$$a = x_0 < x_1 \ldots < x_N = b .$$

There are $N+1$ nodes at x_0 , x_1 ,..., x_N and a corresponding $N+1$ nodal parameters u_0 , u_1 ,...,u_N .

It will be assumed here that the variables of interest can be uniquely specified throughout the solution domain by the nodal parameters associated with the nodal points of the system. These nodal parameters will be the unknown parameters of the problem. It is assumed also that the parameters at a particular node are influenced only by the values of the quantity of interest within the elements that are connected to that particular node. Next, an interpolation function is assumed for the purpose of relating the values of parameters at all the nodes that are connected to a particular element.

To make these assumptions clear, let us consider initially a typical cubic Hermite element $[x_i , x_{i+1}]$ which has nodes at x_i

and x_{i+1} and at those points the nodal parameters are the values of the function u_i and its first derivative u_i' ($u' \equiv u_x$). Over the element the function u is given by:

$$u^e(x,t) = \underset{\sim}{N}(x)\ \underset{\sim}{\delta}^e(t) = \sum_{i=1}^{4} N_i\ \delta_i^e \qquad (4.3.1)$$

where $\underset{\sim}{\delta}^e = [u_i\ ,\ u_i'\ ,\ u_{i+1}\ ,\ u_{i+1}']^T$

is the vector of nodal parameters which is a function of time only and $\underset{\sim}{N}$ is defined [74,75] as:

$$\underset{\sim}{N} = \left[1 - \frac{3\xi^2}{h^2} + \frac{2\xi^3}{h^3},\ \xi - \frac{2\xi^2}{h} + \frac{\xi^3}{h^2},\ \frac{3\xi^2}{h^2} - \frac{2\xi^3}{h^3},\ -\frac{\xi^2}{h} + \frac{\xi^3}{h^2}\right] \qquad (4.3.2)$$

which is the shape function expressed in terms of a local coordinate $\xi = x - x_i$, $0 \le \xi \le h$

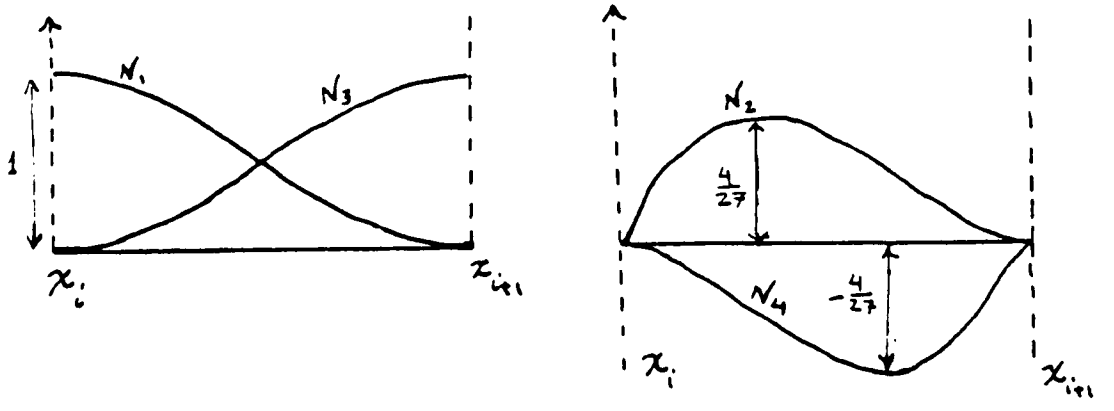

Figure 4.1 Typical element of shape function cubic Hermite.

Since equation (4.2.4) is valid over the whole region [a,b], it is valid in particular, over the element e so that the contribution to the equation (4.2.4) can be written:

$$\int_0^h [v(u_t + \varepsilon\, u\, u_x) - \mu\, v_x u_{xx}]\,dx \qquad (4.3.3)$$

which, if we identify the weight function v with a shape function $\underset{\sim}{N}$, and use equation (4.3.1) becomes:

$$\sum_{j=1}^{4}\left[\int_0^h N_i N_j dx\right]\dot{\delta}_j^e + \varepsilon \sum_{j=1}^{4}\sum_{k=1}^{4}\left[\int_0^h N_i N_j N_k' dx\right]\delta_k^e\delta_j^e - \mu \sum_{j=1}^{4}\left[\int_0^h N_i' N_j'' dx\right]\delta_j^e$$

(4.3.4)

Equation (4.3.4) can be written in matrix form as:

$$A^e \underset{\sim}{\dot{\delta}}^e + \varepsilon \underset{\sim}{\delta}^{eT} L^e \underset{\sim}{\delta}^e - \mu C^e \underset{\sim}{\delta}^e \qquad (4.3.5)$$

where the element matrices are given by the integrals:

$$A_{ij}^e = \int_0^h N_i N_j dx \ , \qquad (4.3.6)$$

$$C_{ij}^e = \int_0^h N_i' N_j'' dx \ , \qquad (4.3.7)$$

and

$$L_{ijk}^e = \int_0^h N_i N_j N_k' dx \qquad (4.3.8)$$

$i , j , k = 1, \ldots, 4 .$

The matrices A^e and C^e are square 4×4 and L^e is a block $4 \times 4 \times 4$ having an associated 4×4 square matrix defined by:

$$B_{ij} = \sum_{k=1}^{4} L_{ijk} \delta_k^e \qquad (4.3.9)$$

where $\underset{\sim}{\delta}^e = [\, u_i \ , \ u_i' \ , \ u_{i+1} \ , \ u_{i+1}' \,]^T$

The element matrices A^e , C^e , L^e have been computed algebraically from equations (4.3.6) - (4.3.8) using the computer Algebra package REDUCE [38] as:

$$A^e = \begin{bmatrix} \frac{13h}{35} & \frac{11h^2}{210} & \frac{9h}{70} & -\frac{13h^2}{420} \\ \frac{11h^2}{210} & \frac{h^3}{105} & \frac{13h^2}{420} & -\frac{h^3}{140} \\ \frac{9h}{70} & \frac{13h^2}{420} & \frac{13h}{35} & -\frac{11h^2}{210} \\ -\frac{13h^2}{420} & -\frac{h^3}{140} & -\frac{11h^2}{210} & \frac{h^3}{105} \end{bmatrix} \qquad (4.3.10)$$

$$C^e = \begin{bmatrix} 0 & \frac{1}{h} & 0 & -\frac{1}{h} \\ -\frac{1}{h} & -\frac{1}{2} & \frac{1}{h} & -\frac{1}{2} \\ 0 & -\frac{1}{h} & 0 & \frac{1}{h} \\ \frac{1}{h} & \frac{1}{2} & -\frac{1}{h} & \frac{1}{2} \end{bmatrix} \qquad (4.3.11)$$

and

$$B^e_{11} = \frac{1}{210}\,(-70\ ,\ 25h\ ,\ 70\ ,\ -17h\ )\ \underset{\sim}{\delta}^e$$

$$B^e_{12} = B^e_{21} = \frac{h}{840}\,(-50\ ,\ 5h\ ,\ 50\ ,\ -11h)\ \underset{\sim}{\delta}^e$$

$$B^e_{13} = B^e_{31} = \frac{1}{210}\,(-35\ ,\ -4h\ ,\ 35\ ,-4h\ )\ \underset{\sim}{\delta}^e$$

$$B^e_{14} = B^e_{41} = \frac{h}{840}\,(\ 34\ ,\ 3h\ ,\ -34\ ,\ 5h)\ \underset{\sim}{\delta}^e$$

$$B^e_{22} = \frac{h^2}{420}\,(\ -5\ ,\ 0\ ,\ 5\ ,\ -h)\ \underset{\sim}{\delta}^e$$

$$B^e_{23} = B^e_{32} = \frac{h}{840}\,(\ -34\ ,\ -5h\ ,\ 34\ ,-3h)\ \underset{\sim}{\delta}^e$$

$$B^e_{24} = B^e_{42} = \frac{h^2}{840}\,(\ 8\ ,\ h\ ,\ -8\ ,h\ )\ \underset{\sim}{\delta}^e$$

$$B_{33}^{e} = \frac{1}{210}\,(\,-70\,,\,-17h\,,\,70\,,25h\,)\,\underset{\sim}{\delta}^{e}$$

$$B_{34}^{e} = B_{43}^{e} = \frac{h}{840}\,(\,50\,,\,11h\,,\,-50\,,\,-5h)\,\underset{\sim}{\delta}^{e}$$

$$B_{44}^{e} = \frac{h^{2}}{420}\,(\,-5\,,\,-h\,,\,5\,,\,0\,)\,\underset{\sim}{\delta}^{e} \qquad (4.3.12)$$

where

$$\underset{\sim}{\delta}^{e} = [\,u_{i}\,,\,u_{i}'\,,\,u_{i+1}\,,\,u_{i+1}'\,]^{T} \qquad (4.3.13)$$

Suppose that the solution region [a,b] is divided into N elements of equal length h. Each element consists of two nodes and each node has two unknown nodal parameters (one for the variable and the other for its derivative). So each element has four unknowns. Since these elements are connected at nodes $1,2,\ldots,N-1$, so for the first element there are four unknowns $\delta_1^{(1)},\delta_2^{(1)},\delta_3^{(1)},\delta_4^{(1)}$ and for the second element the unknowns are $\delta_1^{(2)},\delta_2^{(2)},\delta_3^{(2)},\delta_4^{(2)}$ and so on until the N-th element with unknowns $\delta_1^{(N)},\delta_2^{(N)},\delta_3^{(N)},\delta_4^{(N)}$ where the superscript denotes the element number.

From the continuity of **u** and **u'** we get the following:

$$\delta_1^{(2)}=\delta_3^{(1)},\ \delta_2^{(2)}=\delta_4^{(1)},\ldots,\delta_1^{(N)}=\delta_3^{(N-1)},\delta_2^{(N)}=\delta_4^{(N-1)} \qquad (4.3.14)$$

Now we are going to identify the local nodal parameters with global nodal parameters δ_i , $i=1,2,\ldots,2N+2$ which can be written as:

$$\delta_1^{(1)}=\delta_1,\ \delta_2^{(1)}=\delta_2,\ \delta_3^{(1)}=\delta_5,\ \delta_4^{(1)}=\delta_4,\ldots,\delta_1^{(N)}=\delta_{2N-1}\ ,$$

$$\delta_2^{(N)}=\delta_{2N}\ ,\ \delta_3^{(N)}=\delta_{2N+1}\ ,\ \delta_4^{(N)}=\delta_{2N+2} \qquad (4.3.15)$$

Let us divide the region [a,b] into three elements of equal length. Then assembling the matrices A , B , and C gives:

$$A = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} & & & & \\ a_{21}^{(1)} & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & & & & \\ a_{31}^{(1)} & a_{32}^{(1)} & a_{11}^{*(2)} & a_{12}^{*(2)} & a_{13}^{(2)} & a_{14}^{(2)} & & \\ a_{41}^{(1)} & a_{42}^{(1)} & a_{21}^{*(2)} & a_{22}^{*(2)} & a_{23}^{(2)} & a_{24}^{(2)} & & \\ & & a_{31}^{(2)} & a_{32}^{(2)} & a_{11}^{*(3)} & a_{12}^{*(3)} & a_{13}^{(3)} & a_{14}^{(3)} \\ & & a_{41}^{(2)} & a_{42}^{(2)} & a_{21}^{*(3)} & a_{22}^{*(3)} & a_{23}^{(3)} & a_{24}^{(3)} \\ & & & & a_{31}^{(3)} & a_{32}^{(3)} & a_{33}^{(3)} & a_{34}^{(3)} \\ & & & & a_{41}^{(3)} & a_{42}^{(3)} & a_{43}^{(3)} & a_{44}^{(3)} \end{bmatrix} \quad (4.3.16)$$

$$B = \begin{bmatrix} b_{11}^{(1)} & b_{12}^{(1)} & b_{13}^{(1)} & b_{14}^{(1)} & & & & \\ b_{21}^{(1)} & b_{22}^{(1)} & b_{23}^{(1)} & b_{24}^{(1)} & & & & \\ b_{31}^{(1)} & b_{32}^{(1)} & b_{11}^{*(2)} & b_{12}^{*(2)} & b_{13}^{(2)} & b_{14}^{(2)} & & \\ b_{41}^{(1)} & b_{42}^{(1)} & b_{21}^{*(2)} & b_{22}^{*(2)} & b_{23}^{(2)} & b_{24}^{(2)} & & \\ & & b_{31}^{(2)} & b_{32}^{(2)} & b_{11}^{*(3)} & b_{12}^{*(3)} & b_{13}^{(3)} & b_{14}^{(3)} \\ & & b_{41}^{(2)} & b_{42}^{(2)} & b_{21}^{*(3)} & b_{22}^{*(3)} & b_{23}^{(3)} & b_{24}^{(3)} \\ & & & & b_{31}^{(3)} & b_{32}^{(3)} & b_{33}^{(3)} & b_{34}^{(3)} \\ & & & & b_{41}^{(3)} & b_{42}^{(3)} & b_{43}^{(3)} & b_{44}^{(3)} \end{bmatrix} \quad (4.3.17)$$

$$C = \begin{bmatrix} c_{11}^{(1)} & c_{12}^{(1)} & c_{13}^{(1)} & c_{14}^{(1)} & & & & \\ c_{21}^{(1)} & c_{22}^{(1)} & c_{23}^{(1)} & c_{24}^{(1)} & & & & \\ c_{31}^{(1)} & c_{32}^{(1)} & c_{11}^{*(2)} & c_{12}^{*(2)} & c_{13}^{(2)} & c_{14}^{(2)} & & \\ c_{41}^{(1)} & c_{42}^{(1)} & c_{21}^{*(2)} & c_{22}^{*(2)} & c_{23}^{(2)} & c_{24}^{(2)} & & \\ & & c_{31}^{(2)} & c_{32}^{(2)} & c_{11}^{*(3)} & c_{12}^{*(3)} & c_{13}^{(3)} & c_{14}^{(3)} \\ & & c_{41}^{(2)} & c_{42}^{(2)} & c_{21}^{*(3)} & c_{22}^{*(3)} & c_{23}^{(3)} & c_{24}^{(3)} \\ & & & & c_{31}^{(3)} & c_{32}^{(3)} & c_{33}^{(3)} & c_{34}^{(3)} \\ & & & & c_{41}^{(3)} & c_{42}^{(3)} & c_{43}^{(3)} & c_{44}^{(3)} \end{bmatrix} \quad (4.3.18)$$

where :

$$a_{11}^{*(2)} = a_{33}^{(1)} + a_{11}^{(2)} \quad , \quad a_{12}^{*(2)} = a_{34}^{(1)} + a_{12}^{(2)} \quad , \quad a_{21}^{*(2)} = a_{43}^{(1)} + a_{21}^{(2)}$$

$$a_{22}^{*(2)} = a_{44}^{(1)} + a_{22}^{(2)} \quad , \quad a_{11}^{*(3)} = a_{33}^{(2)} + a_{11}^{(3)} \quad , \quad a_{12}^{*(3)} = a_{34}^{(2)} + a_{12}^{(3)}$$

$$a_{21}^{*(3)} = a_{43}^{(2)} + a_{21}^{(3)} \quad , \quad a_{22}^{*(3)} = a_{44}^{(2)} + a_{22}^{(3)} \qquad (4.3.19)$$

$$b_{11}^{*(2)} = b_{33}^{(1)} + b_{11}^{(2)} \quad , \quad b_{12}^{*(2)} = b_{34}^{(1)} + b_{12}^{(2)} \quad , \quad b_{21}^{*(2)} = b_{43}^{(1)} + b_{21}^{(2)}$$

$$b_{22}^{*(2)} = b_{44}^{(1)} + b_{22}^{(2)} \quad , \quad b_{11}^{*(3)} = b_{33}^{(2)} + b_{11}^{(3)} \quad , \quad b_{12}^{*(3)} = b_{34}^{(2)} + b_{12}^{(3)}$$

$$b_{21}^{*(3)} = b_{43}^{(2)} + b_{21}^{(3)} \quad , \quad b_{22}^{*(3)} = b_{44}^{(2)} + b_{22}^{(3)} \qquad (4.3.20)$$

$$c_{11}^{*(2)} = c_{33}^{(1)} + c_{11}^{(2)} \quad , \quad c_{12}^{*(2)} = c_{34}^{(1)} + c_{12}^{(2)} \quad , \quad c_{21}^{*(2)} = c_{43}^{(1)} + c_{21}^{(2)}$$

$$c_{22}^{*(2)} = c_{44}^{(1)} + c_{22}^{(2)} \quad , \quad c_{11}^{*(3)} = c_{33}^{(2)} + c_{11}^{(3)} \quad , \quad c_{12}^{*(3)} = c_{34}^{(2)} + c_{12}^{(3)}$$

$$c_{21}^{*(3)} = c_{43}^{(2)} + c_{21}^{(3)} \quad , \quad c_{22}^{*(3)} = c_{44}^{(2)} + c_{22}^{(3)} \qquad (4.3.21)$$

$$\underset{\sim}{\delta} = [\delta_1 \, , \delta_2 \, , \delta_3 \, , \delta_4 \, , \delta_5 \, , \delta_6 \, , \delta_7 \, , \delta_8]^T \qquad (4.3.22)$$

Similarly, assembling together contribution from N elements leads to the matrix equation:

$$A \, \dot{\underset{\sim}{\delta}} + \varepsilon \, B(\delta) \, \underset{\sim}{\delta} - \mu \, C \, \underset{\sim}{\delta} = 0 \qquad (4.3.23)$$

The matrices A , $B(\delta)$, and C are septa-diagonal in form. The element matrices A^e , C^e are the same for every finite element and remain constant throughout the run, so A and C are time independent. L^e_{ijk} is also the same for every element and remains fixed, but $B^e(\delta)$ depends upon nodal parameters δ^e which are time

dependent, so that $B(\underset{\sim}{\delta})$ must be reconstructed for every time step. Suppose that between time levels n and $n+1$ $\underset{\sim}{\delta}$ is interpolated by:

$$\underset{\sim}{\delta} = ((1-\theta)\ ,\ \theta) \begin{bmatrix} \underset{\sim}{\delta}^{n} \\ \underset{\sim}{\delta}^{n+1} \end{bmatrix} \qquad (4.3.24)$$

where $t=(n+\theta)\Delta t$ with $0 \leq \theta \leq 1$. Then:

$$\frac{d\,\underset{\sim}{\delta}}{d\,t} = \frac{1}{\Delta t}\ (-1\ ,\ \ 1) \begin{bmatrix} \underset{\sim}{\delta}^{n} \\ \underset{\sim}{\delta}^{n+1} \end{bmatrix} \qquad (4.3.25)$$

Hence equation (4.3.23) can be written as:

$$A\left[\frac{\underset{\sim}{\delta}^{n+1} - \underset{\sim}{\delta}^{n}}{\Delta t}\right] + \varepsilon\ B(\underset{\sim}{\delta}^{n})\left[(1-\theta)\underset{\sim}{\delta}^{n} + \theta\underset{\sim}{\delta}^{n+1}\right] - \mu\ C\left[(1-\theta)\underset{\sim}{\delta}^{n} + \theta\underset{\sim}{\delta}^{n+1}\right] = 0$$

Rearranging we obtain the recurrence relationship:

$$\left[A + \theta\ \Delta t(\varepsilon\ B(\underset{\sim}{\delta}^{n}) - \mu\ C)\right]\underset{\sim}{\delta}^{n+1} = \left[A + (1-\theta)\Delta t(-\varepsilon\ B(\underset{\sim}{\delta}^{n}) + \mu\ C)\right]\underset{\sim}{\delta}^{n} \qquad (4.3.26)$$

The parameter θ takes different values such that:

$\theta = 0$	gives	Forward Difference scheme
$\theta = \frac{1}{2}$	gives	Crank-Nicolson scheme
$\theta = 1$	gives	Backward Difference scheme

Now letting $\theta = \frac{1}{2}$, equation (4.3.26) becomes:

$$\left[A + \frac{\Delta t}{2}\ (\varepsilon\ B(\underset{\sim}{\delta}^{n}) - \mu\ C)\right]\underset{\sim}{\delta}^{n+1} = \left[A - \frac{\Delta t}{2}\ (\varepsilon\ B(\underset{\sim}{\delta}^{n}) - \mu\ C)\right]\underset{\sim}{\delta}^{n} \qquad (4.3.27)$$

which is a recurrence relationship for updating the nodal parameters from time level n to time level $n+1$ which we shall use exclusively in the following two equations.

The matrix $B(\underset{\sim}{\delta})$ is nonlinear so that the system (4.3.27) is also nonlinear. Our approach to the solution is to replace the system (4.3.27) by two equations [56,70]:

$$\left[A + \frac{\Delta t}{2}(\varepsilon\, B(\underset{\sim}{\delta}^{n}) - \mu\, C)\right]\underset{\sim}{\hat{\delta}}^{n+1} = \left[A - \frac{\Delta t}{2}(\varepsilon\, B(\underset{\sim}{\delta}^{n}) - \mu C)\right]\underset{\sim}{\delta}^{n} \quad (4.3.28a)$$

and

$$\left[A + \frac{\Delta t}{2}(\varepsilon\, B\left(\frac{\underset{\sim}{\hat{\delta}}^{n+1} + \underset{\sim}{\delta}^{n}}{2}\right) - \mu\, C)\right]\underset{\sim}{\delta}^{n+1} = \left[A - \frac{\Delta t}{2}(\varepsilon\, B\left(\frac{\underset{\sim}{\hat{\delta}}^{n+1} + \underset{\sim}{\delta}^{n}}{2}\right) - \mu\, C)\right]\underset{\sim}{\delta}^{n} \quad (4.3.28b)$$

The predictor (4.3.28a) gives a first approximation $\underset{\sim}{\hat{\delta}}^{n+1}$ then the corrector (4.3.28b) may be used iteratively to improve the approximation.

The solution of the system (4.3.28) we obtained will be influenced by the boundary and initial conditions. So, firstly we apply the boundary conditions $u_0 = \beta$, $u_N = 0$, $u'_0 = 0$, $u'_N = 0$.

The initial conditions on $u(x,0)$ and $u'(x,0)$ determine the starting nodal vector $\underset{\sim}{\delta}^{0}$.

The system (4.3.28) consists of two systems of $2(N+1)$ equations of $2(N+1)$ unknowns. One can solve the 7-banded systems (4.3.28) using Gaussian elimination .However, this algorithm is an uneconomic method to apply because the matrices A , $B(\underset{\sim}{\delta})$, and C contain a large number of zero elements. One can avoid these difficulties by storing these matrices in rectangular form of order $2(N+1)\times 7$ and then use the septa-diagonal algorithm based on the Thomas algorithm for tridiagonal matrices(see Appendix A3).

4.4 The Test Problems:

We first study the motion of a single soliton. This is derived from the initial condition:

(a) $$u(x,0) = 3\,c\,\mathrm{sech}^2(A_1 x + D_1) \qquad (4.4.1)$$

This follows from the analytic solution of the KdV equation which has been discussed in chapter 2 and it has the form:

$$u(x,t) = 3\,c\,\mathrm{sech}^2(A_1 x - B_1 t + D_1) \qquad (4.4.2)$$

where; $A_1 = \frac{1}{2}(\varepsilon\, c/\mu)^{1/2}$ and $B_1 = \varepsilon\, cA_1$. To permit comparison with Greig and Morris [26], we choose $\varepsilon = 1$, $\mu = 4.84\times10^{-4}$, $c = 0.3$, $D_1 = -6.$, $h = 0.05$, 0.033 , 0.01 , and $\Delta t = 0.025$, 0.01 , 0.005 We shall impose the boundary conditions:

$$\left.\begin{aligned} u(0,t) &= u(2,t) = 0 \\ u_x(0,t) &= u_x(2,t) = 0 \end{aligned}\right\} \quad \text{for all time} \qquad (4.4.3)$$

We show in Figure 4.2 our solution for times from $t = 0.0$ to $t = 3.0$. These graphs compare exactly with those of Greig and Morris [26] for corresponding times and if the exact solution is plotted on the same figure the curves are indistinguishable:

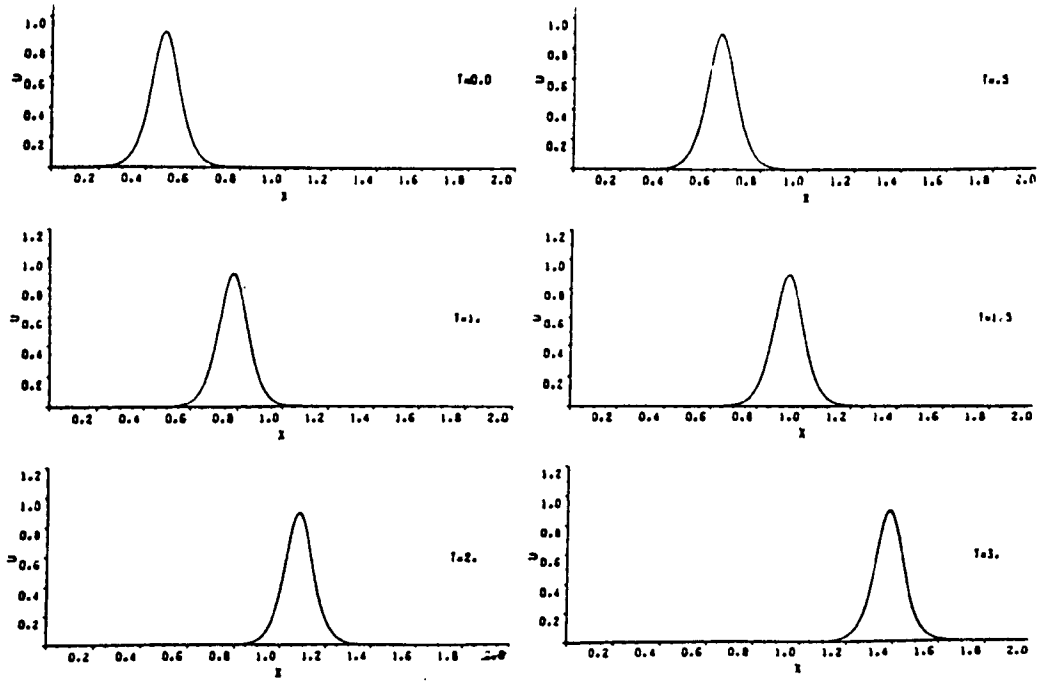

Figure 4.2 Problem (a). The motion of a single soliton with $\Delta t = 0.005$, $h = 0.01$.

Our second example concerns the interaction between solitons. We use the initial condition:

$$\text{(b)} \quad u(x,0) = 3\,c_1 \operatorname{sech}^2(A_1 x + D_1) + 3\,c_2 \operatorname{sech}^2(A_2 x + D_2) \qquad (4.4.4)$$

Figure 4.3 shows the two solitons with larger on the left. As the time increases, the larger soliton catches up with the smaller until, at time t = 0.75, the smaller soliton is in the process of being absorbed, having lost its solitary wave identity. The overlapping process continues until, by time t = 1.5, the larger soliton has overtaken the smaller one and is in the process of separating. At time t = 3, the interaction is complete and the larger soliton has separated completely from the smaller one:

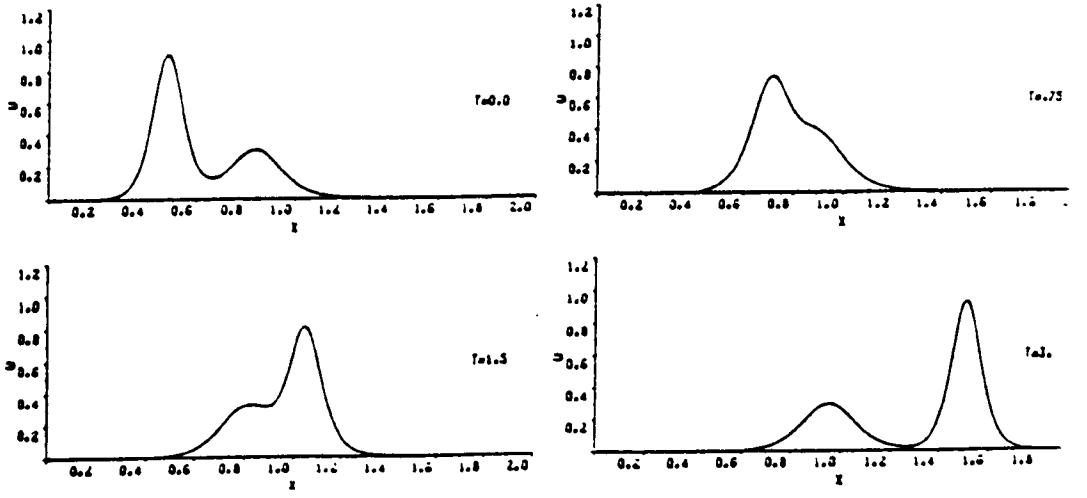

Figure 4.3 Problem (b). The interaction of two overlapping solitons with $\Delta t = 0.005$, $h = 0.01$.

This solution represents two solitons of magnitudes c_1 and c_2 sited initially at $x = -D_1/A_1$ and $-D_2/A_2$ respectively with $c_1 = 0.3$, $c_2 = 0.1$, and $D_1 = D_2 = -6.$, $A_j = \frac{1}{2}\left[\frac{\varepsilon c_j}{\mu}\right]^{1/2}$, $j = 1,2$.

Chosing $c_1 > c_2$ ensures that the velocity, and magnitude of

the soliton at $x = -D_1/A_1$ is the larger and that the solitons interact with increasing time. We take as boundary conditions:

$$\left.\begin{aligned} u(0,t) &= u(2,t) = 0 \\ u_x(0,t) &= u_x(2,t) = 0 \end{aligned}\right\} \text{ for all time} \qquad (4.4.5)$$

Our simulations (Figure 4.3) show that after the interaction of two overlapping solitons the large and small amplitudes of the solitons are altered from the original by about $\approx 1\%$, $\approx 0.39\%$ respectively as was also observed by Greig and Morris [26], the agreement between the solutions is satisfactory.

(b1) Consider the initial condition:

$$u(x,0) = 3\, c_1 \operatorname{sech}^2(A_1 x + D_1) + 3\, c_2 \operatorname{sech}^2(A_2 x + D_2) \qquad (4.4.6)$$

where:

$$A_i = \frac{1}{2}\left[\frac{\varepsilon\, c_i}{\mu}\right]^{1/2} \qquad i = 1,\ 2 \qquad , \qquad B_i = \frac{1}{2}\,\varepsilon\, A_i$$

$$\mu = 0.000484 \ , \ c_1 = 0.3 \ , \ c_2 = 0.1 \ , \ D_1 = -6. \ , \ D_2 = -9.$$

The boundary conditions are imposed:

$$\left.\begin{aligned} u(0,t) &= u(4,t) = 0 \\ u_x(0,t) &= u_x(4,t) = 0 \end{aligned}\right\} \text{ for all time} \qquad (4.4.7)$$

The reason for chosing this initial condition is to produce an initial condition in which the two solitons are well separated.

Figure 4.4 shows the two solitons with larger on the left. As the time increases, the larger soliton catches up with the smaller until, at time t = 3. The overlapping process continues until, by time t = 4, the larger soliton has overtaken the smaller one and is in the process of separating. At time t = 6, the interaction is complete and the larger soliton has separated completely from the smaller one:

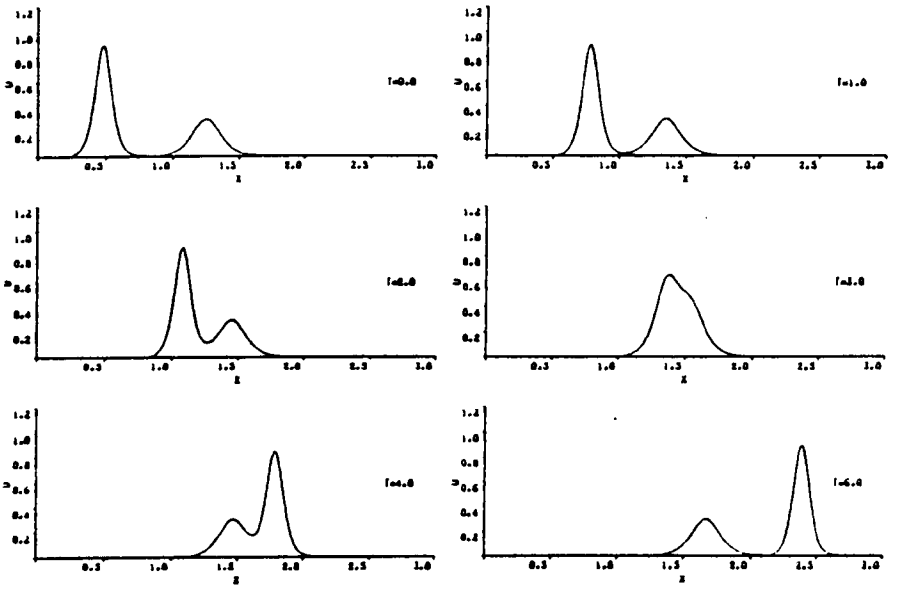

Figure 4.4 Problem (b1). The interaction of two well separated solitons for $\Delta t = 0.005$, $h = 0.01$.

In this case, after the interaction the two solitons emerge virtually unchanged in form and amplitude, the faster soliton now being ahead. This point will be considered in detail in the discussion of section 4.5.

Our third test example has an inital condition:

$$(c) \qquad u(x,0) = \exp(-x^2) \tag{4.4.8}$$

The boundary conditions we impose are:

$$\left.\begin{aligned} u(\mp 15, t) &= 0 \\ u_x(\mp 15, t) &= 0 \end{aligned}\right\} \quad \text{for all } t > 0 \tag{4.4.9}$$

We choose $\varepsilon = 1.0$ and each of:

(c_1) $\mu = 0.029$ $h = 0.1$, $\Delta t = 0.01$

(c_2) $\mu = 0.01$, $h = 0.1$, $\Delta t = 0.01$

(c_3) $\mu = 0.0076$, $h = 0.1$, $\Delta t = 0.01$

(c_4) $\mu = 0.0037$, $h = 0.1$, $\Delta t = 0.01$

successively, so that comparison with the work of Goda [59] is possible for problem (c_2).

The balance between the dispersion and the nonlinearity occurs when $\mu_c \approx 0.0625$ [41,80] and from Figure 4.5 we see that the initial condition with this value of $\mu = \mu_c$ produces a pure (single) soliton. It was found that the behaviour of the numerical solutions was completely different according to whether $\mu \gg \mu_c$ or $\mu \ll \mu_c$. The initial condition breaks up into two solitons when $0.02 < \mu < 0.0625$, into three when $0.0083 < \mu < 0.02$, into four when $\mu \approx 0.0083$, into six when $\mu \approx 0.0039$, and thereafter the number of emergent solitons increases indefinitely with decreasing numerical values of μ. On the other hand, if $\mu \gg \mu_c$, no solution breaks up into solitons at all, but the solutions for $\mu \gg \mu_c$ exhibit rapidly oscillating wave packets. For certain intermediate values of μ, a mixed type of solution was found which consists of a leading soliton and an oscillating tail [15,39,80]. Our numerical results will be compared with their theoretical predictions:

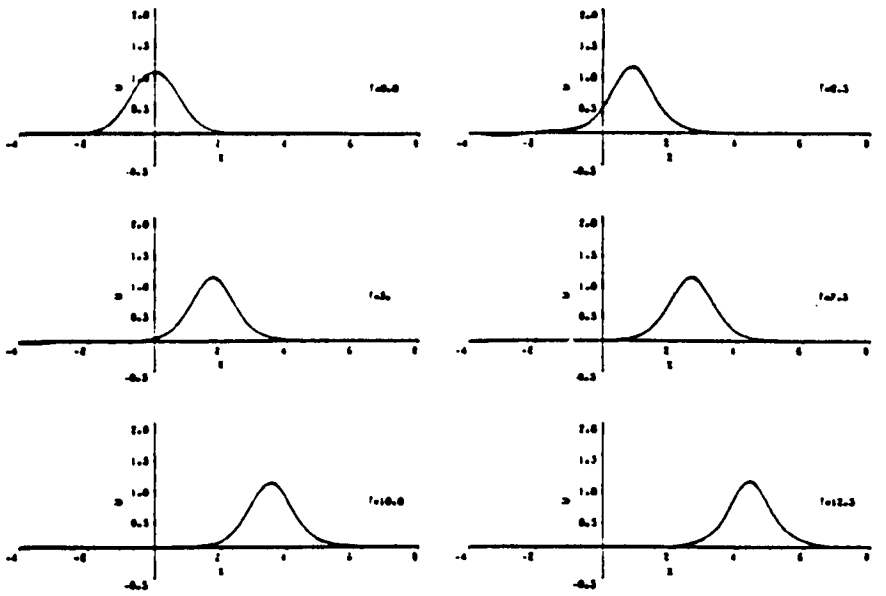

Figure 4.5 The motion of a single soliton with balance initial condition when $\mu = 0.0625$ $h = 0.1$ $\Delta t = 0.01$.

Figure 4.6 shows with $\mu = 0.029$ the numerical solution for times up to $t = 12.5$. We see that the initial condition resolves itself

into two solitons plus an oscillatory tail in agreement with the theoretical results:

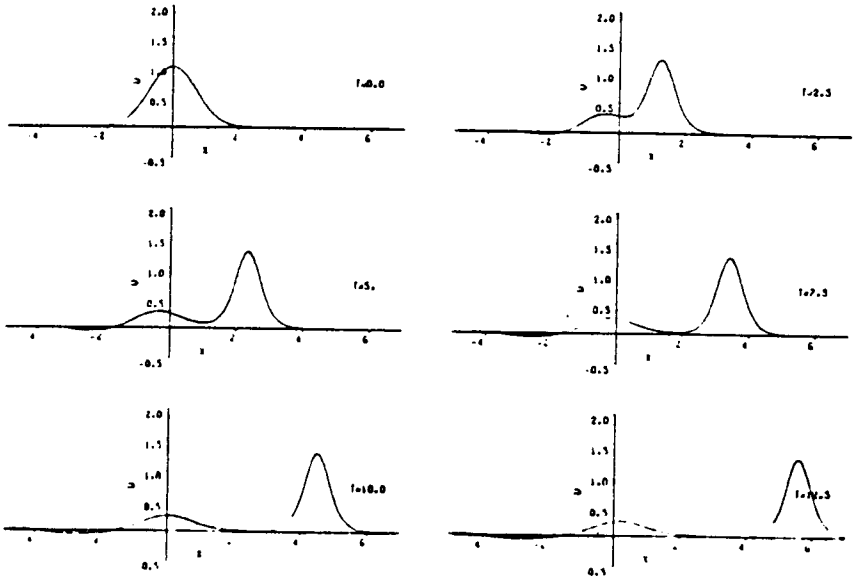

Figure 4.6 Problem (c1). The breakdown of the initial condition into two solitons with $\mu = 0.029$ $h = 0.1$ $\Delta t = 0.01$.

Figure 4.7 shows similar results for $\mu = 0.01$. Now the initial perturbation breaks up into three solitons. The graph obtained in this case is identical with that given by Goda [59], and again agrees with theoretical predictions:

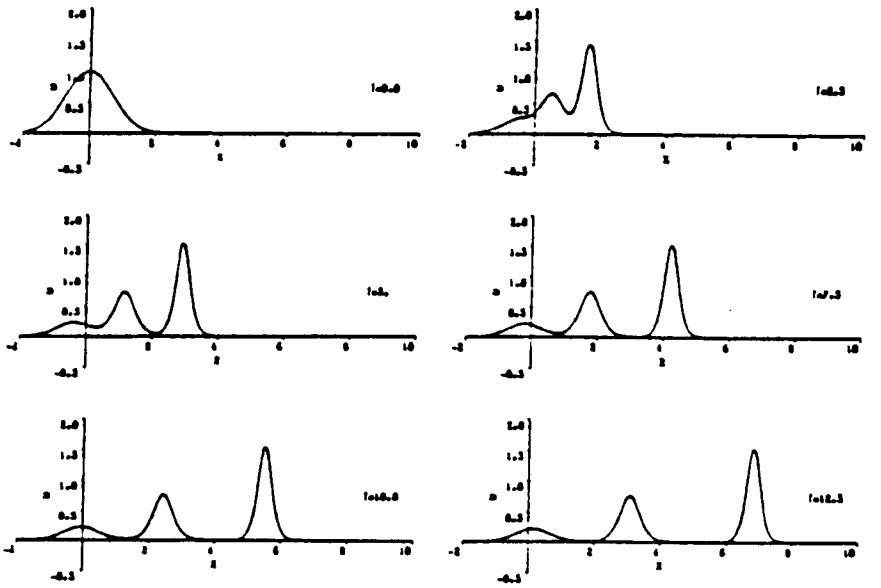

Figure 4.7 Problem (c2). The breakdown of the initial condition into 3 solitons when $\mu = 0.01$, $h = 0.1$, $\Delta t = 0.01$.

In Figure 4.8, we have plotted soliton profiles obtained from

equation (4.4.1) using appropriate values of μ and wave amplitude. If these profiles are superimposed on Figure 4.7, we can confirm, to within plotting error, that the solitary waves we have obtained are in fact solitons:

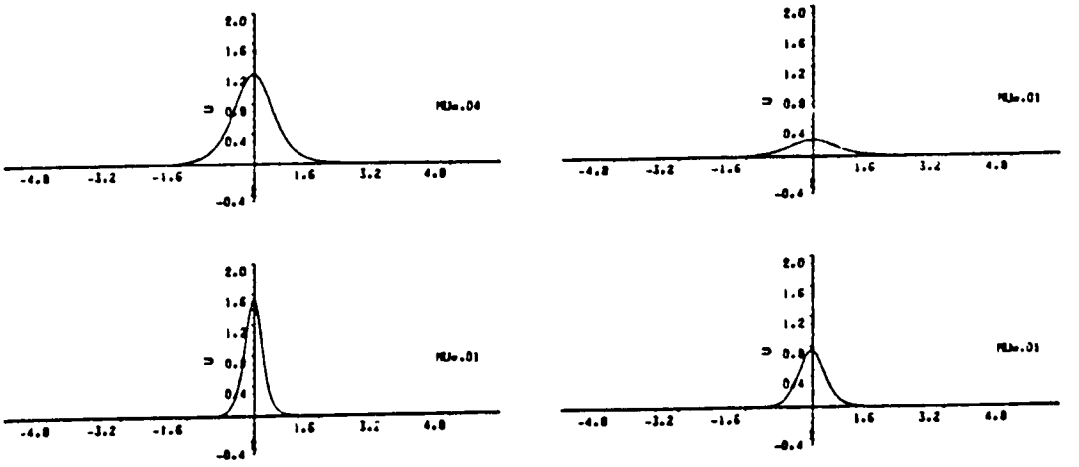

Figure 4.8 Solitons of various amplitudes with $\mu = 0.01$, $\varepsilon = 1.0$.

In Figure 4.9, we see that when the coefficient of the dispersive term is decreased to $\mu = 0.0076$ the nonlinear term dominates, hence the amplitude increases with time and we find that the initial perturbation breaks up into 3 solitons, with the amplitude increasing linearly:

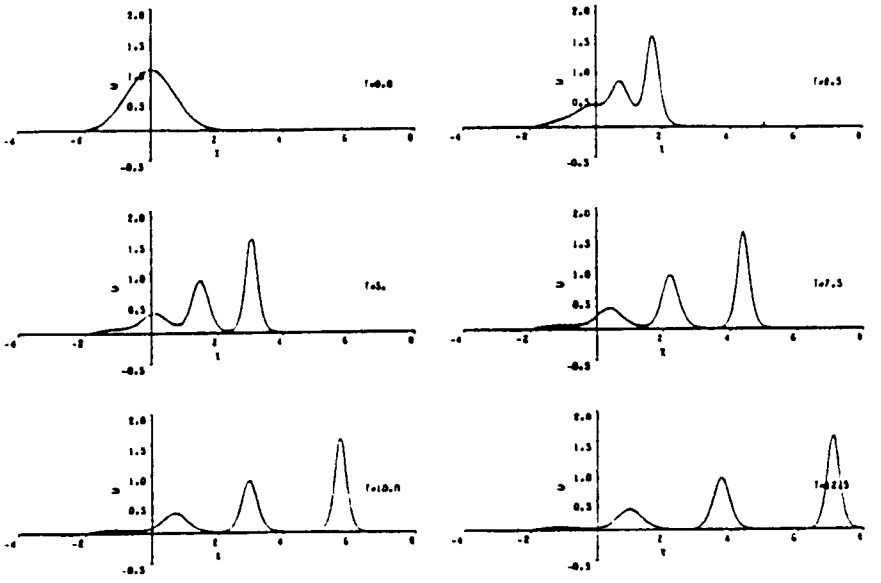

Figure 4.9 Problem (c3). The breakdown of the initial perturbation into 3 solitons with $\mu = 0.0076$, $h = 0.1$, $\Delta t = 0.01$.

Figure 4.10 shows that when the coefficient of the dispersive term is decreased to $\mu = 0.0037$, the initial perturbation breaks down into 5 solitons whose amplitude increases linearly:

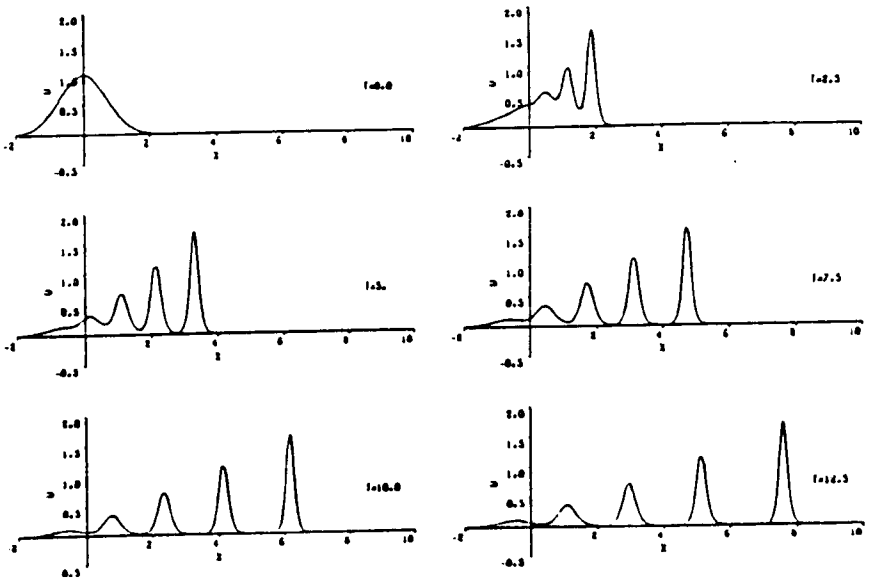

Figure 4.10 Problem (c4). The initial condition splits into 5 solitons when $\mu = 0.0037$, $h = 0.1$, $\Delta t = 0.01$.

A comparison of our results with those obtained by [15,39,80] has been made and we find that there is agreement when $\mu \geq 0.01$. For $\mu = 0.0076$, 0.0037, we obtained three and five solitons respectively which does not agree with their theoretical predictions [15,39,80].

As a final test example, we consider the development of an undular bore in shallow water. This is characterised by the initial condition:

$$\text{(d)} \qquad u(x,0) = \frac{1}{2}\left[1 - \tanh\left[\frac{x - 25}{5} \right] \right] \qquad (4.4.10)$$

with two boundary conditions:

$$\text{(i)} \qquad \left.\begin{aligned} u(0,t) &= 1 \\ u(50,t) &= 0 \\ u_x(0,t) &= u_x(50,t) = 0 \end{aligned}\right\} \text{ for all } t > 0 \qquad (4.4.11)$$

$$\text{(ii)} \qquad \left.\begin{aligned} u(-50,t) &= u(150,t) = 0 \\ u_x(-50,t) &= u_x(150,t) = 0 \end{aligned}\right\} \text{ for all } t > 0 \qquad (4.4.12)$$

There is an earlier numerical solution to this problem by Vliegenthart [44] who gives a graphical solution. We intended to compare our approach with this solution by solving the same problem, choosing $\varepsilon = 0.2$, $\mu = 0.1$ and $\Delta t = 0.05$, $h = 0.4$. Unfortunately the boundary conditions used were not stated explicitly. Soon a study of the graphs presented we concluded that these were most likely to be

$$u(0,t) = 1 \quad , \quad u(50,t) = 0$$

$$u_x(0,t) = u_x(50,t) = 0.$$

We adopted these boundary conditions and undertook two simulations, the first using the method outlined here and the second using Vliegenthart's finite difference scheme. The results were identical, but did not agree with the published figures. We determined the velocity of the leading soliton from the present simulations and from the published graphs. We found that our simulations produced a soliton moving slowly than the theoretical result, whereas Vliegenthart soliton was moving more rapidly. We finally decided to use the initial condition given in Figure 4.13 with boundary conditions (4.4.12). As expected the initial perturbation has degenerated into a train of solitons, which move steadily to the right with constant amplitude and velocity. We see that the amplitudes of the solitons vary approximately linearly. We find that the velocity of the leading soliton is in complete agreement with the theoretical value determined from its amplitude:

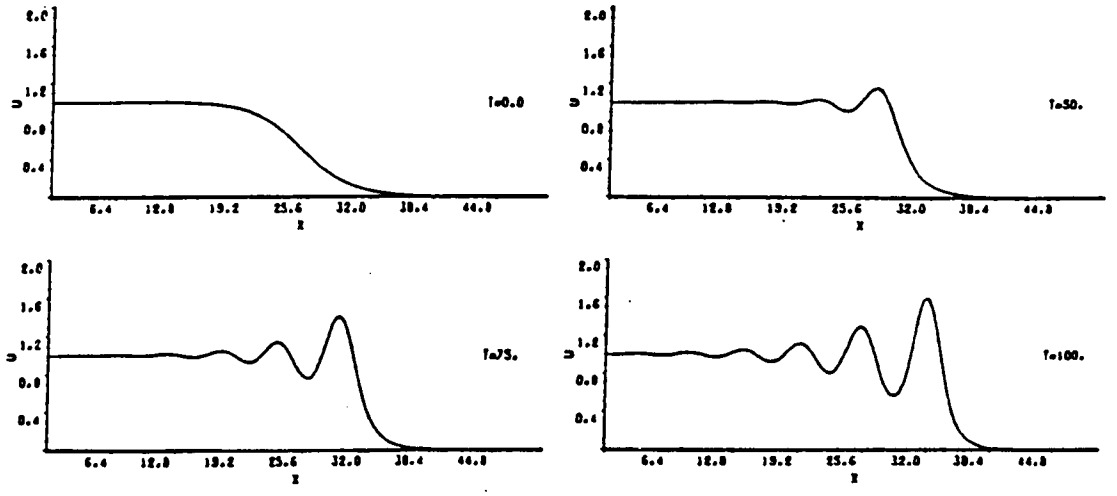

Figure 4.11 Problem (d) boundary conditions (i).The solution graphs for various times with $\Delta t = 0.05$, $h = 0.4$, $\varepsilon = 0.2$, $\mu = 0.1$.

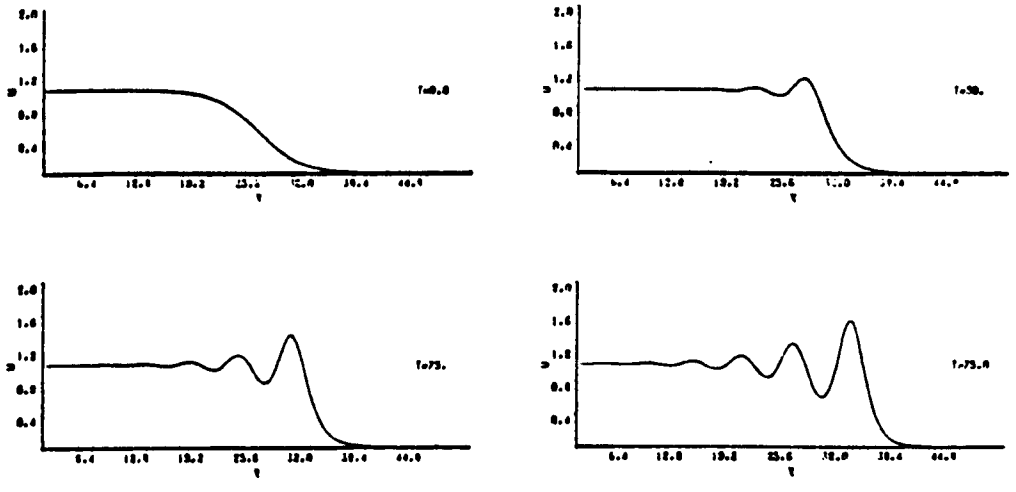

Figure 4.12 Problem (d) boundary conditions (i). The solution graphs for various times with $\Delta t = 0.05$, $h = 0.4$, $\varepsilon = 0.2$, $\mu = 0.1$ using finite difference scheme.

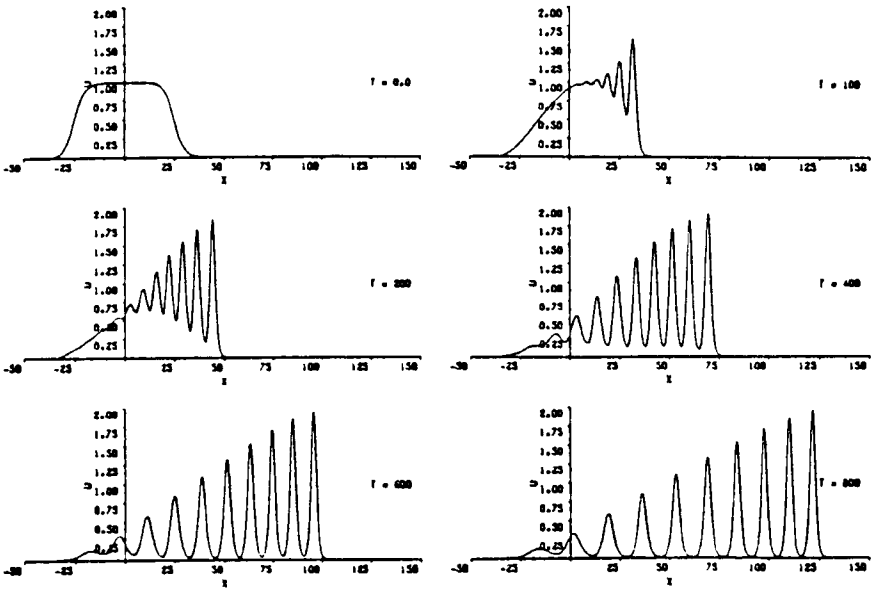

Figure 4.13 Problem (d) boundary conditions (ii). The initial perturbation breaks into 10 solitons $\Delta t=.05$, $h=.4$, $\epsilon=.2$, $\mu=.1$.

4.5 Discussion:

When used to determine the solution of the KdV equation any numerical scheme must be capable of faithfully representing the amplitude of the solution over many time steps and must also predict the progress of the wave front with little error [26].

We have observed graphically that our numerical single soliton solution for problem (a) is indistinguishable from the analytic results.

To measure the accuracy of the numerical methods in solving the KdV equation we compute the difference between the analytic and numerical solutions at each mesh point after each specified time step, and use this in the discrete L_2- and L_∞- error norms which are defined by [29,36,59]:

$$L_2 = \| u^{exact} - u^n \|_2 = \left[h \sum_{j=1}^{N} | u_j^{exact} - u_j^n |^2 \right]^{1/2} \qquad (4.5.1)$$

and

$$L_\infty = \max_j | u_j^{exact} - u_j^n | \qquad (4.5.2)$$

This error is used to compare 5 numerical methods in Tables 4.1, 4.2 for the single soliton solution [36]. We see that the Galerkin cubic Hermite method compares very favourably with the methods of references [27] and [26] and is a competitor to the Petrov-Galerkin method [36]:

Table 4.1

The growth of the discrete L_2-error norm x 10^3 for single soliton

Time	Zabusky-Kruskal [27]	Hopscotch [26]	Petrov-Galerkin [36]	Modified P-G [36]	Galerkin Hermite
$\Delta x = 0.05$		$\Delta t = 0.025$	$h = 0.05$	$\Delta t = 0.025$	
0.25	34.64	61.21	81.39	52.15	1.20
0.50	122.68	122.41	102.54	64.90	2.45
0.75	210.44	181.35	125.84	89.01	4.00
1.00	298.19	228.10	150.57	107.20	5.79
			$h = 0.033$	$\Delta t = 0.01$	
0.25			31.18	5.94	0.19
0.5			43.35	7.56	0.33
0.75			56.21	8.70	0.47
1.00			74.08	9.49	0.62
$\Delta x = 0.01$		$\Delta t = 0.0005$	$h = 0.01$	$\Delta t = 0.005$	
0.25	5.94	3.79	4.46	0.21	0.03
0.50	13.17	9.28	7.01	0.38	0.05
0.75	21.08	14.14	10.08	0.57	0.07
1.00	28.66	18.72	13.26	0.74	0.09

From Table 4.1 we see that the L_2-error norm calculated from our scheme is smaller than that calculated by other authors and when compared with the best method quoted in Table 4.1 it is still smaller by a factor of at least 10:

Table 4.2

The growth of the discrete L_∞-error norm x 10^3 for single soliton

Time	Zabusky-Kruskal [27]	Hopscotch [26]	Petrov-Galerkin [36]	Modified P-G [36]	Galerkin Hermite
$\Delta x = 0.05$		$\Delta t = 0.025$	$h = 0.05$	$\Delta t = 0.025$	
0.25	19.4	32.7	42.18	30.22	3.05
0.50	63.5	67.4	51.85	22.85	5.78
0.75	122.4	99.3	87.60	35.86	11.31
1.00	161.4	141.6	100.41	39.39	14.61
			$h = 0.033$	$\Delta t = 0.01$	
0.25			14.27	2.80	0.43
0.5			21.65	4.53	0.81
0.75			29.78	4.85	1.15
1.00			39.37	5.85	1.66
$\Delta x = 0.01$		$\Delta t = 0.0005$	$h = 0.01$	$\Delta t = 0.005$	
0.25	2.05	1.11	1.21	0.07	0.06
0.50	4.22	2.14	2.15	0.11	0.13
0.75	6.36	3.54	3.09	0.17	0.18
1.00	8.13	4.91	3.83	0.21	0.23

Table 4.2 shows us that the L_∞-error norm computed from our technique for a single soliton using the definition (4.5.2) has been compared very well with all the method which are quoted in Table 4.2. We observe that the L_∞-error norm is greater than L_2-error norm which disagrrees with the authors [26,27,36]. Also we find that the value of L_∞-error norm using our method is smaller than even the best method (Modified Petrov-Galerkin) by factor 0.3 and in the worst case it has the same magnitude.

Table 4.3 shows us that the error is still small by

comparison with the other authors even when the time is increased up to t = 3.0. So we conclude that our numerical scheme is eminently suitable for the determination of solutions to the KdV equation even when relatively large time and space steps are used:

Table 4.3

The growth of the error for a single soliton

Time	h = 0.05 Δt =0.025		h = 0.033 Δt =0.01		h = 0.01 Δt =0.005	
	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$
1.25	7.75	21.65	0.80	2.09	0.11	0.28
1.50	10.03	24.27	0.98	2.45	0.12	0.32
1.75	12.51	35.49	1.16	2.99	0.14	0.38
2.00	15.30	37.82	1.33	3.61	0.16	0.41
2.25	18.25	51.35	1.52	4.07	0.18	0.47
2.50	21.63	53.26	1.71	4.38	0.20	0.54
2.75	24.81	68.56	1.89	4.86	0.22	0.59
3.00	28.63	69.54	2.12	5.80	0.24	0.64

It is important that any scheme used to solve the KdV equation be conservative. To examine this property for our scheme we have determined the quantities I_1, I_2 and I_3 defined by equations (2.4.8)-(2.4.10) respectively for problem (a) at various times; see Table 4.4:

Table 4.4

The computed value I_1, I_2, and I_3 for a single soliton

Time	I_1		I_2		I_3	
	h = .033 Δt = .01	h = .01 Δt =.005	h = .033 Δt = .01	h = .01 Δt =.005	h = .033 Δt = .01	h = .01 Δt=.005
0.0	0.144597	0.144598	0.086759	0.086759	0.046850	0.046850
0.50	0.144588	0.144598	0.086743	0.086760	0.046745	0.046832
1.00	0.144603	0.144599	0.086735	0.086760	0.046611	0.046821
1.50	0.144604	0.144600	0.086730	0.086761	0.046637	0.046820
2.00	0.144603	0.144601	0.086724	0.086762	0.046642	0.046822
2.50	0.144608	0.144602	0.086716	0.086763	0.046595	0.046826
3.00	0.144604	0.144604	0.086705	0.086764	0.046488	0.046831

From Table 4.4 we find that the quantities I_i change by less than 0.008%, 0.063% , 0.773% respectively when h = 0.033 , Δt = 0.01, and 0.005%, 0.006%, 0.065% respectively when h = 0.01 , Δt = 0.005 during the computer run. Thus even when h = 0.033 and Δt = 0.01, they may be considered satisfactorily conserved.

A computer run on the single soliton solution with 200 nodes and 200 time steps took 31 secs of CPU time on a VAX 8650.

With example (b) we have demonstrated the capability of this algorithm to accurately represent the solution when solitons coalesce for a short period and then separate with their profiles unaltered and only their relative amplitudes and positions changed. In fact, from our study of this problem (b) we found that the large and small amplitudes have changed after the interaction by amounts of ≈ 1% and ≈ 0.39% respectively. This may be due to the overlapping of the two solitons at time t = 0. For this reason we chose problem (b1) in which the solitons were initially well separated and we found that after the interaction the two solitons emerged with their profiles unchanged and only their relative

positions changed. Their large and small amplitudes have been only slightly affected (≈ 0.004% ,0.033 respectively). We conclude that the initial condition (b) is not an appropriate one when the interaction of solitons is studied.

The quantities I_1, I_2 and I_3 have been computed for problem (b) and are listed in Table 4.5:

Table 4.5

The computed values I_1, I_2, and I_3 for a double soliton (b) with h = 0.01, Δt = 0.005

Time	I_1	I_2	I_3
0.0	0.228081	0.107062	0.053316
0.50	0.228122	0.107063	0.053307
1.00	0.227942	0.107064	0.053313
1.50	0.227734	0.107065	0.053311
2.00	0.227684	0.107066	0.053302
2.50	0.227725	0.107067	0.053291
3.00	0.227870	0.107068	0.053293

From Table 4.5 we see that the quantities I_i $(i = 1,2,3)$ change by less than 0.175% , 0.006% , and 0.047% respectively during the computer run and so may be considered invariant; this is especially true for I_2.

We have also computed the first three conservative quantities for problem (b1), these are given in Table 4.6:

Table 4.6

The computed values I_1, I_2, and I_3 for a double soliton (b1) with $h = 0.01$, $\Delta t = 0.005$

Time	I_1	I_2	I_3
0.0	0.228082	0.103466	0.049864
1.0	0.228084	0.103468	0.049835
2.0	0.228085	0.103470	0.049840
3.0	0.228088	0.103472	0.049864
4.0	0.228092	0.103474	0.049849
5.0	0.228094	0.103476	0.049844
6.0	0.228098	0.103478	0.049853
7.0	0.228101	0.103480	0.049855
8.0	0.228105	0.103482	0.049852

Table 4.6 shows us that the quantities I_i $(i = 1,2,3)$ change by less than 0.011% , 0.016% , 0.059% respectively and so can be considered conserved even over much longer periods than that used above.

(b2) In this problem we consider the motion of two solitons with the initial conditions determined from the analytic solutions (2.3.4.1) with $t = 0.0$ in the following cases:

(i) problem (b) where $\alpha_i = \sqrt{c_i/\mu}$, $d_1 = -12$, $d_2 = -12 + \Delta$

(ii) problem **(b1)** where $\alpha_i = \sqrt{c_i/\mu}$, $d_1 = -12$, $d_2 = -18 + \Delta$

(iii) $\alpha_1 = 4.0$, $\alpha_2 = 2.0$, $d_1 = d_2 = 0.0$, $\varepsilon = 6$, $\mu = 1$.

The boundary conditions are chosen:

$$\left.\begin{aligned} u(\mp 12, t) &= 0 \\ u_x(\mp 12, t) &= 0 \end{aligned}\right\} \quad \text{for } -0.5 \le t \le 0.5 \qquad (4.5.3)$$

Before the interaction the position of the smaller amplitude is shifted forward by Δ_2. After the interaction the soliton with larger amplitude is shifted forward by Δ_1 and the soliton with

small amplitude is shifted backward by Δ_2.

The errors and the quantities I_i ($i = 1,\ldots,3$) have been computed for the problem (b2) case (i) and are given in Table 4.7:

Table 4.7

The computed values of the errors , I_1, I_2, and I_3 for double soliton case (i) with $h = 0.01$, $\Delta t = 0.005$, $0 \le x \le 2$

Time	$L_2 \times 10^3$	$L_\infty \times 10^3$	I_1	I_2	I_3
0.0			0.228074	0.103456	0.049855
0.50	0.061	0.130	0.228066	0.103457	0.049842
1.00	0.066	0.150	0.228068	0.103458	0.049858
1.50	0.083	0.214	0.228075	0.103459	0.049853
2.00	0.113	0.269	0.228077	0.103460	0.049837
2.50	0.150	0.386	0.228079	0.103461	0.049830
3.00	0.160	0.390	0.228076	0.103462	0.049838

From this Table we note that the L_2- and L_∞-error norms are still small even when the time reaches 3, and the quantities I_i ($i = 1,2,3$) are changed by less than 0.004% , 0.006% , 0.051% respectively during the computer run. We find that the computed value I_1 is better than that computed in problem (b). We conclude that the quantities I_i are virtually constants.

In case (i) the analytic solution (2.3.4.7) predicts that the two solitons will coalese near $x \approx 0.74$ at a time of $t \approx 0.85$. In our numerical solution this happens around at $t \approx 0.85$ and $x \approx 0.74$ which agrees with the analytic results. After the interaction the position of the maximum amplitudes at time 3 are determined analytically. The larger amplitude is at $x \simeq 1.49$ and the smaller amplitude at $x \simeq 0.95$.

The numerical solution agrees exactly with these analytic values.

It was found that after the interaction the solitary waves reappeared with their original amplitudes, correct to a numerical error of less than 0.8% , 0.38% respectively. We have also calculated the L_2- and L_∞-error norms and the first three conservative quantities. They are listed in Table 4.8:

Table 4.8

The computed values of the errors , I_1, I_2, and I_3 for double soliton case (ii) with h = 0.01, Δt = 0.005, $0 \leq x \leq 4$

Time	$L_2 \times 10^3$	$L_\infty \times 10^3$	I_1	I_2	I_3
0.0			0.228082	0.103456	0.049855
1.0	0.088	0.232	0.228084	0.103458	0.049826
2.0	0.151	0.336	0.228086	0.103460	0.049830
3.0	0.124	0.409	0.228088	0.103462	0.049855
4.0	0.199	0.504	0.228092	0.103464	0.049840
5.0	0.253	0.669	0.228094	0.103466	0.049835
6.0	0.279	0.743	0.228097	0.103467	0.049844
7.0	0.291	0.775	0.228100	0.103469	0.049845
8.0	0.296	0.793	0.228103	0.103471	0.049842

We observe that the behaviour of the L_2- and L_∞-error norms as the time increases to 8 are quite good, and the quantities I_i $(i = 1,2,3)$ change by less than 0.010% , 0.015% , 0.059% respectively during the computer run. Comparing these quantities with those obtained for problem (b1) we find that they are very similar.

The analytic solution predicts that the two well separated solitons will interact in the neighbourhood of $x \simeq 1.37$ at time $t \simeq 2.95$. In the numerical solution, we observe the interaction of $x \simeq 1.37$ at time $t \simeq 2.95$. After the interaction the soliton amplitudes have been changed from their original values by less than 0.006% , 0.032% respectively.

We have also studied the interaction of solitons of large amplitudes. The L_2- and L_∞-error norms and the quantities I_i for two well separated solitons of large amplitudes are recorded in Table 4.9:

Table 4.9

The computed values of the errrors , I_1, I_2, and I_3 for double soliton case (iii) with h = 0.1, Δt = 0.0005, $-12 \le x \le 12$

Time	$L_2 \times 10^3$	$L_\infty \times 10^3$	I_1	I_2	I_3
-0.5			11.99991	47.99998	211.2000
-0.4	1.431	1.536	12.00003	48.00010	210.9441
-0.3	2.604	2.728	12.00010	48.00057	210.9020
-0.2	3.625	3.873	12.00018	48.00109	210.9023
-0.1	4.273	4.521	12.00027	48.00162	210.9600
0.0	3.592	4.521	12.00038	48.00219	211.1746
0.1	4.605	4.669	12.00045	48.00283	211.0532
0.2	5.694	6.006	12.00054	48.00340	210.9515
0.3	6.110	6.580	12.00063	48.00395	210.9382
0.4	6.348	6.901	12.00075	48.00456	210.9532
0.5	6.426	6.901	12.00080	48.00508	210.9747

Table 4.9 shows us that the L_2- and L_∞-error norms increase as the time increases and these errors have the same magnitude, and that the quantities I_i ($i = 1,2,3$) change by less than 0.008% , 0.011% , 0.142% respectively during the computer run. We conclude that these quantities are relatively constant, particulary I_1 and I_2. Therefore, this method has the capability of dealing with the interaction of two solitons with large amplitudes.

The phase shifts Δ_1 , Δ_2 defined by equation (2.3.4.8) have been determined theoretically for problems (b), (b1), (b2(i-ii)) as:

$$\alpha_1 = \sqrt{c_{1/\mu}} \simeq 24.896 , \quad \alpha_2 = \sqrt{c_{2/\mu}} \simeq 14.374$$

then

$$\Delta_1 \simeq 0.11, \quad \text{and} \quad \Delta_2 \simeq -0.18 \tag{4.5.4}$$

For problem (b2(iii))

$$\Delta_1 \simeq 0.55, \quad \text{and} \quad \Delta_2 \simeq -1.1 \tag{4.5.5}$$

The forward and backward phase shifts have been computed from the numerical solution for problems (b), (b1), (b2(i-ii)) as

$$\Delta_1 \simeq 0.11, \quad \text{and} \quad \Delta_2 \simeq -0.18$$

and for problem (b2(iii))

$$\Delta_1 \simeq 0.50, \quad \text{and} \quad \Delta_2 \simeq -1.1$$

which agree exactly with the analytic results except in problem (b) there is error in the forward phase shift about 1%. Also in problem (b2(iii)) Δ_1 does not agree with its analytic result since $h = 0.1$.

From the analytic solution, we predict that the two well separated solitons with large amplitudes will interact in the neighbourhood of $x = 0$ at time $t = 0$. This event is observed in the numerical solution. the larger and smaller amplitudes have changed from their original values by less than 0.006% and 0.013% respectively.

Similar results are given in Table 4.10 for the conservative quantities I_1 , I_2 , I_3 of problems (c1) and (c2). We found that each of the quantities I_1 are very satisfactorily constant, I_2 particularly so:

Table 4.10

The computed values of I_1, I_2 and I_3 for $u(x,0) = \exp(-x^2)$

Time	I_1		I_2		I_3	
	μ =.04	μ =.01	μ = .04	μ = .01	μ =.04	μ =.01
0.0	1.772454	1.772454	1.253314	1.253314	0.872929	0.985728
2.5	1.772496	1.772474	1.253332	1.253342	0.872364	0.983127
5.0	1.772603	1.772507	1.253352	1.253358	0.872281	0.982074
7.5	1.771416	1.772538	1.253371	1.253367	0.871154	0.981986
10.0	1.775650	1.772548	1.253389	1.253375	0.868684	0.982014
12.5	1.770914	1.772469	1.253406	1.253387	0.860662	0.982004

From Table 4.10 we observe that the quantities I_i change by less than 0.181%, 0.008%, 1.406% respectively for $\mu = 0.04$, and 0.006%, 0.006%, 0.378% respectively for $\mu = 0.01$. Hence the degree of conservation observed for I_1, I_2, and I_3 could depend on the magnitude of the coefficient of the dispersive term (i.e. the value of μ).

The total number of solitons which are generated from a Gaussian initial condition can be determined [80,81] from:

$$N = \left[\frac{1}{13\,\mu}\right]^{1/2} \qquad (4.5.6)$$

We found this formula to be in agreement with the number of solitons observed in Figures 4.5 - 4.7 and 4.9 - 4.10 above.

We have computed the first three conservative quantities I_i for problem (d) with boundary conditions (ii). These are given in Table 4.11:

Table 4.11

The computed values I_1, I_2, and I_3 for problem (d) with boundary conditions (ii) $h = 0.4$, $\Delta t = 0.05$, $\varepsilon = 0.2$, $\mu = 0.1$

Time	I_1	I_2	I_3
0.0	50.00010	45.00046	42.30069
100.0	50.00142	45.00165	42.29480
200.0	50.00801	45.00554	42.23354
300.0	50.00586	45.00994	42.23244
400.0	49.99699	45.01448	42.25957
500.0	49.97831	45.01851	42.22482
600.0	49.96850	45.02282	42.22728
700.0	49.98072	45.02714	42.23331
800.0	50.00647	45.03198	42.30136

Table 4.11 shows us that even with computer runs of long duration that the quantities I_i $(i = 1,2,3)$ have changed by less than 0.064% , 0.071% , 0.180% respectively and so may be considered to be satisfactorily conserved. The analytic velocity c_a of a soliton is determined from its amplitude a by the formula:

$$c_a = a\,\varepsilon\,/3 \qquad (4.5.7)$$

where ε the coefficient of the nonlinear term. For this problem $a \simeq 1.96293$, $\varepsilon = 0.2$ so that $c_a \simeq 0.1309$. The observed velocity has been found to be $c_n \simeq 0.128$ which is consistent with c_a.

From the above discussion we deduce that Galerkin's method with cubic Hermite polynomial trial and test functions is a useful technique for solving the **KdV** equation with large space and time steps.

CHAPTER 5

CUBIC SPLINE INTERPOLATION FUNCTIONS

5.1 Introduction:

As we know well, the best choice for approximation functions are, in general, polynomials. As an alternative to the cubic Hermite functions discussed in the previous chapter we now choose an approximation polynomial which even has a continuous second derivative across element boundaries. This is the cubic spline interpolation polynomial.

We confine our attention in this chapter to finding a finite element solution of the **KdV** equation based on the Bubnov-Galerkin method using cubic splines as "shape" functions.

5.2 The Governing Equation:

We will study the Korteweg-de Vries equation:

$$u_t + \varepsilon\, u\, u_x + \mu\, u_{xxx} = 0 , \qquad a \le x \le b \qquad (5.2.1)$$

The boundary conditions will be chosen from:

$$\left.\begin{aligned} u(a,t) &= \beta_1 \\ u(b,t) &= \beta_2 \\ u_x(a,t) &= u_x(b,t) = 0 \\ u_{xx}(a,t) &= u_{xx}(b,t) = 0 \end{aligned}\right\} \qquad (5.2.2)$$

If we apply the Galerkin approach, with continuous weight functions $v(x)$, to equation (5.2.1) it produces:

$$\int_a^b v(u_t + \varepsilon\, u\, u_x + \mu\, u_{xxx})\, dx = 0 \qquad (5.2.3)$$

The presence of the third spatial derivative in the integral implies that the interpolation functions together with their first and second derivatives must be continuous throughout the region

$a \le x \le b$. The order of the highest derivative in the integral can, in this case, be reduced using integration by parts to obtain:

$$\int_a^b v\,(u_t + \varepsilon\, u\, u_x)dx - \int_a^b \mu\, v_x u_{xx}\,dx = -\left[\mu\, v\, u_{xx}\right]_a^b \qquad (5.2.4)$$

where; the right hand side of (5.2.4) is evaluated only at the boundaries. The condition on the interpolation functions is now simply that only the functions and their first derivatives need to be continuous throughout the region. Hermite cubic polynomials, and quadratic **B**-Splines are thus possible choices. However, we have chosen to use, as trial functions in this chapter, the very adaptable cubic splines with their well known advantages. We can thus proceed to a solution using either equation (5.2.4) or equation (5.2.3).

5.3 The Finite Element Solution [72,82,83,84]:

Now we are going to approximate the solution $u(x,t)$ using cubic **B**-Spline interpolation functions.

Let us consider

π: $a = x_0 < x_1 \ldots < x_N = b$ as a partition of $[a,b]$ by the knots x_i, and let $\phi_i(x)$ be those cubic **B**-Splines with knots at the points of π . Then $X_N = \text{span}\{\phi_{-1}, \phi_0, \ldots, \phi_N, \phi_{N+1}\}$ form a basis for functions defined over $[a,b]$. We seek the approximation $u_N(x,t)$ to the solution $u(x,t)$ which uses these splines as trial functions:

$$u_N(x,t) = \delta_{-1}(t)\,\phi_{-1}(x) + \delta_0(t)\,\phi_0(x) + \ldots + \delta_{N+1}(t)\,\phi_{N+1}(x)$$

$$= \sum_{m=-1}^{N+1} \delta_m(t)\,\phi_m(x) \qquad (5.3.1)$$

where; the δ_m are time dependent quantities to be determined from the boundary conditions (5.2.2) and from conditions based on either equation (5.2.3) or (5.2.4)

Cubic B-Splines ϕ_m with the required properties are defined by the relationships [35,71,72]:

$$\phi_m(x) = \frac{1}{h^3}\begin{cases} (x-x_{m-2})^3 & [x_{m-2}, x_{m-1}] \\ h^3 + 3h^2(x-x_{m-1}) + 3h(x-x_{m-1})^2 - 3(x-x_{m-1})^3 & [x_{m-1}, x_m] \\ h^3 + 3h^2(x_{m+1}-x) + 3h(x_{m+1}-x)^2 - 3(x_{m+1}-x)^3 & [x_m, x_{m+1}] \\ (x_{m+2}-x)^3 & [x_{m+1}, x_{m+2}] \\ 0 & \text{otherwise} \end{cases}$$

$$m = -1, 0, \ldots, N+1 \qquad (5.3.2)$$

where; $h = (x_{m+1} - x_m)$ for all m, implying that all intervals $[x_m, x_{m+1}]$ are of equal size.

The spline $\phi_m(x)$ and its two principle derivatives vanish outside the interval $[x_{m-2}, x_{m+2}]$. In Table (5.1) we list for convenience the values of $\phi_m(x)$ and its derivatives $\phi'_m(x)$, $\phi''_m(x)$ at the knots:

Table 5.1

x	x_{m-2}	x_{m-1}	x_m	x_{m+1}	x_{m+2}
ϕ_m	0	1	4	1	0
ϕ'_m	0	$\frac{3}{h}$	0	$-\frac{3}{h}$	0
ϕ''_m	0	$\frac{6}{h^2}$	$-\frac{12}{h^2}$	$\frac{6}{h^2}$	0

We now identify the finite elements for the problem with the intervals $[x_m, x_{m+1}]$ and the element nodes with the knots x_m,

x_{m+1} . Using equation (5.3.1) and Table 5.1, we see that the nodal parameters u_m are given in terms of the parameters δ_m by:

$$\left.\begin{aligned} u_m &= u(x_m) = \delta_{m-1} + 4\delta_m + \delta_{m+1} \\ u_{m+1} &= u(x_{m+1}) = \delta_m + 4\delta_{m+1} + \delta_{m+2} \end{aligned}\right\} \qquad (5.3.3)$$

and the variation of u over the element $[x_m , x_{m+1}]$ is given by:

$$u = \sum_{j=m-1}^{m+2} \delta_j \phi_j \qquad (5.3.4)$$

In addition, we have the valuable property that δ_{m-1} , δ_m , δ_{m+1} , δ_{m+2} determine also the first and second derivatives at the nodes (element boundaries) and that these are also continuous and given by:

$$u'_m = u'(x_m) = \frac{3}{h}\left[\delta_{m+1} - \delta_{m-1}\right] \qquad (5.3.5)$$

$$u''_m = u''(x_m) = \frac{6}{h^2}\left[\delta_{m-1} - 2\delta_m + \delta_{m+1}\right] \qquad (5.3.6)$$

The finite element equations we shall set up will not be expressed in terms of the nodal parameters u_m , u'_m , u''_m but in terms of the element parameters δ_m , so we shall not directly determine the nodal values as is the case with the usual finite element formulations. However these can always be recovered using equations (5.3.3), (5.3.5) and (5.3.6).

We now set up the element matrices relevant to equation (5.2.4). For a typical element $[x_m , x_{m+1}]$ we have the contribution:

$$\int_{x_m}^{x_{m+1}} \left[v(u_t + \varepsilon u u_x) - \mu v_x u_{xx} \right] dx \qquad (5.3.7)$$

From the equation (5.3.2) we see that each spline covers 4 elements so that each element $[x_m , x_{m+1}]$ is covered by 4 splines.

We define a local coordinate system ξ for the element by $\xi = x - x_m$, where $0 \le \xi \le h$, which enables the expressions for the element splines to be expressed independently of the actual element coordinates as:

$$\begin{matrix}\phi_{m-1}\\ \phi_m\\ \phi_{m+1}\\ \phi_{m+2}\end{matrix} = \frac{1}{h^3}\begin{cases}(h-\xi)^3\\ h^3 + 3h^2(h-\xi) + 3h(h-\xi)^2 - 3(h-\xi)^3\\ h^3 + 3h^2\xi + 3h\xi^2 - 3\xi^3\\ \xi^3\end{cases} \qquad (5.3.8)$$

$$0 \le \xi \le h$$

These splines act like "shape" functions for the element (see Figure 5.1) when we set up equations in terms of the element parameters δ_m^e.

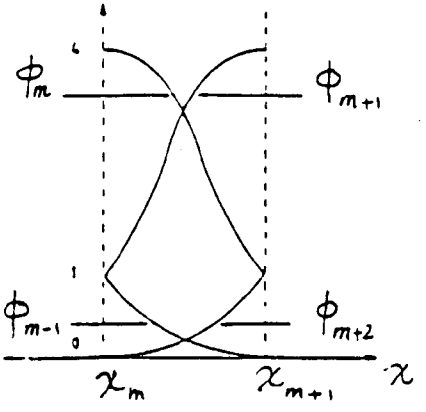

Figure 5.1 Cubic spline shape function for a typical element.

Now using (5.3.4) and (5.3.8) in (5.3.7) and identifying the weight functions with cubic splines we obtain:

$$\sum_{j=m-1}^{m+2}\left[\int_{x_m}^{x_{m+1}} \phi_i\,\phi_j\,dx\right]\dot{\delta}_j^e + \varepsilon\sum_{j=m-1}^{m+2}\sum_{k=m-1}^{m+2}\left[\int_{x_m}^{x_{m+1}} \phi_i\,\phi_j\phi_k'\,dx\right]\delta_k^e\,\delta_j^e$$

$$- \mu\sum_{j=m-1}^{m+2}\left[\int_{x_m}^{x_{m+1}} \phi_i'\,\phi_j''\,dx\right]\delta_j^e \qquad (5.3.9)$$

which can be written in matrix form as:

$$A^e \dot{\underset{\sim}{\delta}}^e + \varepsilon \, \underset{\sim}{\delta}^{eT} L^e \, \underset{\sim}{\delta}^e - \mu \, C^e \underset{\sim}{\delta}^e \qquad (5.3.10)$$

where:

$$\underset{\sim}{\delta}^e = (\delta_{m-1} \, , \, \delta_m \, , \, \delta_{m+1} \, , \, \delta_{m+2})^T$$

The element matrices are given by the integrals:

$$A^e_{ij} = \int_{x_m}^{x_{m+1}} \phi_i \, \phi_j \, dx \qquad (5.3.11)$$

$$C^e_{ij} = \int_{x_m}^{x_{m+1}} \phi'_i \, \phi''_j \, dx \qquad (5.3.12)$$

and

$$L^e_{ijk} = \int_{x_m}^{x_{m+1}} \phi_i \, \phi_j \, \phi'_k \, dx \qquad (5.3.13)$$

where; i , j , k take only the values $m-1$, m , $m+1$, $m+2$ for this typical element $[x_m \, , \, x_{m+1}]$. The matrices A^e , C^e are therefore 4×4 and the matrix L^e is $4 \times 4 \times 4$. An associated 4×4 matrix can be defined as:

$$B^e_{ij} = \sum_{k=m-1}^{m+2} L^e_{ijk} \, \delta^e_k \qquad (5.3.14)$$

which also depends on the parameters. δ^e_k will be used in the following theoretical discussions.

The element matrices A^e , C^e , L^e are independent of the parameters δ^e_k and can be determined algebraically from equations (5.3.11)-(5.3.13) using REDUCE [38] as:

$$A^e = \frac{h}{140}\begin{bmatrix} 20 & 129 & 60 & 1 \\ 129 & 1188 & 933 & 60 \\ 60 & 933 & 1188 & 129 \\ 1 & 60 & 129 & 20 \end{bmatrix} \qquad (5.3.15)$$

$$C^e = \frac{1}{2h^2}\begin{bmatrix} -9 & 15 & -3 & -3 \\ -15 & 9 & 27 & -21 \\ 21 & -27 & -9 & 15 \\ 3 & 3 & -15 & 9 \end{bmatrix} \qquad (5.3.16)$$

and

$$B^e_{m-1,m-1} = -\frac{1}{3}\delta_{m-1} - \frac{5}{28}\delta_m + \frac{1}{2}\delta_{m+1} + \frac{1}{84}\delta_{m+2}$$

$$B^e_{m-1,m} = B^e_{m,m-1} = -\frac{107}{56}\delta_{m-1} - \frac{87}{56}\delta_m + \frac{927}{280}\delta_{m+1} + \frac{43}{280}\delta_{m+2}$$

$$B^e_{m-1,m+1} = B^e_{m+1,m-1} = -\frac{3}{4}\delta_{m-1} - \frac{33}{35}\delta_m + \frac{219}{140}\delta_{m+1} + \frac{9}{70}\delta_{m+2}$$

$$B^e_{m-1,m+2} = B^e_{m+2,m-1} = -\frac{1}{168}\delta_{m-1} - \frac{1}{40}\delta_m + \frac{1}{40}\delta_{m+1} + \frac{1}{168}\delta_{m+2}$$

$$B^e_{m,m} = -\frac{361}{28}\delta_{m-1} - 21\,\delta_m + \frac{4167}{140}\delta_{m+1} + \frac{289}{70}\delta_{m+2}$$

$$B^e_{m,m+1} = B^e_{m+1,m} = -\frac{1783}{280}\delta_{m-1} - \frac{5847}{280}\delta_m + \frac{5847}{280}\delta_{m+1} + \frac{1783}{280}\delta_{m+2}$$

$$B^e_{m,m+2} = B^e_{m+2,m} = -\frac{9}{70}\delta_{m-1} - \frac{219}{140}\delta_m + \frac{33}{35}\delta_{m+1} + \frac{3}{4}\delta_{m+2}$$

$$B^e_{m+1,m+1} = -\frac{289}{70}\delta_{m-1} - \frac{4167}{140}\delta_m + 21\delta_{m+1} + \frac{361}{28}\delta_{m+2}$$

$$B^e_{m+1,m+2} = B^e_{m+2,m+1} = -\frac{43}{280}\delta_{m-1} - \frac{927}{280}\delta_m + \frac{87}{56}\delta_{m+1} + \frac{107}{56}\delta_{m+2}$$

$$B^e_{m+2,m+2} = -\frac{1}{84}\delta_{m-1} - \frac{1}{2}\delta_m + \frac{5}{28}\delta_{m+1} + \frac{1}{3}\delta_{m+2} \qquad (5.3.17)$$

Let us divide the region [a,b] into four elements of equal length h and the corresponding **KdV** equation becomes:

$$A\,\dot{\underset{\sim}{\delta}} + \varepsilon\, B(\underset{\sim}{\delta})\,\underset{\sim}{\delta} - \mu\,\underset{\sim}{\delta} = 0 \qquad (5.3.18)$$

where:

$$\underset{\sim}{\delta} = (\delta_{-1}, \delta_0, \delta_1, \delta_2, \delta_3, \delta_4, \delta_5)^T \qquad (5.3.19)$$

the matrices A , $B(\underset{\sim}{\delta})$, C assembled from the element matrices A^e ,

$B^e(\delta)$, C^e and are given in the form:

$$A = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} & & & \\ a_{21}^{(1)} & a_{11}^{*(2)} & a_{12}^{*(2)} & a_{13}^{*(2)} & a_{14}^{(2)} & & \\ a_{31}^{(1)} & a_{21}^{*(2)} & a_{11}^{*(3)} & a_{12}^{*(3)} & a_{13}^{*(3)} & a_{14}^{(3)} & \\ a_{41}^{(1)} & a_{31}^{*(2)} & a_{21}^{*(3)} & a_{11}^{*(4)} & a_{12}^{*(4)} & a_{13}^{*(4)} & a_{14}^{(4)} \\ & a_{41}^{(2)} & a_{31}^{*(3)} & a_{21}^{*(4)} & a_{22}^{*(4)} & a_{23}^{*(4)} & a_{24}^{(4)} \\ & & a_{41}^{(3)} & a_{31}^{*(4)} & a_{32}^{*(4)} & a_{33}^{*(4)} & a_{34}^{(4)} \\ & & & a_{41}^{(4)} & a_{42}^{(4)} & a_{43}^{(4)} & a_{44}^{(4)} \end{pmatrix} \quad (5.3.20)$$

where

$$a_{11}^{*(2)} = a_{22}^{(1)} + a_{11}^{(2)} \;,\; a_{12}^{*(2)} = a_{23}^{(1)} + a_{12}^{(2)} \;,\; a_{13}^{*(2)} = a_{24}^{(1)} + a_{13}^{(2)}$$

$$a_{21}^{*(2)} = a_{32}^{(1)} + a_{21}^{(2)} \;,\; a_{11}^{*(3)} = a_{11}^{(3)} + a_{22}^{(2)} + a_{33}^{(1)} \;,\; a_{12}^{*(3)} = a_{34}^{(1)} + a_{23}^{(2)} + a_{12}^{(3)}$$

$$a_{31}^{*(2)} = a_{42}^{(1)} + a_{31}^{(2)} \;,\; a_{21}^{*(3)} = a_{43}^{(1)} + a_{32}^{(2)} + a_{21}^{(3)} \;,\; a_{11}^{*(4)} = a_{11}^{(4)} + a_{22}^{(3)} + a_{33}^{(2)} + a_{44}^{(1)}$$

$$a_{31}^{*(3)} = a_{42}^{(2)} + a_{31}^{(3)} \;,\; a_{12}^{*(4)} = a_{34}^{(2)} + a_{23}^{(3)} + a_{12}^{(4)} \;,\; a_{13}^{*(4)} = a_{24}^{(3)} + a_{13}^{(4)}$$

$$a_{21}^{*(4)} = a_{43}^{(2)} + a_{32}^{(3)} + a_{21}^{(4)} \;,\; a_{22}^{*(4)} = a_{44}^{(2)} + a_{33}^{(3)} + a_{11}^{(4)} \;,\; a_{23}^{*(4)} = a_{34}^{(3)} + a_{23}^{(4)}$$

$$a_{31}^{*(4)} = a_{42}^{(3)} + a_{31}^{(4)} \;,\; a_{32}^{*(4)} = a_{43}^{(3)} + a_{32}^{(4)} \;,\; a_{33}^{*(4)} = a_{44}^{(3)} + a_{33}^{(4)} \;,\; a_{13}^{*(3)} = a_{24}^{(2)} + a_{13}^{(3)}$$

Similarly the matrices $B(\delta)$ and C can be expressed in the form (5.3.20).

Generally, dividing the region [a,b] into N elements of equal length h and combining contributions from all elements and following the procedure for four elements, produces the matrix equation:

$$A\,\dot{\delta} + \varepsilon\, B(\delta)\,\delta - \mu\, C\,\delta = 0 \quad (5.3.21)$$

where $\delta = (\delta_{-1}\,,\ \delta_0\,,\ \delta_1\,,\ \ldots\,,\ \delta_{N+1})^T$ (5.3.22)

$\underset{\sim}{\delta}$ are element parameters to be determined and A , $B(\underset{\sim}{\delta})$, C are matrices assembled from the element matrices A^e , $B^e(\underset{\sim}{\delta})$, C^e . The matrices A , $B(\underset{\sim}{\delta})$, C are $(N+3)x(N+3)$ 7-banded matrices.

We introduce a θ family of approximations which give a weighted average of the dependent variable and its time derivative:

$$\dot{\underset{\sim}{\delta}} = \frac{\underset{\sim}{\delta}^{n+1} - \underset{\sim}{\delta}^{n}}{\Delta t} \quad \text{and} \quad \underset{\sim}{\delta} = (1-\theta)\underset{\sim}{\delta}^{n} + \theta\underset{\sim}{\delta}^{n+1} \qquad (5.3.23)$$

where; $\underset{\sim}{\delta}^{n}$ are the parameters at time $n\Delta t$, and Δt is the time step

Substituting (5.3.23) in (5.3.21), we have:

$$\left[A + \theta\,\Delta t(\varepsilon B(\underset{\sim}{\delta}^{n}) - \mu\, C)\right]\underset{\sim}{\delta}^{n+1} = \left[A - (1-\theta)\Delta t(\varepsilon B(\underset{\sim}{\delta}^{n}) - \mu\, C)\right]\underset{\sim}{\delta}^{n} \qquad (5.3.24)$$

Giving the parameter θ the values o , $\frac{1}{2}$, and 1 produces forward, Crank-Nicolson and backward difference schemes respectively

Now let $\theta = \frac{1}{2}$ and equation (5.3.24) becomes:

$$\left[A + \frac{\Delta t}{2}(\varepsilon B(\underset{\sim}{\delta}^{n}) - \mu\, C)\right]\underset{\sim}{\delta}^{n+1} = \left[A - \frac{\Delta t}{2}(\varepsilon B(\underset{\sim}{\delta}^{n}) - \mu\, C)\right]\underset{\sim}{\delta}^{n} \qquad (5.3.25)$$

The matrices A , C are independent of the time so, they will remain constant throughout the calculations. While the matrix $B(\underset{\sim}{\delta})$ is dependent on the time, it must therefore be recalculated at each time step.

Since the matrix B depends on the time through the parameter $\underset{\sim}{\delta}$, the matrix equation (5.3.25) is nonlinear and our approach is modified so that instead of solving the equation (5.3.25) we solve an equivalent system [56,70]:

$$\left[A + \frac{\Delta t}{2}\left(\varepsilon B(\underset{\sim}{\delta}^{n}) - \mu C\right)\right]\underset{\sim}{\hat{\delta}}^{n+1} = \left[A - \frac{\Delta t}{2}\left(\varepsilon B(\underset{\sim}{\delta}^{n}) - \mu C\right)\right]\underset{\sim}{\delta}^{n} \qquad (5.3.26a)$$

and

$$\left[A + \frac{\Delta t}{2}\left(\varepsilon B\left(\frac{\underset{\sim}{\hat{\delta}}^{n+1} + \underset{\sim}{\delta}^{n}}{2}\right) - \mu C\right)\right]\underset{\sim}{\delta}^{n+1} = \left[A - \frac{\Delta t}{2}\left(\varepsilon B\left(\frac{\underset{\sim}{\hat{\delta}}^{n+1} + \underset{\sim}{\delta}^{n}}{2}\right) - \mu C\right)\right]\underset{\sim}{\delta}^{n} \qquad (5.3.26b)$$

The predictor (5.3.26a) gives a first approximation $\underset{\sim}{\hat{\delta}}^{n+1}$ then the corrector (5.3.26b) may be used iteratively to improve the approximation.

Before solving the system (5.3.26), we must apply the boundary conditions which for the present formulation require the products:

$$u(a,t)\, u_{xx}(a,t) = u(b,t)\, u_{xx}(b,t) = 0$$

In particular, if we choose to prescribe the boundary conditions:

$$u(a,t) = \beta_1 \quad , \quad u(b,t) = \beta_2$$

then

$$u_{xx}(a,t) = u_{xx}(b,t) = 0$$

and we must impose the conditions:

$$\left.\begin{aligned} \delta_{-1} + 4\delta_0 + \delta_1 &= \beta_1 \\ \delta_{-1} - 2\delta_0 + \delta_1 &= 0 \\ \delta_{N-1} + 4\delta_N + \delta_{N+1} &= \beta_2 \\ \delta_{N-1} - 2\delta_N + \delta_{N+1} &= 0 \end{aligned}\right\} \qquad (5.3.27)$$

Eliminating δ_{-1} , δ_0 , δ_N , δ_{N+1} from equations (5.3.26) which then becomes a recurrence relationship for $(\delta_1^n , \delta_2^n , \ldots , \delta_{N-1}^n)^T$.

Now equations (5.3.26) are (N-1)x(N-1) 7-banded matrices. In solving equations (5.3.26) we first store these matrices in rectangular form (N-1)x7 and then use a septa-diagonal algorithm,

based on the Thomas algorithm for tridiagonal matrices (see Appendix A3), to solve the equations directly. The boundary parameters δ_{-1} , δ_0 , δ_N , δ_{N+1} can be calculated at each time step from equations (5.3.27).

To start the solution procedure (5.3.26) a starting vector $\underset{\sim}{\delta}^0$ must be determined from the initial condition on $u(x,t)$. Once the parameters $\underset{\sim}{\delta}$ have been found at a time t, then we can evaluate the solution at each node from the formula:

$$u(x_i,t) = \delta_{i-1} + 4\delta_i + \delta_{i+1} \qquad (5.3.28)$$
$$i = 0 , 1 , \ldots , N$$

5.4 An Alternative Formulation:

When inhomogenous boundary conditions of the form $u(a,t) = \beta_1$, $u(b,t) = \beta_2$ hold and u_{xx} is not prescribed zero at the ends Galerkin method with cubic spline shape functions can still be used since the second derivative is continuous across element boundaries. An alternative formulation based on equation (5.2.3):

$$\int_a^b v(u_t + \varepsilon\, u\, u_x + \mu\, u_{xxx}\ )\, dx = 0 \qquad (5.4.1)$$

is used.

Proceeding as in section (5.3) we obtain the recurrence relationship replacing (5.3.26) as:

$$\left[A + \frac{\Delta t}{2}\left(\varepsilon B(\underset{\sim}{\delta}^{n}) + \mu K\right)\right]\underset{\sim}{\hat{\delta}}^{n+1} = \left[A - \frac{\Delta t}{2}\left(\varepsilon B(\underset{\sim}{\delta}^{n}) + \mu K\right)\right]\underset{\sim}{\delta}^{n} \qquad (5.4.2a)$$

and

$$\left[A + \frac{\Delta t}{2}\left(\varepsilon B\left(\frac{\underset{\sim}{\hat{\delta}}^{n+1} + \underset{\sim}{\delta}^{n}}{2}\right) + \mu K\right)\right]\underset{\sim}{\delta}^{n+1} = \left[A - \frac{\Delta t}{2}\left(\varepsilon B\left(\frac{\underset{\sim}{\hat{\delta}}^{n+1} + \underset{\sim}{\delta}^{n}}{2}\right) + \mu K\right)\right]\underset{\sim}{\delta}^{n} \qquad (5.4.2b)$$

where; the matrix K has replaced the matrix C . Matrix K is obtained from element matrices K^e which have again been determined using REDUCE [38]:

$$K^e = \frac{1}{2h^2}\begin{bmatrix} -3 & 9 & -9 & 3 \\ -33 & 99 & -99 & 33 \\ -33 & 99 & -99 & 33 \\ -3 & 9 & -9 & 3 \end{bmatrix} \qquad (5.4.3)$$

Follow the procedure given in section (5.3) to obtain the solution corresponding to this approach.

5.5 The Initial State:

From the initial condition $\mathbf{u}(x,0)$ on the function $\mathbf{u}(x,t)$ we determine the initial vector $\underset{\sim}{\delta}^{0}$ by interpolating $\mathbf{u}(x,0)$ using cubic splines.

We firstly rewrite equation (5.3.1) for the initial condition:

$$\mathbf{u}_N(x,0) = \sum_{i=-1}^{N+1} \delta_i^0 \, \phi_i(x) \qquad (5.5.1)$$

where the δ_i^0 must be determined. To do this we require $\mathbf{u}_N(x,0)$ to satisfy the following constraints:

(a) It shall agree with the initial condition $\mathbf{u}_N(x,0)$ at the knots x_i, $i = 0, 1, \ldots, N$; leading to $N+1$ conditions, and

(b) The second derivative of the approximation initial condition

shall agree with that of the exact initial condition at both ends of the range and two futher conditions.

The above conditions (a) and (b) can be written as:

$$\left.\begin{aligned} u_N''(x_0,0) &= 0 \\ u_N(x_i,0) &= u(x_i,0) \qquad 0 \le i \le N \\ u_N''(x_N,0) &= 0 \end{aligned}\right\} \tag{5.5.2}$$

Using Table (5.1) the system (5.5.2) leads to a matrix equation of the form:

$$A\,\underset{\sim}{\delta}^0 = \underset{\sim}{b} \tag{5.5.3}$$

where:

$$A = \begin{bmatrix} 6 & -12 & 6 & & & & & & \\ 1 & 4 & 1 & & & & & & \\ & 1 & 4 & 1 & & & & & \\ & & 1 & 4 & 1 & & & & \\ & & & \vdots & & & & & \\ & & \cdots & \vdots & \cdots & & & & \\ & & & \vdots & & & & & \\ & & & & & 1 & 4 & 1 & \\ & & & & & & 1 & 4 & 1 \\ & & & & & & 6 & -12 & 6 \end{bmatrix} \tag{5.5.4}$$

$$\underset{\sim}{\delta}^0 = \left[\, \delta^0_{-1} \,,\, \delta^0_0 \,,\, \ldots \,,\, \delta^0_{N+1} \,\right]^T \tag{5.5.5}$$

and

$$\underset{\sim}{b} = \left[\, 0 \,,\, u(x_0) \,,\, u(x_1) \,,\, \ldots \,,\, u(x_N) \,,\, 0 \,\right]^T \tag{5.5.6}$$

To solve this matrix equation, first reduce it to tridiagonal form by eliminating the first and last equations and then apply the Thomas algorithm (see Appendix A1) to get the initial vector $\underset{\sim}{\delta}^0$.

5.6 Stability Analysis:

Like other authors [32,35,36,44], our stability analysis will be based on the Von Neumann theory in which the growth factor of a

typical Fourier mode defined as:

$$\delta_j^n = \hat{\delta}^n e^{ijkh} \tag{5.6.1}$$

where; k is the mode number and h is the element size, is determined for a linearisation of the numerical scheme.

The nonlinear term $\mathbf{u}\,\mathbf{u}_x$ of the **KdV** equation cannot be handled by the Fourier mode method, therefore we linearise it [32,35,36,44]. To do this assume that the quantity **u** in the nonlinear term $\mathbf{u}\,\mathbf{u}_x$ is locally constant. This is equivalent to assuming that all δ_j^n are equal to a local constant d hence the linearised matrix B can be computed using REDUCE [38] to be:

$$B = d \begin{bmatrix} -3 & -\frac{213}{10} & -\frac{57}{5} & -\frac{3}{10} \\ -\frac{27}{10} & -45 & -\frac{549}{10} & \frac{27}{5} \\ \frac{27}{5} & \frac{549}{10} & 45 & \frac{27}{10} \\ \frac{3}{10} & \frac{57}{5} & \frac{213}{10} & 3 \end{bmatrix}^T \tag{5.6.2}$$

A typical member of equation (5.3.25), given that B is determined from (5.6.2), is of the form:

$$\alpha_1\delta_{i-3}^{n+1} + \alpha_2\delta_{i-2}^{n+1} + \alpha_3\delta_{i-1}^{n+1} + \alpha_4\delta_i^{n+1} + \alpha_5\delta_{i+1}^{n+1} + \alpha_6\delta_{i+2}^{n+1} + \alpha_7\delta_{i+3}^{n+1}$$

$$= \alpha_7\delta_{i-3}^{n} + \alpha_6\delta_{i-2}^{n} + \alpha_5\delta_{i-1}^{n} + \alpha_4\delta_i^{n} + \alpha_3\delta_{i+1}^{n} + \alpha_2\delta_{i+2}^{n} + \alpha_1\delta_{i+3}^{n} \tag{5.6.3}$$

where:

$$\alpha_1 = \alpha - 6\beta - \gamma \quad , \quad \alpha_2 = 120\alpha - 336\beta - 8\gamma \quad ,$$

$$\alpha_3 = 1191\alpha - 1470\beta + 19\gamma \quad , \quad \alpha_4 = 2416\alpha, \quad \alpha_5 = 1191\alpha + 1470\beta - 19\gamma$$

$$\alpha_6 = 120\alpha + 336\beta + 8\gamma \quad , \quad \alpha_7 = \alpha + 6\beta + \gamma$$

$$\alpha = \frac{h}{140} \quad , \quad \beta = \frac{\varepsilon d \Delta t}{40} \quad , \quad \gamma = \frac{3\mu\Delta t}{4h^2} \tag{5.6.4}$$

Substituting equation (5.6.1) into equation (5.6.3), we obtain:

$$(a + ib)\hat{\delta}^{n+1} = (a - ib)\hat{\delta}^{n} \qquad (5.6.5)$$

where:

$$i = \sqrt{-1}$$

$$a = \alpha(1208 + \cos(3kh) + 120\cos(2kh) + 1191\cos(kh)$$

$$b = (6\beta + \gamma)\sin(3kh) + (336\beta + 8\gamma)\sin(2kh) + (1470\beta + 19\gamma)\sin(kh) \qquad (5.6.6)$$

Let $\hat{\underset{\sim}{\delta}}^{n+1} = g\,\hat{\underset{\sim}{\delta}}^{n}$ and substitute in (5.6.5) to give:

$$g = \frac{a - ib}{a + ib} \qquad (5.6.7)$$

where g is the growth factor for the mode.

The modulus of the growth factor is $|g| = \sqrt{g\bar{g}} = 1$. Hence the linearised scheme is unconditionally stable.

5.7 The Test Problems:

In this section we are going to test our algorithm by studying four classical problems concerning the motion of solitons, their interactions and their generation from arbitrary initial conditions.

(a) The initial condition:

$$u(x,0) = 3\,c\,\mathrm{sech}^2(A_1 x + D_1)\,, \qquad (5.7.1)$$

follows from the analytic solution of the **KdV** equation which has the form:

$$u(x,t) = 3\,c\,\mathrm{sech}^2(A_1 x - B_1 t + D_1)\,, \qquad (5.7.2)$$

where $A_1 = \frac{1}{2}(\varepsilon\, c/\mu)^{1/2}$ and $B_1 = \varepsilon\, cA_1$. To permit comparison with Greig and Morris [26] and Sanz-Serna [36] we choose $\varepsilon = 1$, $\mu = 4.84\times10^{-4}$, $c = 0.3$, $D_1 = -6$, $h = 0.05$, 0.033 , 0.01 , and $\Delta t = 0.025$, 0.01 , 0.005 . We shall impose the boundary conditions:

$$\left.\begin{aligned} u(0,t) &= u(2,t) = 0 \\ u_{xx}(0,t) &= u_{xx}(2,t) = 0 \end{aligned}\right\} \text{ for all time} \qquad (5.7.3)$$

These conditions represent a single soliton moving to the right with constant speed c and unchanged amplitude 3c. Our solution is plotted in Figure 5.2 from time t = 0.0 to 3.0. When the exact solution is plotted on the same diagram the curves cannot be distinguished. These graphs compared exactly with those of Greig and Morris [26] for corresponding times.

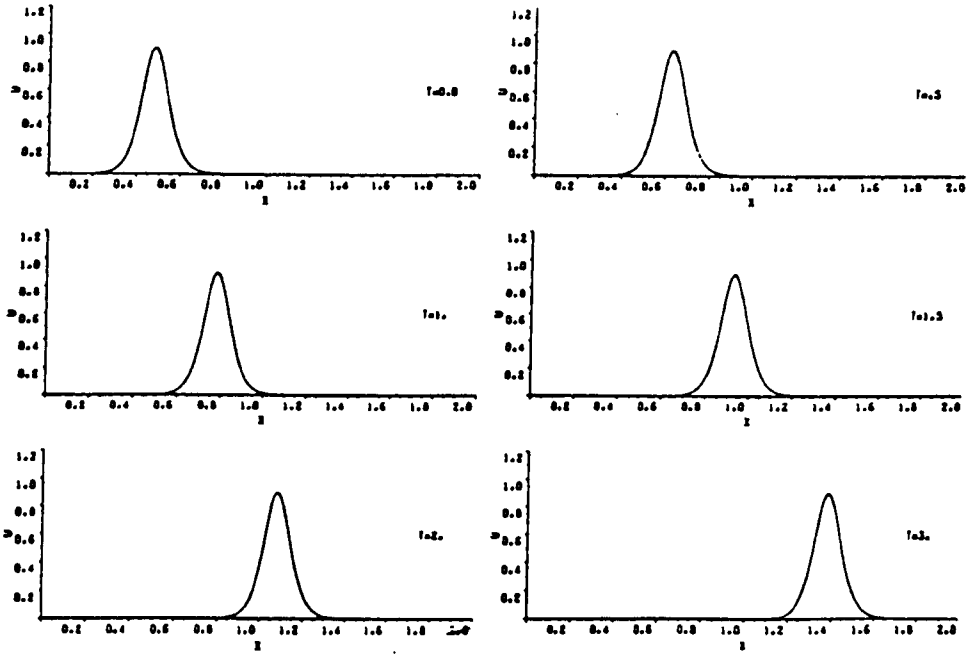

Figure 5.2 Problem (a). The moving of a single solitary wave from t = 0.0 to t = 3.0 with h = 0.01 , Δt = 0.005.

Our second example concerns the interaction between solitons. We use the initial condition:

$$\text{(b)} \quad u(x,o) = 3\,c_1 \operatorname{sech}^2(A_1 x + D_1) + 3\,c_2 \operatorname{sech}^2(A_2 x + D_2) \qquad (5.7.4)$$

and the boundary conditions:

$$\left.\begin{aligned} u(0,t) &= u(2,t) = 0 \\ u_{xx}(0,t) &= u_{xx}(2,t) = 0 \end{aligned}\right\} \text{ for all time.} \qquad (5.7.5)$$

For comparison with an earlier solution [26] we have chosen $c_1 = 0.3$, $c_2 = 0.1$, $D_1 = D_2 = -6.$ and $A_j = \frac{1}{2}\left(\frac{\varepsilon c}{\mu} j\right)^{1/2}$, $j = 1, 2$.

This choice gives us two solitons initially sited at $x = -D_1/A_1$ and $-D_2/A_2$ with velocities proportional to their amplitudes, which interact as time increases. In Figure 5.3. we see that the two solitons with the taller one to the left of the shorter one. Because of the greater speed, the taller soliton eventually catches up with the shorter one and they undergo a nonlinear interaction according to the KdV equation at time $t = 0.75$. The overlapping process continues until, at time $t = 1.5$, the larger soliton has overtaken the smaller one and is in the process of separating. At time $t = 3.0$ the interaction is complete and the larger soliton has separated completely from the smaller one:

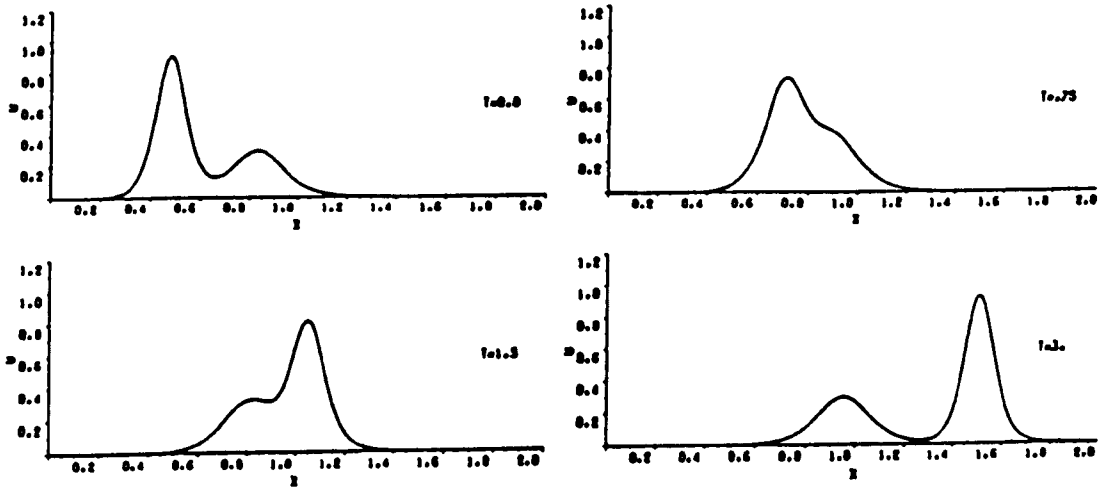

Figure 5.3 Problem (b). The interaction of the solitons with $\Delta t = 0.005$, $h = 0.01$.

Our simulations in Figure 5.3 show that initially the two solitons overlap and that after the interaction the large and small amplitudes of the solitons are modified by about 0.94% , 0.45% respectively as was also observed by Greig and Morris [26]; the agreement between the solutions is satisfactory.

Consider the initial condition of the two well separated

solitons:

(b1) $u(x,o) = 3\,c_1 \mathrm{sech}^2(A_1 x + D_1) + 3\,c_2 \mathrm{sech}^2(A_2 x + D_2)$ (5.7.6)

where; the values of the parameters are given in problem (b) except that now we take $D_2 = -9.0$. The boundary conditions are taken to be:

$$\left.\begin{aligned} u(0,t) &= u(4,t) = 0 \\ u_{xx}(0,t) &= u_{xx}(4,t) = 0 \end{aligned}\right\} \text{ for all time.} \qquad (5.7.7)$$

Figure 5.4 shows us that after the interaction of the two well separated solitons the large and small amplitudes have changed by only a very small amount (0.017% , 0.04%) respectively. So, we can say that after the interaction the amplitudes are virtually unchanged:

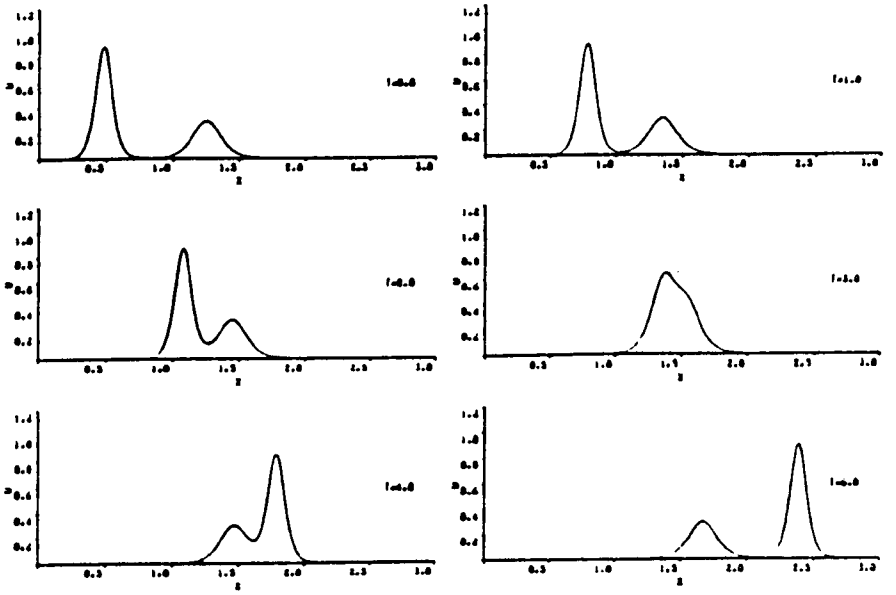

Figure 5.4 Problem (b1). The motion of the two well separated solitons with h = 0.01 , Δt = 0.005.

Figure 5.4 shows us that initially the two solitons placed up with the larger to the left of and separated from the smaller. As the time increases, the larger soliton catches up the smaller at time t = 3. The larger soliton overtakes the smaller, accelerates through it and emerges unaffected when the time reaches t = 4. By time t = 6, the interaction is complete and the larger soliton has separated completely from the smaller one.

Our third test example has the inital condition:

$$(\ c\ ) \qquad u(x\ ,0) = \exp(\ -\ x^2) \tag{5.7.8}$$

The boundary conditions are taken as:

$$\left.\begin{aligned} u(\mp\ 15, t) &= 0 \\ u_{xx}(\mp\ 15, t) &= 0 \end{aligned}\right\} \quad \text{for all } t > 0\ . \tag{5.7.9}$$

We choose $\varepsilon = 1.0$ and each of the following:

(c1) $\mu = 0.04$, $h = 0.1$, $\Delta t = 0.01$

(c2) $\mu = 0.01$, $h = 0.1$, $\Delta t = 0.01$

(c3) $\mu = 0.001$, $h = 0.025$, $\Delta t = 0.005$

(c4) $\mu = 0.0005$, $h = 0.025$, $\Delta t = 0.005$

Comparison with the work of Goda [59] in the cases (c1) and (c2) is made.

Figure 5.5 shows the numerical solution of problem (c1) for times up to 12.5. We see that a mixed type of solution was found which consists of a leading soliton and an oscillating tail. We observe the velocity of the soliton to be $c_n \simeq 0.4$ which agrees with the value calculated from its amplitude of $c_a \simeq 0.3993$:

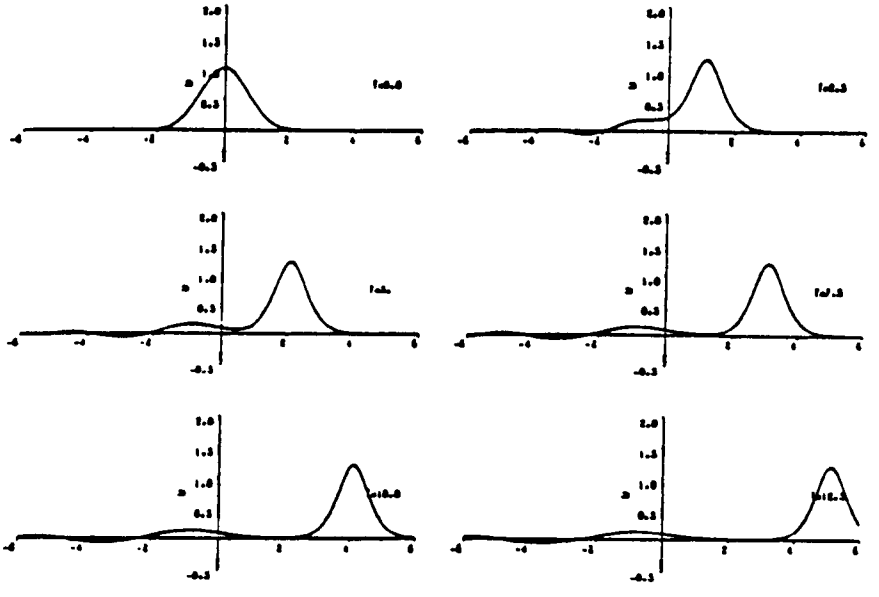

Figure 5.5 Problem (c1). A single soliton with oscillating tail for $\mu = 0.04$ from $t = 0.0$ to $t = 12.5$.

Figure 5.6 shows similar results for $\mu = 0.01$. Now the initial perturbation splits into three solitons. The graphs

produced in both cases in the present work are identical with those given by Goda [59]. The observed velocity of the leading solution $c_n \simeq 0.52$ agrees with that calculated from its amplitude ($c_a \simeq 0.5148$):

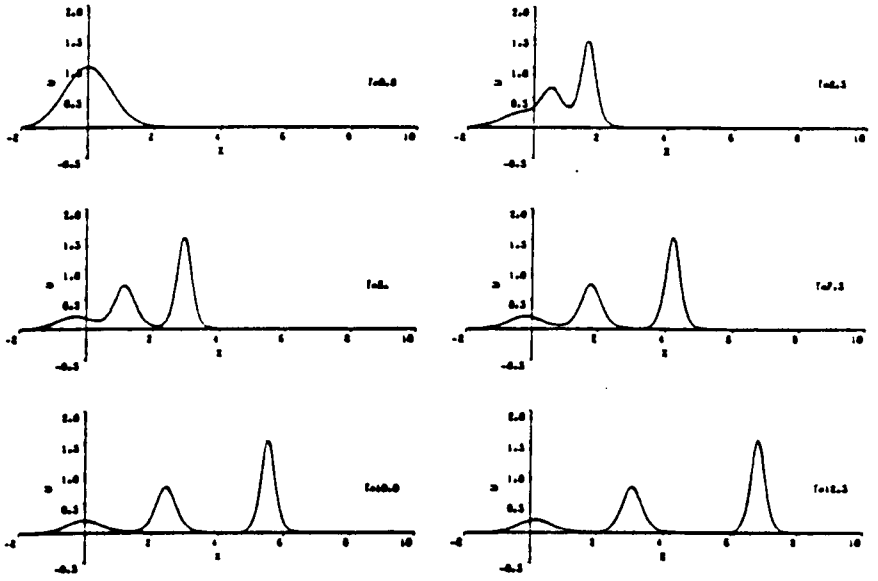

Figure 5.6 Problem (c2). The spliting of the initial condition into 3 solitons when $\mu = 0.01$, $h = 0.1$, $\Delta t = 0.01$.

In Figure 5.7 we see that when the coefficient of the dispersive term is decreased to $\mu = 0.001$, the nonlinear term dominates. The initial perturbation breaks up into 9 solitons moving to the right, decreasing in amplitude and velocity from right to left:

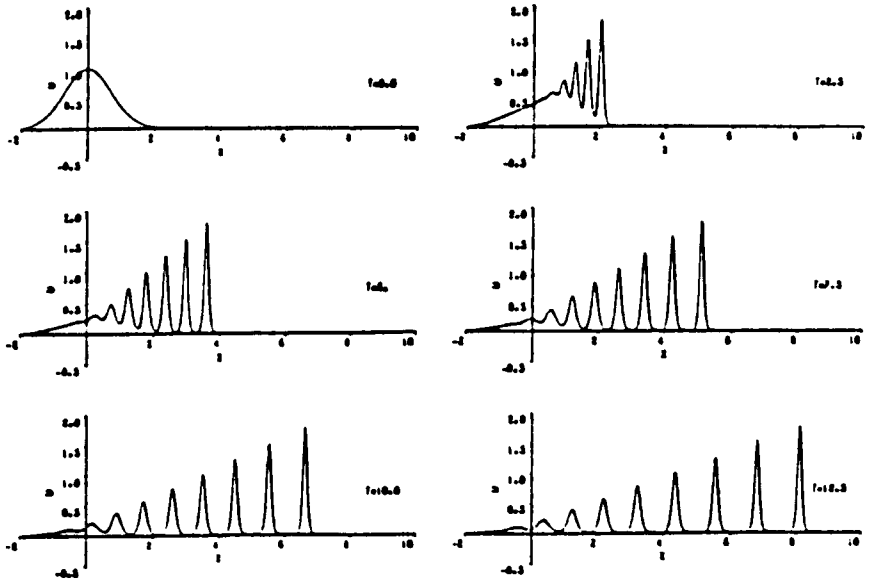

Figure 5.7 Problem (c3). The breakdown of the initial perturbation into 9 solitons with $\mu = 0.001$, $h = 0.025$, $\Delta t = 0.005$.

Figure 5.8 shows that when the coefficient of the dispersive term is made smaller ($\mu = 0.0005$), the initial perturbation breaks down into 12 solitons:

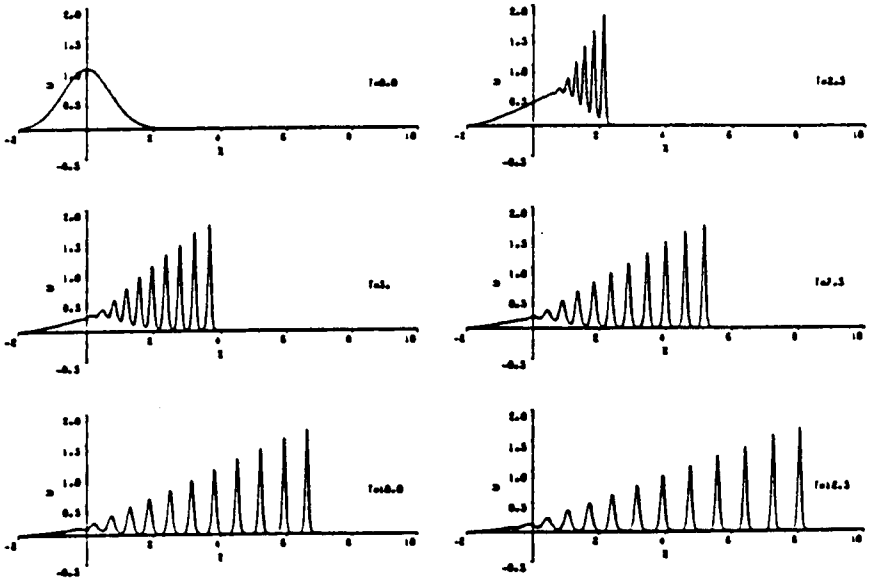

Figure 5.8 Problem (c4). The initial condition split into 12 solitons when $\mu = 0.0005$, $h = 0.025$, $\Delta t = 0.005$.

It was found that the behaviour of the numerical solutions varied according to the value of μ chosen. The initial perturbation (5.7.8) was observed to split into a train of solitons, the numbers of which increased as the coefficient of the dispersive term was decreased.

A comparison between our numerical results and those obtained in references [15,39,59,80] has been made for $\mu = 0.04$, 0.01 and they were in complete agreement.

As a final test example, we consider the development of an undular bore in shallow water. This is characterised by the initial condition:

$$\text{(d)} \qquad u(x,0) = \frac{1}{2}\left[1 - \tanh\left[\frac{x - 25}{5}\right]\right] \qquad (5.7.10)$$

and the boundary conditions we impose are:

$$\text{(i)}\qquad \left.\begin{aligned} u(0,t) &= 1 \\ u(50,t) &= 0 \\ u_{xx}(0,t) &= u_{xx}(50,t) = 0 \end{aligned}\right\} \text{ for all } t > o \qquad (5.7.11)$$

$$\text{(ii)}\qquad \left.\begin{aligned} u(-50,t) &= u(150,t) = 0 \\ u_{xx}(-50,t) &= u_{xx}(150,t) = 0 \end{aligned}\right\} \text{ for all } t > o \qquad (5.7.12)$$

To allow comparison of problem (d) boundary conditions (i) with Vliegenthart [44], we have chosen $\varepsilon = 0.2$, $\mu = 0.1$ and taken $\Delta t = 0.05$ and $h = 0.4$. The solution we obtained (see Figure 5.9) shows all the general features obtained in the earlier solution [44]. However, we cannot make a direct comparison with those results since the boundary conditions used by Vliegenthart are not given. Instead we have used his finite difference scheme and parameters with our boundary condtions to produce comparable figures. If these are plotted also on Figure 5.9, the graphs are indistinguishable with those obtained in the present study:

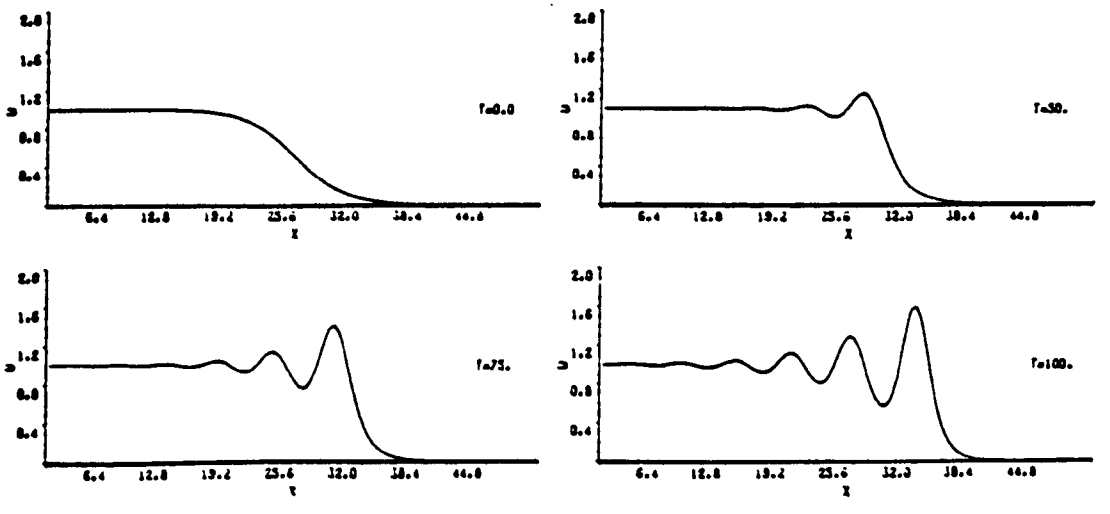

Figure 5.9 Problem (d) boundary conditions (i). The solution graphs for various times $\Delta t = 0.05$, $h = 0.4$, $\varepsilon = 0.2$, $\mu = 0.1$.

We find that the quantities I_1 of the problem (d) with

boundary conditions (i) are not conserved. Therefore, we have chosen to use the boundary conditions (ii) together with an alternative initial condition which can be seen in Figure 5.10:

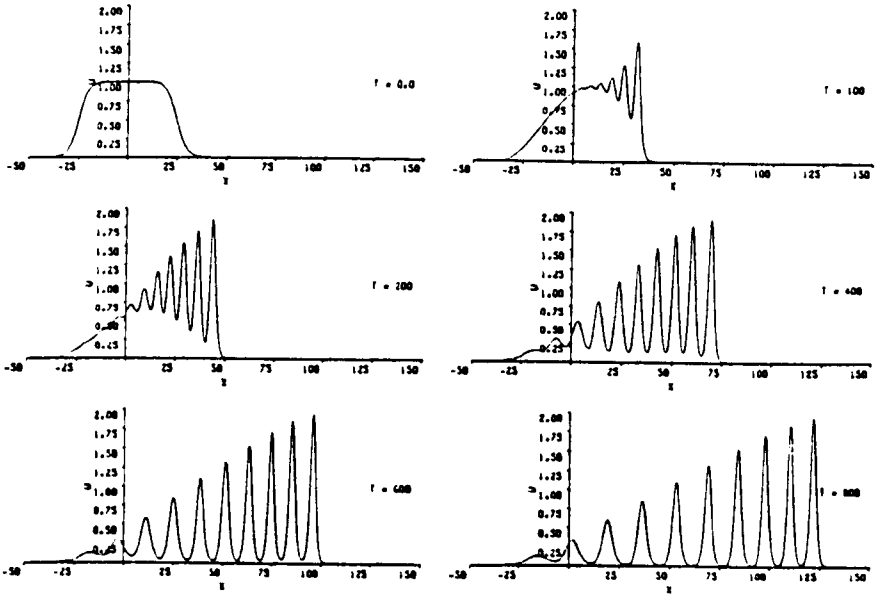

Figure 5.10 Problem (d) boundary conditions (ii). The solution graphs for various times $\Delta t = 0.05$, $h = 0.4$, $\varepsilon = 0.2$, $\mu = 0.1$.

As expected the initial perturbation of problem (d) with boundary conditions (ii) degenerated into a 10-train of solitons, which move steadily to the right with constant amplitudes and velocities during the computer runs up to time $t = 800$. The agreement between the analytic velocity $c_a \simeq 0.131$ and the numerical velocity $c_n \simeq 0.128$ for the leading soliton was very satisfactory.

5.8 Discussion.

5.8.1 Simulations Using Scheme (5.3.26):

We have found that our numerical single solution soliton of problem (a) is indistinguishable from the analytic results to within plotting error. To investigate more accurately how

faithfully the numerical scheme calculates the amplitude and position of the solution we use the L_2- and L_∞-error norms to compare the numerical and exact solutions.

The L_2-error norm is defined by (4.5.1). This error is used to compare 5 numerical methods in Table 5.2 for a single soliton solution [36] with two different boundary conditions the first (spI) the function and its second derivative vanishing at the endpoints, and the second (spII) the function and its first derivative vanishing at the endpoints:

Table 5.2

The growth of the $L_2 \times 10^3$ for single soliton

Time	Zabusky Kruskal [27]	Hopscotch [26]	Petrov-Galerkin [36]	Modified P-G [36]	Galerkin cubic spline spI	Galerkin cubic spline spII
$\Delta x = 0.05$ $\Delta t = 0.025$			$h = 0.05$ $\Delta t = 0.025$			
0.25	34.64	61.21	81.39	52.15	13.27	13.50
0.50	122.68	122.41	102.54	64.90	21.95	19.93
0.75	210.44	181.35	125.84	89.01	25.67	18.01
1.00	298.19	228.10	150.57	107.20	29.45	18.44
			$h = 0.033$ $\Delta t = 0.01$			
0.25			31.18	5.94	1.34	0.93
0.5			43.35	7.56	1.82	0.97
0.75			56.21	8.70	2.30	0.94
1.00			74.08	9.49	2.69	1.60
$\Delta x = 0.01$ $\Delta t = 0.0005$			$h = 0.01$ $\Delta t = 0.005$			
0.25	5.94	3.79	4.46	0.21	0.02	0.02
0.50	13.17	9.28	7.01	0.38	0.04	0.04
0.75	21.08	14.14	10.08	0.57	0.06	0.06
1.00	28.66	18.72	13.26	0.74	0.08	0.08

We find that the Galerkin cubic spline method (spI) compares well

with the best of the other methods and the error of this numerical method is smaller than the best numerical method quoted in Table 5.2 by a factor of at least 4, which increases to 10 when $h = 0.01$, $\Delta t = 0.005$. Furthermore, the error calculated by the method spII is better than that calculated by the method spI when $h = 0.05$, $\Delta t = 0.025$ and $h = 0.033$, $\Delta t = 0.01$. But for $h = 0.01$, $\Delta t = 0.005$ they have the same error of magnitude:

Table 5.3

The growth of the $L_\infty \times 10^3$ for single soliton

Time	Zabusky Kruskal [27]	Hopscotch [26]	Petrov-Galerkin [36]	Modified P-G [36]	Galerkin cubic spI	Galerkin spline spII
$\Delta x = 0.05$ $\Delta t = 0.025$			$h = 0.05$, $\Delta t = 0.025$			
0.25	19.4	32.7	42.18	30.22	15.36	15.44
0.50	63.5	67.4	51.85	22.85	56.28	26.47
0.75	122.4	99.3	87.60	35.86	52.54	29.52
1.00	161.4	141.6	100.41	39.39	49.78	27.45
			$h = 0.033$ $\Delta t = 0.01$			
0.25			14.27	2.80	4.02	1.77
0.5			21.65	4.53	3.40	2.06
0.75			29.78	4.85	3.63	2.10
1.00			39.37	5.85	3.78	2.66
$\Delta x = 0.01$ $\Delta t = 0.0005$			$h = 0.01$ $\Delta t = 0.005$			
0.25	2.05	1.11	1.21	0.07	0.07	0.07
0.50	4.22	2.14	2.15	0.11	0.11	0.11
0.75	6.36	3.54	3.09	0.17	0.16	0.16
1.00	8.13	4.91	3.83	0.21	0.20	0.20

Table 5.3 shows us that the L_∞-error norm computed from the methods spI and spII for a single soliton using the definition (4.5.2) have been compared with all the method which are quoted in

Table 5.3. We observe that the L_∞-error norm is greater than L_2-error norm which disagrrees with the authors [26,27,36]. Also we find that the value of L_∞-error norm using spI scheme for all the values of space and time sterps is smaller than all the other methods quoted in Table 5.3 except Modified Petrov-Galerkin when h = 0.05, Δt = 0.025 is still smaller and has the same error of magnitude when the space and the time steps decrease. The L_∞-error using the scheme spII is smaller than all other method quoted in Table 5.3 especially when h = 0.05 , Δt = 0.025 and h = 0.033 , Δt = 0.01. It has the same error of magnitude with the best method (Modified Petrov-Galerkin) when h = 0.01 , Δt = 0.005.

Table 5.4 gives the error for a single soliton problem (a) and from it we can say that the cubic spline method (spII) gives an acceptably small error even when the time is increased to t = 3.0, particular for the small values of h and Δt:

Table 5.4

The growth of the errors x 10^3 for a single soliton

Time	h = 0.05 Δt = 0.025				h =0.033 Δt=0.01				h = 0.01 Δt = 0.005			
	L_2		L_∞		L_2		L_∞		L_2		L_∞	
	spI	spII	spI	spII	spI	spII	spI	spII	spI	spII	spI	spII
1.25	40.2	23.4	69.5	37.7	3.2	1.3	5.1	2.2	0.09	0.09	0.24	0.23
1.50	44.0	23.6	106.1	32.9	4.0	1.6	6.2	4.3	0.10	0.10	0.26	0.28
1.75	46.2	24.4	70.9	41.9	4.6	1.8	6.4	2.9	0.11	0.11	0.30	0.29
2.00	63.2	33.7	153.4	75.9	5.2	2.1	8.2	4.2	0.12	0.12	0.32	0.32
2.25	67.0	36.3	146.4	61.1	5.6	2.3	7.5	4.3	0.13	0.13	0.35	0.35
2.50	72.7	33.6	121.4	47.3	6.5	2.3	11.	4.1	0.14	0.14	0.39	0.39
2.75	89.0	38.8	169.3	56.4	7.3	3.2	13.	7.6	0.15	0.15	0.42	0.41
3.00	99.3	48.4	200.4	91.0	7.4	2.9	10.	4.5	0.16	0.16	0.44	0.43

This Table shows us that the error computed by the scheme spII is better than that computed by the scheme spI especially for large

space and time steps. For small space and time steps they the same error of magnitude.

It is important that any numerical scheme used to solve the KdV equation satisfies at least the lower order conservation laws which they are for the present shape functions the quantities I_i $(i = 1,\ldots,4)$ defined by (2.4.8)-(2.4.11) respectively.

The quantities I_1 , I_2 , I_3 have been computed and are given in Table 5.5:

Table 5.5

The computed value I_1, I_2, and I_3 for a single soliton scheme spi

	I_1		I_2		I_3	
Time	h = .033 Δt = .01	h = .01 Δt =.005	h = .033 Δt = .01	h = .01 Δt =.005	h = .033 Δt = .01	h = .01 Δt=.005
0.0	0.144597	0.144598	0.086759	0.086759	0.046850	0.046850
0.50	0.144539	0.144599	0.086752	0.086761	0.046933	0.046852
1.00	0.144738	0.144601	0.086748	0.086762	0.046918	0.046853
1.50	0.144464	0.144602	0.086747	0.086764	0.046898	0.046855
2.00	0.144304	0.144604	0.086746	0.086765	0.046870	0.046856
2.50	0.144926	0.144605	0.086747	0.086767	0.046800	0.046858
3.00	0.144679	0.144606	0.086749	0.086768	0.046770	0.046859

From Table 5.5 we found that the computed quantities I_1 , I_2 , I_3 have been changed by less than 0.23% , 0.015% , 0.18% for h = 0.033, Δt = 0.01 respectively, and 0.006% , 0.011% , 0.020% for h = 0.01, Δt = 0.005 respectively during the computer run. They are satisfactorily constant even with relatively large values of h = 0.033 and Δt = 0.01:

Table 5.6

Computed value of I_4 for a single soliton with h = .01 Δt = .005 scheme spi

T	0.0	0.5 0	1 .0	1 .5	2.0	2.5	3.0
I_4	.024094	.024215	.024216	.024217	.024218	.024219	.02422

The computed value of the conservative quantity I_4 is changed by less than 0.52% during the computer run. We consider this very satisfactory.

A computer run on the single soliton solution with 200 nodes and 200 time steps took 22 secs of CPU time on a VAX 8650.

With example (b) we have verified that our algorithm copes adequately when two overlapping solitary waves coalesce for a brief period and then separate with their original profiles intact, but with their large and small amplitudes affected by $\approx$ 0.94% , $\approx$ 0.45% respectively and their positions interchanged. The four conservative quantities I_i $i = 1,\ldots,4$ have been listed in Table 5.7:

Table 5.7

The computed values I_1, I_2, I_3 and I_4 for two overlapping poblem (b) solitons with h = 0.01, Δt = 0.005 scheme spi

Time	I_1	I_2	I_3	I_4
0.0	0.228081	0.107062	0.053316	0.027181
0.50	0.228085	0.107064	0.053317	0.027240
1.00	0.228082	0.107066	0.053318	0.027187
1.50	0.228085	0.107068	0.053321	0.027237
2.00	0.228089	0.107071	0.053323	0.027296
2.50	0.228093	0.107073	0.053325	0.027316
3.00	0.228094	0.107074	0.053327	0.027326

Table 5.7 shows that the conservative quantities I_1 , I_2 , I_3 ,

and I_4 are changed by less than 0.006% , 0.012% , 0.021% , and 0.54% respectively during the computer run. So these quantities can be considered as constant.

From the study of problem (b) we observed that after the interaction the large and small amplitudes have been changed from the original by 0.94% and 0.45% respectively possibly because the two solitons are overlapping. So we chose two well separated solitons as the initial condition problem (b1). We found that after the interaction, their large and small amplitudes have been changed by less than 0.017% , 04% respectively, which gives a more satisfactory result.

We have also computed the first four conservative quantities for problem (b1) which are given in Table 5.8:

Table 5.8

The computed values I_1, I_2, I_3 and I_4 for two well separated solitons problem (b1) with h = 0.01, Δt = 0.005 scheme spi

Time	I_1	I_2	I_3	I_4
0.0	0.228082	0.103466	0.049864	0.024616
1.0	0.228086	0.103469	0.049868	0.024738
2.0	0.228090	0.103473	0.049870	0.024721
3.0	0.228095	0.103477	0.049872	0.024628
4.0	0.228099	0.103481	0.049876	0.024709
5.0	0.228103	0.103484	0.049879	0.024744
6.0	0.228108	0.103488	0.049882	0.024749
7.0	0.228112	0.103492	0.049885	0.024751
8.0	0.228117	0.103495	0.049888	0.024753

Table 5.8 shows us that the quantities I_i $(i = 1,\ldots,4)$ change by less than 0.016% , 0.029% , 0.049% , 0.56% respectively during the computer run. So they can be considered as conserved.

(b2) Let us study the interaction of two soliton initial conditions which follow from the analytic solution (2.3.4.1) when

t = 0.0 in the following cases:

(i) problem (b) where $\alpha_i = \sqrt{c_i/\mu}$, $d_1 = -12$, $d_2 = -12 + \Delta$

(ii) problem (b1) where $\alpha_i = \sqrt{c_i/\mu}$, $d_1 = -12$, $d_2 = -18 + \Delta$

(iii) $\alpha_1 = 4.0$, $\alpha_2 = 2.0$, $d_1 = d_2 = 0.0$, $\varepsilon = 6$, $\mu = 1$. The boundary conditions are chosen:

$$\left.\begin{aligned} u(\mp 12, t) &= 0 \\ u_{xx}(\mp 12, t) &= 0 \end{aligned}\right\} \quad \text{for } -0.5 \le t \le 0.5 \qquad (5.8.1.1)$$

Before the interaction the position of the smaller amplitude is shifted forward by Δ_2. After the interaction the phase of the soliton with larger amplitude is shifted forward by Δ_1 while the phase of the soliton with small amplitude is shifted backward by Δ_2.

The values of the L_2- and L_∞-error norms and the first four conservative quantities are given in Table 5.9:

Table 5.9

The compute values of errors x 10^3, I_1, I_2, I_3, and I_4 for double soliton poblem (b2) case (i) with h = 0.01, Δt = 0.005 scheme spi

Time	L_2 x 10^3	L_∞ x 10^3	I_1	I_2	I_3	I_4
0.0			0.228074	0.103456	0.049855	0.024610
0.50	0.040	0.100	0.228077	0.103458	0.049857	0.024661
1.00	0.045	0.113	0.228079	0.103460	0.049857	0.024615
1.50	0.057	0.121	0.228081	0.103462	0.049859	0.024652
2.00	0.083	0.208	0.228083	0.103464	0.049862	0.024705
2.50	0.106	0.280	0.228086	0.103466	0.049864	0.024728
3.00	0.109	0.278	0.228088	0.103467	0.049865	0.024735

From this Table, we find that the error is still very small even when the time reaches t = 3 and that the quantities I_i $(i = 1, \ldots, 4)$ are changed by less than 0.007% , 0.011% , 0.021% and 0.51% respectively during the computer run. We consider these

quantities as virtually constants, particularly I_1, I_2, and I_3.

After the interaction, two overlapping solitons (problem (b2) case (i)) reappeared with their amplitudes unchanged, correct to a numerical error of less than 0.8% and 0.4% respectively.

The L_2- and L_∞-error norms and the first four invariant quantities for problem (b2) case (ii) for times up to t = 8 are listed in Tanble 5.10:

Tabe 5.10

The computed values of errors x 10^3, I_1, I_2, I_3 and I_4 for double soliton problem (b2) case (ii) with h =.01, Δt =.005 scheme spi

Time	L_2x 10^3	L_∞x 10^3	I_1	I_2	I_3	I_4
0.0			0.228082	0.103456	0.049855	0.024610
1.0	0.081	0.214	0.228086	0.103459	0.049859	0.024731
2.0	0.129	0.344	0.228090	0.103463	0.049861	0.024715
3.0	0.100	0.276	0.228095	0.103467	0.049863	0.024621
4.0	0.128	0.323	0.228099	0.103471	0.049867	0.024702
5.0	0.128	0.342	0.228103	0.103474	0.049870	0.024738
6.0	0.106	0.270	0.228107	0.103478	0.049873	0.024742
7.0	0.086	0.169	0.228108	0.103482	0.049876	0.024744
8.0	0.117	0.238	0.228116	0.103485	0.049879	0.024746

Table 5.10 shows us that the errors remain satisfactorily small even when the time reaches 8. Also, the quantities I_i $(i = 1,\ldots,4)$ change by less than 0.015% , 0.029% , 0.049% and 0.56% respectively during the computer run. We conclude that these quantities are virtually constants especially I_1, I_2, and I_3. After their interaction, they reappeared with their original large and small amplitudes correct to a numerical error of less than 0.009% and 0.041% respectively.

We have also computed the L_2- and L_∞-error norms and the first four conservative quantities for double solitons with large

amplitudes for problem (b2) case (iii). They are given in Table 5.11:

Table 5.11

The computed values of errors $x10^3$, I_1, I_2, I_3, and I_4 for double soliton prblem (b2) case (iii) with h =.1, Δt =.0005 scheme spI

Time	$L_2 x\ 10^3$	$L_\infty x\ 10^3$	I_1	I_2	I_3	I_4
-0.5			11.99991	47.99998	211.2000	943.5421
-0.4	1.273	1.496	12.00002	48.00067	211.2253	956.2892
-0.3	2.276	2.526	12.00015	48.00148	211.2353	956.3051
-0.2	3.039	3.166	12.00022	48.00227	211.2366	956.0941
-0.1	3.445	3.517	12.00034	48.00298	211.2372	953.8008
0.0	2.824	3.001	12.00045	48.00352	211.2242	944.7191
0.1	3.300	3.314	12.00056	48.00454	211.2414	950.2620
0.2	3.684	3.799	12.00068	48.00548	211.2580	955.7248
0.3	3.553	3.709	12.00082	48.00636	211.2658	956.4736
0.4	3.215	3.456	12.00088	48.00714	211.2715	956.5763
0.5	2.657	2.749	12.00100	48.00793	211.2772	956.6202

Table 5.11 shows us that the errors are still small even when the time goes to t = 0.5, and the quantities I_i $(i = 1,\ldots,4)$ change by less than 0.01% , 0.017% , 0.037% and 1.387% respectively during the computer run. We consider these quantities as virtually conserved, particularly I_1 , I_2 and I_3. After the interaction, the two well separated solitons reappeared with their original large and small amplitudes changed by less than 0.95% and 0.012% respectively.

Using equation (2.3.4.8), the forward and backward phase shifts have been computed numerically for problems (b), (b1), (b2(i-ii)) and obtained as:

$$\Delta_1 \simeq 0.11, \quad \text{and} \quad \Delta_2 \simeq -0.18$$

which agree with the analytic results equation (4.5.4).

For problem (b2(iii)):

$\Delta_1 \simeq 0.50$, and $\Delta_2 \simeq -1.1$

we find that Δ_2 agrees with the analytic result given by equation (4.5.5) but Δ_1 does not agree because h = 0.1.

Similar results are given in Table 5.12 for the quantities I_i of problems (c1) and (c2). We found that each of the quantities I_i are very satisfactorily constant.

Table 5.12

The computed values of I_1, I_2 and I_3 for $u(x,0) = \exp(-x^2)$ problems (c1), (c2) for h = 0.1 Δt = 0.01 scheme (5.3.26) (spi)

Time	I_1		I_2		I_3	
	μ =.04	μ =.01	μ = .04	μ = .01	μ =.04	μ =.01
0.0	1.772454	1.772454	1.253314	1.253314	0.872929	0.985728
2.5	1.772492	1.772527	1.253344	1.253420	0.872990	0.986513
5.0	1.772488	1.772345	1.253370	1.253478	0.873026	0.986923
7.5	1.772065	1.772629	1.253392	1.253504	0.873058	0.986950
10.0	1.772340	1.772491	1.253434	1.253527	0.873083	0.986940
12.5	1.773357	1.772293	1.253442	1.253547	0.873110	0.986953

Table 5.12 shows that the quantities I_i (i = 1,2,3) are changed by less than 0.051% , 0.011% , 0.021% respectively when μ = 0.04 and 0.010% , 0.019% , 0.13% respectively when μ = 0.01 during the computer run. Hence, we can consider these quantities as conserved.

The total number of solitons which are generated from a Gaussian initial condition has been determined using equation (4.5.6) for different values of μ and we found an agreement with those given in the above Figures 5.5-5.8.

The computed values of the first four conservative quantities for problem (d) boundary conditions (ii) are given in Table 5.13 up to time t = 800:

Table 5.13

The computed values I_1, I_2, I_3 and I_4 for problem (d) boundary conditions (ii) when h = 0.4, Δt = 0.05, ε = 0.2, μ = 0.1 using scheme (5.3.26) (spi)

Time	I_1	I_2	I_3	I_4
0.0	50.00010	45.00046	42.30069	40.44194
100.0	50.00573	45.01091	42.31632	40.55037
200.0	50.01130	45.01989	42.33185	40.86950
300.0	50.01472	45.02795	42.34541	41.05238
400.0	50.01933	45.03569	42.35786	41.12419
500.0	50.02303	45.04345	42.37004	41.16071
600.0	50.02792	45.05129	42.38197	41.18266
700.0	50.03316	45.05902	42.39391	41.20121
800.0	50.03825	45.06654	42.40585	41.22260

Table 5.13 shows us that the quantities I_i $(i = 1,\ldots,4)$ are changed by less than 0.0763% , 0.147% , 0.249% , 1.94% respectively during the computer run. So they can be considered as conserved even with this very large time and large space step.

5.8.2 Simulations Using Scheme (5.4.2):

All the simulations undertaken using scheme (5.3.26) have been repeated using scheme (5.4.2).

The discrete L_2-error norm for problem (a) using scheme (5.4.2) is given in Table 5.14:

Table 5.14

The growth of the discrete L_2-error norm x 10^3 for single soliton problem (a) Scheme (5.4.2)

Time	Zabusky-Kruskal [27]	Hopscotch [26]	Petrov-Galerkin [36]	Modified P-G [36]	Galerkin cubic spline
$\Delta x = 0.05$		$\Delta t = 0.025$	$h = 0.05$	$\Delta t = 0.025$	
0.25	34.64	61.21	81.39	52.15	13.15
0.50	122.68	122.41	102.54	64.90	21.79
0.75	210.44	181.35	125.84	89.01	25.55
1.00	298.19	228.10	150.57	107.20	29.31
			$h = 0.033$	$\Delta t = 0.01$	
0.25			31.18	5.94	1.34
0.5			43.35	7.56	1.82
0.75			56.21	8.70	2.30
1.00			74.08	9.49	2.69
$\Delta x = 0.01$		$\Delta t = 0.0005$	$h = 0.01$	$\Delta t = 0.005$	
0.25	5.94	3.79	4.46	0.21	0.02
0.50	13.17	9.28	7.01	0.38	0.05
0.75	21.08	14.14	10.08	0.57	0.07
1.00	28.66	18.72	13.26	0.74	0.09

If we compare the magnitudes of the L_2-error norm given in Tables 5.2 and 5.14 we see that in most cases they are the same whether we use scheme (5.3.26) or (5.4.2) . Only when h and Δt have smaller values of $h = 0.01$ and $\Delta t = 0.005$ is there a difference. In that case, scheme (5.3.26) leads to slightly smaller errors.

Table 5.15 gives the error for a single soliton problem (a) and we observed that this error has the same magnitudes as the error given in Table 5.2 . It has been found that for $h = 0.01$ and

$\Delta t = 0.005$ there is a small increase in the errors given in Tables 5.14 and 5.15 compared with those given in Tables 5.2 and 5.4:

Table 5.15

L_2-error x 10^3 of a single soliton problem (a) for the scheme(5.4.2)

T	1.25	1.50	1.75	2.00	2.25	2.50	2.75	3.00
h = 0.05 Δt = 0.025	40.37	44.69	47.01	63.11	70.15	78.90	98.48	110.27
h = 0.033 Δt = 0.01	3.25	4.01	4.57	5.22	5.64	6.55	7.32	7.49
h = 0.01 Δt = 0.005	0.11	0.13	0.15	0.18	0.20	0.23	0.26	0.28

The quantities I_1 , I_2 , I_3 have also been determined and are given in Table 5.16:

Table 5.16

The computed value I_1, I_2, and I_3 for a single soliton problem (a) using scheme (5.4.2)

	I_1		I_2		I_3	
Time	h = .033 Δt = .01	h = .01 Δt =.005	h = .033 Δt = .01	h = .01 Δt =.005	h = .033 Δt = .01	h = .01 Δt=.005
0.0	0.144597	0.144598	0.086759	0.086759	0.046850	0.046850
0.50	0.144539	0.144598	0.086753	0.086759	0.046933	0.046851
1.00	0.144739	0.144598	0.086749	0.086759	0.046919	0.046851
1.50	0.144465	0.144599	0.086748	0.086760	0.046899	0.046851
2.00	0.144305	0.144599	0.086748	0.086760	0.046871	0.046851
2.50	0.144927	0.144599	0.086749	0.086759	0.046802	0.046851
3.00	0.144680	0.144599	0.086751	0.086759	0.046772	0.046851

From Table 5.16 we found that the computed quantities I_1 , I_2 , I_3 have been changed by less than 0.21% , 0.013% , 0.18% for h = 0.033, Δt = 0.01 respectively and 0.0007% , 0.0012% , 0.0022% for h = 0.01, Δt = 0.005 respectively during the computer run.

This conservation is even better than the very good results quoted in Table 5.5:

Table 5.17

Computed value of I_4 for a single soliton with h = .01 Δt = .005

T	0.0	0.5	1 .0	1 .5	2.0	2.5	3.0
I_4	.024094	.024214	.024214	.024215	.024215	.024215	.024214

The value of the quantity I_4 changed by less than 0.50% during the computer run compares well with the value of 0.52% found for Table 5.6.

The four conservative quantities I_i ($i = 1,\ldots,4$) for problem (b) have been computed and are listed in Table 5.18:

Table 5.18

The computed values I_1, I_2, I_3 and I_4 for two overlapping solitons problem (b) with h = 0.01, Δt = 0.005 using scheme (5.4.2)

Time	I_1	I_2	I_3	I_4
0.0	0.228081	0.107062	0.053316	0.027181
0.50	0.228083	0.107062	0.053317	0.027239
1.00	0.228079	0.107062	0.053315	0.027185
1.50	0.228080	0.107063	0.053316	0.027234
2.00	0.228081	0.107063	0.053317	0.027291
2.50	0.228083	0.107063	0.053318	0.027310
3.00	0.228083	0.107063	0.053318	0.027320

Table 5.18 shows that the conservative quantities I_1 , I_2 , I_3 , and I_4 changed by less than 0.0009% , 0.001% , 0.004% , and 0.52% respectively during the computer run. So these quantities can be considered as constant. Moreover, the quantities I_1 , I_2 , I_3 are constant to 5 decimal places. This conservative quantities are better than the very good results obtained in Table 5.7. After

the interaction of the two overlapping solitons, their large and small amplitudes are changed from their original values by less than 0.94% and 0.46% respectively.

We have also computed the first four conservative quantities for problem (b1), these are given in Table 5.19:

Table 5.19

The computed values I_1, I_2, I_3 and I_4 for a two well separated solitons with h = 0.01, Δt = 0.005 using scheme (5.4.2)

Time	I_1	I_2	I_3	I_4
0.0	0.228082	0.103466	0.049864	0.024616
1.0	0.228082	0.103466	0.049865	0.024736
2.0	0.228083	0.103466	0.049865	0.024718
3.0	0.228084	0.103467	0.049864	0.024622
4.0	0.228084	0.103467	0.049865	0.024701
5.0	0.228084	0.103467	0.049866	0.024735
6.0	0.228085	0.103467	0.049866	0.024737
7.0	0.228086	0.103467	0.049866	0.024737
8.0	0.228086	0.103467	0.049866	0.024737

Table 5.19 shows us that the quantities I_i $(i = 1,\ldots,4)$ changed by less than 0.0018% , 0.001% , 0.0041% , 0.492% respectively during the computer run. So they can be considered as conserved. Moreover, the quantities I_1 , I_2 , I_3 are constant to 5 decimal places. This conservation is even better than the very good results quoted in Table 5.8. After the interaction of the two well separated solitons, their large and small amplitudes have changed by very small amounts, less than 0.009% and 0.024% respectively. So we can take it that after the interaction the amplitudes are virtually unchanged.

We have calculated the L_2- and L_∞-error norms and the first four conservative quantities for the two overlapping solitons

(problem (b2) case (i)). These are recorded in Table 5.20:

Table 5.20

The computed values of the error x 10^3, I_1, I_2, I_3 and I_4 for double soliton problem (b2) case (i) with h = 0.01, Δt = 0.005

Time	$L_2 x10^3$	$L_\infty x10^3$	I_1	I_2	I_3	I_4
0.0			0.228074	0.103456	0.049855	0.024610
0.50	0.042	0.103	0.228075	0.103456	0.049855	0.024660
1.00	0.049	0.111	0.228075	0.103456	0.049855	0.024613
1.50	0.070	0.168	0.228075	0.103457	0.049856	0.024649
2.00	0.117	0.291	0.228076	0.103457	0.049857	0.024702
2.50	0.167	0.442	0.228077	0.103457	0.049857	0.024723
3.00	0.198	0.519	0.228076	0.103457	0.049857	0.024729

In Table 5.20 we give the values of the errors and I_1 , I_2 , I_3 and I_4. The error is still small even when the time is increased up to t = 3.0. The change in the quantities I_1 during the computer run are less than 0.0014% , 0.001% , 0.0041% and 0.484% respectively. Therefore, we conclude that these quantities may be considered as constants. Furthermore, the quantities I_1 , I_2 , I_3 are constant to 5 decimal places. The conservation of these quantities is even better than the very good results obtained in Table 5.9, with the error having the same magnitude. After the interaction of the two overlapping solitons problem (b2) case (i) their large and small amplitudes have changed by only small amounts of about 0.81% and 0.37% respectively.

The first four conservative quantities and the L_2- and L_∞-error norms have been determined for the two well separated solitons (problem (b2) case (ii)) and are listed in Table 5.21:

Table 5.21

The computed values of the errors x 10^3, I_1, I_2, I_3 and I_4 for double soliton problem (b2) case (ii) with h = 0.01, Δt = 0.005

Time	$L_2 x10^3$	$L_\infty x10^3$	I_1	I_2	I_3	I_4
0.0			0.228082	0.103456	0.049855	0.024610
1.0	0.093	0.247	0.228082	0.103456	0.049856	0.024711
2.0	0.171	0.445	0.228083	0.103456	0.049856	0.024711
3.0	0.148	0.367	0.228084	0.103457	0.049855	0.024616
4.0	0.269	0.707	0.228084	0.103457	0.049857	0.024695
5.0	0.374	0.987	0.228085	0.103458	0.049857	0.024729
6.0	0.455	1.198	0.228085	0.103458	0.049857	0.024731
7.0	0.534	1.404	0.228086	0.103457	0.049857	0.024731
8.0	0.615	1.621	0.228086	0.103457	0.049857	0.024731

Table 5.21 shows us that the error is still small even when the time reaches t = 8. The quantities I_i $(i = 1,\ldots,4)$ change during the computer run by less than 0.0018% , 0.002% , 0.0041% and 0.492% respectively making them virtually constants. Moreover, the first three quantities are constant to 5 decimal places. After the interaction of two well separated solitons in problem (b2) case (ii), their large and small amplitudes have changed by less than 0.018% and 0.024% respectively so that we can say they are virtually unchanged. We find that conservation of the first four quantities is even better than the very good results obtained in Table 5.10 with, the error having the same magnitude.

The first four conservative quantities and the L_2- and L_∞-error norms for two well separated solitons with large amplitudes (problem (b2) case (iii)) have been computed and are given in Table 5.22:

Table 5.22

The computed values of the errors x 10^3 , and the conservative quantities for double solitons problem (b2) case (iii) with h = 0.1 , Δt = 0.0005

Time	$L_2 x10^3$	$L_\infty x10^3$	I_1	I_2	I_3	I_4
-0.5			11.99991	47.99998	211.2000	943.5421
-0.4	1.273	1.494	12.00002	48.00070	211.2255	956.2907
-0.3	2.279	2.530	12.00015	48.00146	211.2307	956.3029
-0.2	3.063	3.202	12.00022	48.00218	211.2358	956.0893
-0.1	3.497	3.568	12.00033	48.00287	211.2365	953.7959
0.0	2.885	3.053	12.00044	48.00336	211.2229	944.7109
0.1	3.406	3.430	12.00056	48.00444	211.2406	950.2565
0.2	3.847	3.987	12.00066	48.00534	211.2569	955.7175
0.3	3.776	3.930	12.00080	48.00615	211.2641	956.4627
0.4	3.499	3.758	12.00086	48.00694	211.2700	956.5660
0.5	2.995	3.099	12.00097	48.00769	211.2754	956.6082

We found that the error is still small, and quantities I_i $(i = 1,\ldots,4)$ have changed during the computer run by less than 0.009% , 0.0161% , 0.036% and 1.385% respectively so that they are very satisfactorily constants particularly the first three invariant quantities. After the interaction, the large and small amplitudes of the two well separated solitons in problem (b2) case (iii) have changed by less than 0.95% and 0.013% respectively. We notice that the conservation of the first four invariant quantities is even better than the very satisfactory results quoted in Table 5.11 while the error has the same magnitude.

The last results are given in Table 5.23 for the first three conservative quantities of problems (c1) and (c2). We found that each of the quantities I_i are satisfactorily constant:

Table 5.23

The computed values of I_1, I_2 and I_3 for $u(x,0) = \exp(-x^2)$

problems (c1) (c2) for h = 0.1 Δt = 0.01

Time	I_1		I_2		I_3	
	μ =.04	μ =.01	μ = .04	μ = .01	μ =.04	μ =.01
0.0	1.772454	1.772454	1.253314	1.253314	0.872929	0.985728
2.5	1.772491	1.772535	1.253344	1.253433	0.872990	0.986529
5.0	1.772488	1.772359	1.253371	1.253501	0.873027	0.986951
7.5	1.772066	1.772654	1.253395	1.253538	0.873060	0.986994
10.0	1.772339	1.772524	1.253435	1.253572	0.873084	0.986998
12.5	1.773353	1.772333	1.253442	1.253600	0.873110	0.987022

Table 5.23 shows that the quantities I_i (i = 1,2,3) are changed by less than 0.051% , 0.011% , 0.021% respectively when μ = 0.04 and by less than 0.012% , 0.023% , 0.13% respectively when μ = 0.01 during the computer run. Therefore, we can consider that these quantities are conserved. These quantities have the same magnitude as the results given in Table 5.12 .

The first four conservative quantities for problem (d) using boundary conditions (ii) are listed in Table 5.24:

Table 5.24

The computed values I_1, I_2, I_3 and I_4 for problem (d) with boundary conditions (ii) h = 0.4, Δt = 0.05 using scheme (5.4.2)

Time	I_1	I_2	I_3	I_4
0.0	50.00010	45.00046	42.30069	40.44194
100.0	50.00229	45.00484	42.30769	40.53928
200.0	50.00456	45.00779	42.31437	40.84664
300.0	50.00464	45.00981	42.31874	41.01703
400.0	50.00592	45.01152	42.32178	41.07610
500.0	50.00625	45.01318	42.32444	41.09961
600.0	50.00786	45.01502	42.32704	41.10881
700.0	50.00979	45.01677	42.32954	41.11444
800.0	50.01158	45.01818	42.33188	41.12217

Table 5.24 shows us that the quantities I_i $(i = 1,\ldots,4)$ are changed by less than 0.023% , 0.040% , 0.074% , 1.682% respectively during the very long computer run. So they are virtually conserved, particularly the qantities I_1 , I_2 , I_3. This conservation is even better than the very good results quoted in Table **5.13**. The agreement between the analytic velocity $c_a \simeq 0.1308$ and the observed velocity $c_n \simeq 0.128$ was very satisfactory even with this large space step (h = 0.4).

In summary it can be stated that Galerkin's method with a cubic spline interpolation polynomial as trial and test functions is a suitable algorithm for determining the solution of the **KdV** equation for runs of long duration.

CHAPTER 6

QUADRATIC SPLINE INTERPOLATION FUNCTIONS

6.1 Introduction:

In the previous chapter we set up a finite element solution to the **KdV** equation using cubic splines as trial functions. That choice was made with the knowledge of the special properties that such splines possess. However, there are advantages to be gained by choosing lower order polynomials if possible. One such benefit is the reduction of the order of the stiffness matrix. The only lower order fuctions, that we are aware of, which possess the required first order continuity, are quadratic splines.

In this chapter we will investigate the finite element approach using Galerkin's method with quadratic spline interpolation functions.

6.2 The Governing Equation:

Consider the **KdV** equation:

$$u_t + \varepsilon\, u\, u_x + \mu\, u_{xxx} = 0 \qquad a \le x \le b \qquad (6.2.1)$$

where; ε , μ are positive parameters.

The boundary conditions will be chosen from:

$$\left.\begin{aligned} u(a,t) &= \beta_1 \\ u(b,t) &= \beta_2 \\ u_x(a,t) &= u_x(b,t) = 0 \\ u_{xx}(a,t) &= u_{xx}(b,t) = 0 \end{aligned}\right\} \quad \text{for all } t > 0 \qquad (6.2.2)$$

Let us apply the Galerkin technique to equation (6.2.1) with weight functions v(x) . Integrating by parts leads to the equation:

$$\int_a^b v\,(u_t + \varepsilon\, u\, u_x)dx - \int_a^b \mu\, v_x u_{xx} dx = -\left[\mu\, v\, u_{xx}\right]_a^b \qquad (6.2.3)$$

and using the boundary conditions (6.2.2) equation (6.2.3) reduces to:

$$\int_a^b v\,(u_t + \varepsilon\, u\, u_x)dx - \int_a^b \mu\, v_x u_{xx} dx = 0\ . \qquad (6.2.4)$$

The presence of the second spatial derivative within the integrand means that the interpolation functions and their first derivatives must be continuous throughout the region. Quadratic **B**-splines as trial functions satisfy this requirement.

6.3 The Finite Element Solution [82,84,85]:

In this section we approximate the solution $u(x,t)$ using quadratic **B**-spline interpolation functions.

Partition the region [a,b] into N finite elements of equal length h by knots x_i such that

$a = x_0 \le x_1 \ldots \le x_N = b$ and let $\phi_i(x)$ be those quadratic **B**-splines with knots at x_i . Then the splines $\{\phi_{-1}, \phi_0, \ldots, \phi_{N-1}, \phi_N\}$ form a basis for functions defined over [a,b]. Our aim is to find an approximate solution $u_N(x,t)$ to the solution $u(x,t)$ which can be expressed in terms of quadratic spline trial functions:

$$u_N(x,t) = \delta_{-1}(t)\,\phi_{-1}(x) + \delta_0(t)\,\phi_0(x) + \ldots + \delta_N(t)\,\phi_N(x)$$

$$= \sum_{m=-1}^{N} \delta_m(t)\,\phi_m(x) \qquad (6.3.1)$$

where; ϕ_m are quadratic spline functions and δ_m are time dependent quantities to be determined from the boundary conditions (6.2.2) and from conditions based on equation (6.2.4).

Quadratic B-splines ϕ_m with the required properties are defined by [71,72,73]:

$$\phi_m(x) = \frac{1}{h^2}\begin{cases} (x_{m+2} - x)^2 - 3(x_{m+1} - x)^2 + 3(x_m - x)^2 & [x_{m-1}, x_m] \\ (x_{m+2} - x)^2 - 3(x_{m+1} - x)^2 & [x_m, x_{m+1}] \\ (x_{m+2} - x)^2 & [x_{m+1}, x_{m+2}] \\ 0 & \text{otherwise} \end{cases}$$

$$m = -1, 0, \ldots, N \qquad (6.3.2)$$

where; $h = (x_{m+1} - x_m)$ for all m, assuming that all intervals $[x_m, x_{m+1}]$ are of equal size.

The quadratic spline $\phi_m(x)$ and its first derivative vanish outside the interval $[x_{m-1}, x_{m+2}]$. In Table 6.1 we list the values of $\phi_m(x)$ and its derivative $\phi_m'(x)$ at the knots:

Table 6.1

x	x_{m-1}	x_m	x_{m+1}	x_{m+2}
$\phi_m(x)$	0	1	1	0
$\phi_m'(x)$	0	$\frac{2}{h}$	$-\frac{2}{h}$	0

We shall identify the intervals $[x_m, x_{m+1}]$ with finite elements with nodes at the knots x_m, x_{m+1}. Discussing only the internal elements from equation (6.3.2) we see that each spline covers 3 intervals so that 3 splines ϕ_{m-1}, ϕ_m, ϕ_{m+1} cover each finite element $[x_m, x_{m+1}]$. All other splines are zero in this region. These 3 splines act as "shape" functions for the element (see Figure 6.1). Using equation (6.3.1) and Table 6.1 the nodal values $\mathbf{u}_m$ can be expressed in terms of the parameters δ_m by:

$$\left.\begin{aligned} u_m &= u(x_m) = \delta_{m-1} + \delta_m \\ u_{m+1} &= u(x_{m+1}) = \delta_m + \delta_{m+1} \end{aligned}\right\} \qquad (6.3.3)$$

and the variation of u over the element $[x_m, x_{m+1}]$ is defined by:

$$u = \delta_{m-1}\phi_{m-1} + \delta_m\phi_m + \delta_{m+1}\phi_{m+1} = \sum_{j=m-1}^{m+1} \delta_j \phi_j \qquad (6.3.4)$$

where; δ_{m-1} , δ_m , δ_{m+1} act as element parameters with the element "shape" functions ϕ_{m-1} , ϕ_m , ϕ_{m+1} . Defining a local coordinate system ξ for the finite element $[x_m, x_{m+1}]$ by $\xi = x - x_m$, where $0 \le \xi \le h$, we obtain for the shape functions expressions that are independent of the elements position:

$$\begin{matrix} \phi_{m-1} \\ \phi_m \\ \phi_{m+1} \end{matrix} = \frac{1}{h^2} \begin{cases} (h-\xi)^2 \\ h^2 + 2h\xi - 2\xi^2 \\ \xi^2 \end{cases} \qquad 0 \le \xi \le h \qquad (6.3.6)$$

These " shape " functions are the same for every element (see Figure 6.1). An element contributes to equation (6.2.4) through the following integral which can be written in terms of the element parameters δ^e as:

$$\int_{x_m}^{x_{m+1}} \left[v\,(u_t + \varepsilon\, u\, u_x) - \mu\, v_x\, u_{xx} \right] dx = 0 \qquad (6.3.7)$$

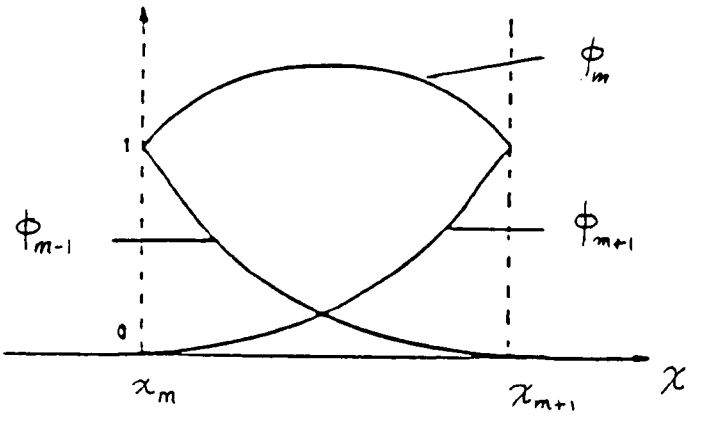

Figure 6.1 Quadratic spline shape function for a typical element.

So identifying the weight functions v with quadratic splines ϕ_j and using (6.3.5) and (6.3.6) we obtain:

$$\sum_{j=m-1}^{m+1} \left[\int_{x_m}^{x_{m+1}} \phi_i \, \phi_j dx \right] \dot{\delta}_j^e + \varepsilon \sum_{j=m-1}^{m+1} \sum_{k=m-1}^{m+1} \left[\int_{x_m}^{x_{m+1}} \phi_i \, \phi_j \phi_k' dx \right] \delta_k^e \, \delta_j^e$$

$$- \mu \sum_{j=m-1}^{m+1} \left[\int_{x_m}^{x_{m+1}} \phi_i' \, \phi_j'' \, dx \right] \delta_j^e \qquad (6.3.8)$$

where $\underset{\sim}{\delta}^e = (\delta_{m-1} , \delta_m , \delta_{m+1})^T$ (6.3.9)

are the relevant element parameters.

Equation (6.3.8) can be written in matrix form as:

$$A^e \, \dot{\underset{\sim}{\delta}}^e + \varepsilon \, \underset{\sim}{\delta}^{eT} L^e \, \underset{\sim}{\delta}^e - \mu \, C^e \underset{\sim}{\delta}^e \qquad (6.3.10)$$

where $\underset{\sim}{\delta}^e$ is given by (6.3.9) and the element matrices are given by the integral formulas:

$$A^e_{ij} = \int_{x_m}^{x_{m+1}} \phi_i \, \phi_j \, dx \qquad (6.3.11)$$

$$C^e_{ij} = \int_{x_m}^{x_{m+1}} \phi'_i \, \phi''_j \, dx \qquad (6.3.12)$$

and

$$L^e_{ijk} = \int_{x_m}^{x_{m+1}} \phi_i \, \phi_j \, \phi'_k \, dx \qquad (6.3.13)$$

where; i , j , k take only the values $m-1$, m , $m+1$ for the typical element $[x_m , x_{m+1}]$. The matrices A^e , C^e are thus $3x3$ and the matrix L^e $3x3x3$. It is convenient to associate a $3x3$ matrix B^e with L^e defined by:

$$B^e_{ij} = \sum_{k=m-1}^{m+1} L^e_{ijk} \, \delta^e_k \qquad (6.3.14)$$

which also depends on the parameters δ^e_k will be used in the following theoretical discusions.

The element matrices A^e , C^e , L^e are independent of the parameters δ^e_k and can be determined algebraically from equations (6.3.11)-(6.3.13) using REDUCE [38] as:

$$A^e = \frac{h}{30}\begin{bmatrix} 6 & 13 & 1 \\ 13 & 54 & 13 \\ 1 & 13 & 6 \end{bmatrix} \qquad (6.3.15)$$

$$C^e = \frac{2}{h^2}\begin{bmatrix} -1 & 2 & -1 \\ 0 & 0 & 0 \\ 1 & -2 & 1 \end{bmatrix} \qquad (6.3.16)$$

and

$$B^e_{m-1,m-1} = -\frac{1}{3}\delta_{m-1} + \frac{4}{15}\delta_m + \frac{1}{15}\delta_{m+1}$$

$$B^e_{m-1,m} = B^e_{m,m-1} = -\frac{19}{30}\delta_{m-1} + \frac{2}{5}\delta_m + \frac{7}{30}\delta_{m+1}$$

$$B^e_{m-1,m+1} = B^e_{m+1,m-1} = -\frac{1}{30}\delta_{m-1} + \frac{1}{30}\delta_{m+1}$$

$$B^e_{m,m} = -\frac{9}{5}\delta_{m-1} + \frac{9}{5}\delta_{m+1}$$

$$B^e_{m,m+1} = B^e_{m+1,m} = -\frac{7}{30}\delta_{m-1} - \frac{2}{5}\delta_m + \frac{19}{30}\delta_{m+1}$$

$$B^e_{m+1,m+1} = -\frac{1}{15}\delta_{m-1} - \frac{4}{15}\delta_m + \frac{1}{3}\delta_{m+1} \qquad (6.3.17)$$

If we partition the region [a,b] into three elements of equal length h and combine together the contributions from each element we obtain:

$$A\,\dot{\underset{\sim}{\delta}} + \varepsilon\, B(\underset{\sim}{\delta})\,\underset{\sim}{\delta} - \mu\,\underset{\sim}{\delta} = 0 \qquad (6.3.18)$$

where:

$$\underset{\sim}{\delta} = (\delta_{-1}, \delta_0, \delta_1, \delta_2, \delta_3)^T \qquad (6.3.19)$$

The matrices A, $B(\underset{\sim}{\delta})$, C, assembled from the element matrices A^e, $B^e(\underset{\sim}{\delta})$, C^e have the form:

$$A = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & & \\ a_{21}^{(1)} & a_{11}^{*(2)} & a_{12}^{*(2)} & a_{13}^{(2)} & \\ a_{31}^{(1)} & a_{21}^{*(2)} & a_{11}^{*(3)} & a_{12}^{*(3)} & a_{13}^{(3)} \\ & a_{31}^{(2)} & a_{21}^{*(3)} & a_{22}^{*(3)} & a_{23}^{(3)} \\ & & a_{31}^{(3)} & a_{32}^{(3)} & a_{33}^{(3)} \end{bmatrix} \qquad (6.3.20)$$

where

$$a_{11}^{*(2)} = a_{22}^{(1)} + a_{11}^{(2)}, \qquad a_{12}^{*(2)} = a_{23}^{(1)} + a_{12}^{(2)}$$

$$a_{21}^{*(2)} = a_{32}^{(1)} + a_{21}^{(2)}, \qquad a_{11}^{*(3)} = a_{11}^{(3)} + a_{22}^{(2)} + a_{33}^{(1)}$$

$$a_{12}^{*(3)} = a_{23}^{(2)} + a_{12}^{(3)}, \qquad a_{21}^{*(3)} = a_{32}^{(2)} + a_{21}^{(3)}$$

$$a_{22}^{*(3)} = a_{33}^{(2)} + a_{22}^{(3)}$$

Matrices $B(\delta)$ and C can be expressed in the same way (6.3.20).

Generally, dividing the region [a,b] into N elements of equal length h and combining contributions from all elements and following the procedure for three elements produces the matrix equation:

$$A\,\dot{\underset{\sim}{\delta}} + \varepsilon\, B(\delta)\,\underset{\sim}{\delta} - \mu\, C\,\underset{\sim}{\delta} = 0 \qquad (6.3.21)$$

where $\underset{\sim}{\delta} = (\delta_{-1}, \delta_0, \delta_1, \ldots, \delta_N)^T$. (6.3.22)

$\underset{\sim}{\delta}$ are element parameters to be determined and A, $B(\delta)$, C are matrices assembled from the element matrices A^e, $B^e(\delta)$, C^e. The matrices A, B, C are $(N+2)\times(N+2)$ 5-banded matrices.

Let $\underset{\sim}{\delta}$ be linearly interpolated between two levels n and n+1 by:

$$\underset{\sim}{\delta} = (1 - \theta)\underset{\sim}{\delta}^{n} + \theta\underset{\sim}{\delta}^{n+1} \qquad (6.3.23)$$

where $t = (n+\theta)\Delta t$ and $0 \le \theta \le 1$. Then the time derivative of $\underset{\sim}{\delta}$ is:

$$\dot{\underset{\sim}{\delta}} = \frac{\underset{\sim}{\delta}^{n+1} - \underset{\sim}{\delta}^{n}}{\Delta t} . \qquad (6.3.24)$$

Using the definitions (6.3.23) and (6.3.24), equation (6.3.21) becomes:

$$\left[A + \theta\Delta t(\varepsilon B(\underset{\sim}{\delta}^{n}) - \mu C)\right]\underset{\sim}{\delta}^{n+1} = \left[A - (1-\theta)\Delta t(\varepsilon B(\underset{\sim}{\delta}^{n}) - \mu C)\right]\underset{\sim}{\delta}^{n} \qquad (6.3.25)$$

Giving the parameter θ the values 0, $\frac{1}{2}$, and 1 produces forward, Crank-Nicolson and backward difference schemes respectively.

If we let $\theta = \frac{1}{2}$ equation (6.3.25) becomes:

$$\left[A + \frac{\Delta t}{2}(\varepsilon B(\underset{\sim}{\delta}^{n}) - \mu C)\right]\underset{\sim}{\delta}^{n+1} = \left[A - \frac{\Delta t}{2}(\varepsilon B(\underset{\sim}{\delta}^{n}) - \mu C)\right]\underset{\sim}{\delta}^{n} \qquad (6.3.26)$$

Since the matrix $B(\underset{\sim}{\delta})$ depends on $\underset{\sim}{\delta}$, the matrix equation (6.3.26) is nonlinear. We handle this problem not by solving equation (6.3.26) directly but by setting up an equivalent system and then solving that [56,70]. Such a system is:

$$\left[A + \frac{\Delta t}{2}(\varepsilon B(\underset{\sim}{\delta}^{n}) - \mu C)\right]\hat{\underset{\sim}{\delta}}^{n+1} = \left[A - \frac{\Delta t}{2}(\varepsilon B(\underset{\sim}{\delta}^{n}) - \mu C)\right]\underset{\sim}{\delta}^{n} \qquad (6.3.27a)$$

and

$$\left[A + \frac{\Delta t}{2}(\varepsilon B\left(\frac{\hat{\underset{\sim}{\delta}}^{n+1} + \underset{\sim}{\delta}^{n}}{2}\right) - \mu C)\right]\underset{\sim}{\delta}^{n+1} = \left[A - \frac{\Delta t}{2}(\varepsilon B\left(\frac{\hat{\underset{\sim}{\delta}}^{n+1} + \underset{\sim}{\delta}^{n}}{2}\right) - \mu C)\right]\underset{\sim}{\delta}^{n} \qquad (6.3.27b)$$

The predictor (6.3.27a) gives a first approximation $\hat{\underset{\sim}{\delta}}^{n+1}$ then the corrector (6.3.27b) is used iteratively to improve the approximation.

Before solving the system (6.3.27) we must apply the boundary conditions which are chosen to be

$$u(a,t) = \beta_1 ,$$

$$u(b,t) = \beta_2 ,$$

$$u_x(a,t) = u_x(b,t) = 0 ,$$

and from the Table 6.1 these conditions become:

$$\left.\begin{aligned} \delta_{-1} + \delta_0 &= \beta_1 \\ \delta_{-1} - \delta_0 &= 0 \\ \delta_{N-1} + \delta_N &= \beta_2 \\ \delta_{N-1} - \delta_N &= 0 \end{aligned}\right\} \qquad (6.3.28)$$

By eliminating δ_{-1} , δ_0 , δ_{N-1} , δ_N from equation (6.3.27) we obtain a recurrence relationship for $(\delta_1^n , \delta_2^n , \ldots , \delta_{N-2}^n)^T$.

Equation (6.3.27) contains $(N-2)\times(N-2)$ 5-banded matrices, so to solve these equations we store the matrices in rectangular form $(N-2)\times 5$ and then use a penta-diagonal algorithm (see Appendix A2). The boundary parameters δ_{-1} , δ_0 , δ_{N-1} , δ_N can be computed at each time step from equations (6.3.28).

To start the iterative procedure (6.3.27), a starting vector $\underset{\sim}{\delta}^0$ must be determined from the initial condition on $u(x,t)$. Once the parameters $\underset{\sim}{\delta}$ have been found at a specified time then we can compute the solution at the knots from the formula:

$$u(x_i,t) = \delta_{i-1} + \delta_i \qquad (6.3.29)$$
$$i = 0 , 1 , \ldots , N$$

6.4 The Initial State:

The starting value $\underset{\sim}{\delta}^0$ is determined from the initial condition $u(x,0)$ on $u(x,t)$ which can be calculated by interpolating $u(x,0)$ using quadratic splines. We firstly rewrite equation (6.3.1) for the initial condition as:

$$u_N(x,0) = \sum_{i=-1}^{N} \delta_i^0 \, \phi_i(x) \quad , \qquad (6.4.1)$$

where; the δ_i^0 are unknowns to be determined. To do so we require $u_N(x,0)$ to satisfy the following two conditions:

(a) It shall agree with the initial condition $u(x,0)$ at the knots x_i , $i = 0, 1, \ldots, N$; leading to $N+1$ conditions,

(b) Its first derivative shall agree with that of the exact initial condition at x_0 i.e. $u'(x_0,0) = 0$.

These two conditions (a) and (b) can be written as:

$$\left.\begin{aligned} u'_N(x_0,0) &= 0 \\ u_N(x_i,0) &= u(x_i,0) \qquad 0 \le i \le N \end{aligned}\right\} \qquad (6.4.2)$$

Using Table 6.1 the system (6.4.2) can be written in a matrix equation of the form:

$$A \underset{\sim}{\delta}^0 = \underset{\sim}{b} \qquad (6.4.3)$$

where:

$$A = \begin{bmatrix} 1 & -1 & & & & & & & & \\ 1 & 1 & & & & & & & & \\ & 1 & 1 & & & & & & & \\ & & 1 & 1 & & & & & & \\ & & & \ddots & & & & & & \\ & & \cdots & \vdots & \ddots & & & & & \\ & & & & & 1 & 1 & & & \\ & & & & & & 1 & 1 & & \\ & & & & & & & 1 & 1 \end{bmatrix} \qquad (6.4.4)$$

$$\underset{\sim}{\delta}^0 = \left[\delta_{-1}^0 , \delta_0^0 , \ldots , \delta_N^0 \right]^T \qquad (6.4.5)$$

and

$$\underset{\sim}{b} = \left[0 , u(x_0,0) , u(x_1,0) , \ldots , u(x_N,0) \right]^T \qquad (6.4.6)$$

This system of equations can be solved directly using the following technique:

Let $u_j = u(x_j , 0)$, $j = 0, 1, \ldots, N$

$$\delta_{-1} = \frac{u_0}{2} \quad , \quad \delta_0 = \frac{u_0}{2}$$

for j = 1 to N do

$$\delta_j = u_j - \delta_{j-1}$$

So the initial vector $\underset{\sim}{\delta}^0$ is computed by this algorithm.

6.5 Stability Analysis:

The stability analysis of nonlinear partial differential equations is not an easy task to undertake. Most researchers cope with the problem by linearising the partial differential equation.

Our stability analysis will be based on the von Neuman theory in which the growth factor of a typical Fourier mode defined as:

$$\delta_j^n = \hat{\delta}^n e^{ijkh} \qquad (6.5.1)$$

where; k is the mode number and h is the element size, is determined for a linearisation of the numerical scheme.

To linearise the KdV equation (6.2.1) assume that the quantity u in the nonlinear term $u u_x$ is locally constant. This is equivalent to assuming that all the δ_j^n are equal to a local constant d, so that the matrix B in equation (6.3.25) is determined from (6.3.7) to be:

$$B = d \begin{bmatrix} -1 & -\frac{8}{3} & -\frac{1}{3} \\ \frac{2}{3} & 0 & -\frac{2}{3} \\ \frac{1}{3} & \frac{8}{3} & 1 \end{bmatrix}^T \qquad (6.5.2)$$

A typical member of equation (6.5.3), using the linearised matrix B (6.5.2), is given by:

$$\alpha_1\delta_{i-2}^{n+1} + \alpha_2\delta_{i-1}^{n+1} + \alpha_3\delta_i^{n+1} + \alpha_4\delta_{i+1}^{n+1} + \alpha_5\delta_{i+2}^{n+1} =$$

$$\alpha_5\delta_{i-2}^{n} + \alpha_4\delta_{i-1}^{n} + \alpha_3\delta_i^{n} + \alpha_2\delta_{i+1}^{n} + \alpha_1\delta_{i+2}^{n} \qquad (6.5.3)$$

where:

$$\alpha_1 = \alpha - \beta - \gamma \quad , \quad \alpha_2 = 26\alpha + 10\beta + 2\gamma \ ,$$

$$\alpha_3 = 66\alpha \quad , \quad \alpha_4 = 26\alpha + 10\beta - 2\gamma \ ,$$

$$\alpha = \alpha + \beta + \gamma ,$$

$$\alpha = \frac{h}{30} \quad , \quad \beta = \frac{\varepsilon d \Delta t}{6} \quad , \quad \gamma = \frac{\mu \Delta t}{h^2} . \qquad (6.5.4)$$

The Fourier method (6.5.1) applied to equation (6.5.3) leads to:

$$\underset{\sim}{\hat{\delta}}^{n+1}\left[\alpha_1 e^{-2ikh} + \alpha_2 e^{-ikh} + \alpha_3 + \alpha_4 e^{ikh} + \alpha_5 e^{2ikh}\right] = \underset{\sim}{\hat{\delta}}^{n}\left[\alpha_5 e^{-2ikh}\right.$$

$$\left. + \alpha_4 e^{-ikh} + \alpha_3 + \alpha_2 e^{ikh} + \alpha_1 e^{2ikh}\right] \qquad (6.5.5)$$

Equation (6.5.5) is thus of the form:

$$(a + ib)\hat{\delta}^{n+1} = (a - ib)\hat{\delta}^{n} \qquad (6.5.6)$$

where:

$$\left.\begin{aligned} i &= \sqrt{-1} \\ a &= \alpha(33 + \cos(2kh) + 26\cos(kh)) \\ b &= (\beta + \gamma)\sin(2kh) + (10\beta - 2\gamma)\sin(kh) \end{aligned}\right\} \qquad (6.5.7)$$

Write $\hat{\delta}^{n+1} = g\,\hat{\delta}^{n}$ where g is the amplification factor and substitute in (6.5.6) to obtain:

$$g = \frac{a - ib}{a + ib} \qquad (6.5.8)$$

Taking the modulus of this equation gives:

$$|g| = \sqrt{g\bar{g}} = 1 ,$$

therefore the linearised scheme is unconditionally stable.

6.6 The Test Problems:

The purpose of this section is to examine our algorithm using different test problems concerned with the development, migration and interaction of solitons.

Let us compute the solution of the **KdV** equation for the following problems:

(a) Consider the motion of a single soliton with initial condition given by:

$$u(x,o) = 3c\ \mathrm{sech}^2(A_1 x + D_1) \qquad (6.6.1)$$

This can be derived from the analytic solution of the KdV equation which has the form:

$$u(x,t) = 3\ c\ \mathrm{sech}^2(A_1 x - B_1 t + D_1) \qquad (6.6.2)$$

where; $A_1 = \frac{1}{2}(\varepsilon c/\mu)^{1/2}$ and $B_1 = \varepsilon c A_1$. To make a comparison with Sanz-Serna and Christie [36] we choose $\varepsilon = 1$, $\mu = 4.84\times10^{-4}$, $c = 0.3$, $D_1 = -6$, $h = 0.05$, 0.033 , 0.01 , and $\Delta t = 0.025$, 0.01 , 0.005 .

We shall impose the boundary conditions:

$$\left.\begin{aligned} u(0,t) &= u(2,t) = 0 \\ u_x(0,t) &= u_x(2,t) = 0 \end{aligned}\right\} \text{ for all time} \qquad (6.6.3)$$

Figure 6.2 shows us the behaviour of the computed solution for times from $t = 0.0$ to $t = 3.0$. These graphs have been compared with those of Greig and Morris [26] for corresponding times and if the exact solution is plotted on the same figure all curves are indistinguishable.

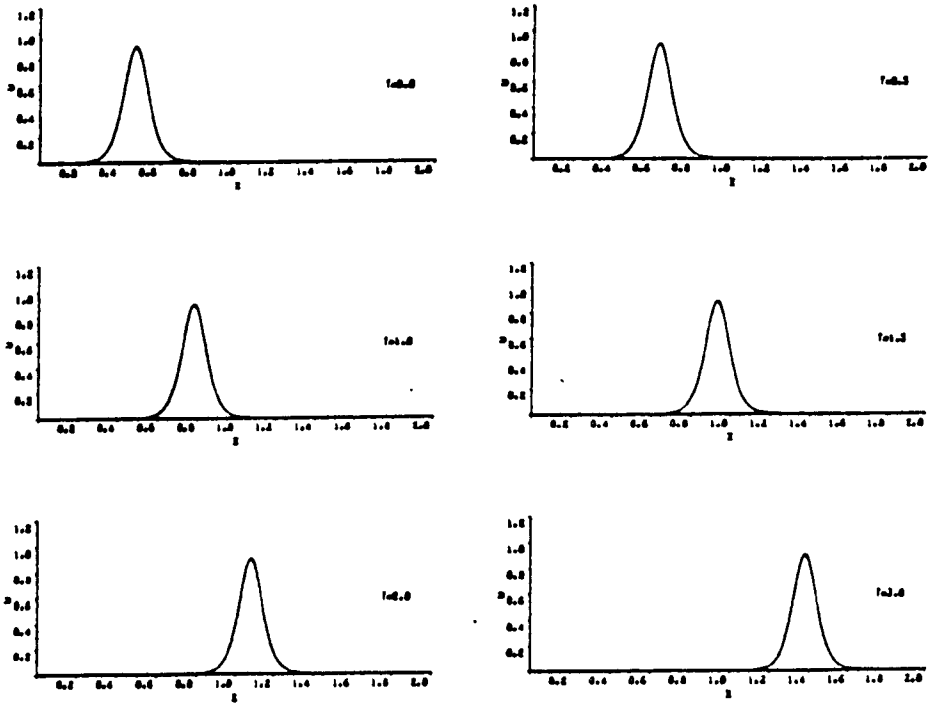

Figure 6.2 Problem (a). The motion of a single soliton with $h = 0.01$ $\Delta t = 0.005$.

(b) The interaction of two solitons with initial condition:

$$u(x,0) = 3\,c_1 \operatorname{sech}^2(A_1 x + D_1) + 3\,c_2 \operatorname{sech}^2(A_2 x + D_2) \qquad (6.6.4)$$

together with the boundary conditions which are given by:

$$\left.\begin{aligned} u(0,t) &= u(2,t) = 0 \\ u_x(0,t) &= u_x(2,t) = 0 \end{aligned}\right\} \text{ for all time} \qquad (6.6.5)$$

These conditions represent two solitons, one with amplitude $3c_1$ sited initially at $x = -D_1/A_1$ and a second with amplitude $3c_2$ placed at $x = -D_2/A_2$. It is well known that the velocity of a soliton depends directly upon its amplitude. So choosing $c_1 > c_2$ and $-D_1/A_1 < -D_2/A_2$ ensures that these solitary waves interact with increasing time. For comparison with the Greig and Morris [26] solution we have chosen $c_1 = 0.3$, $c_2 = 0.1$, $D_1 = D_2 = -6.$, $h = 0.01$, $\Delta t = 0.005$ and $A_j = \frac{1}{2}\left(\frac{\varepsilon c}{\mu}_j\right)^{1/2}$, $j = 1,2$.

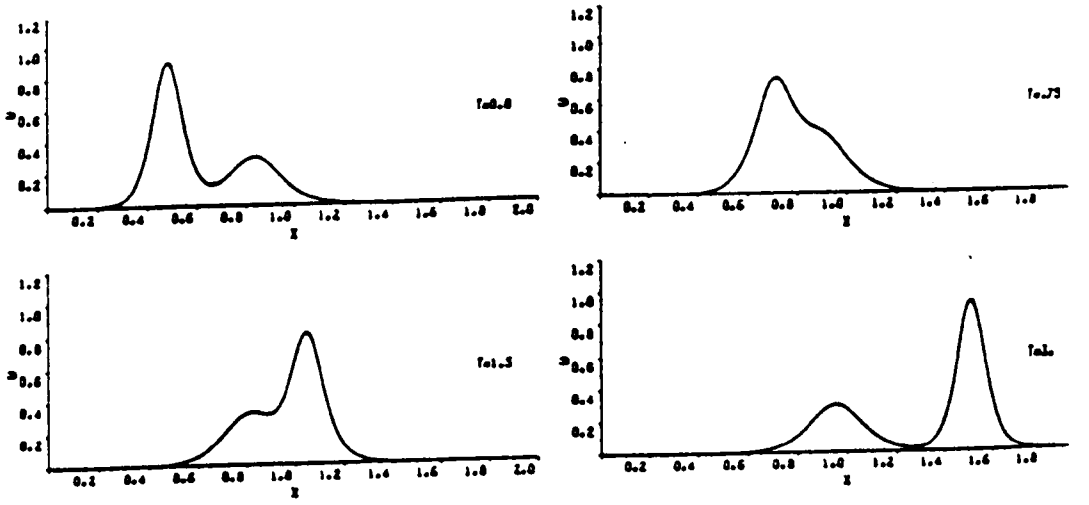

Figure 6.3 Problem (b). The interaction of the overlapping solitons with $h = 0.01$ $\Delta t = 0.005$.

The interaction of overlapping solitons produced by our algorithm is shown in Figure 6.3 and agrees well with those obtained by [26]. We find that the solitons emerge from the interaction with large and small amplitudes only slightly changed

from their original values by ≈ 1% , ≈ 0.39% respectively. The agreement with Greig and Morris [26] is very satisfactory. Figure 6.3 shows us that the two solitons placed with the larger one on the left of the smaller one. The larger soliton catches up the smaller soliton at time t = 0.75. The overlapping process continues until, at time t = 1.5 the larger soliton has overtaken the smaller one. At time t = 3, the interaction is complete and the larger soliton has separated completely from the smaller one.

(b1) Consider the motion of two well separated solitons as an initial condition:

$$u(x,0) = 3\,c_1 \operatorname{sech}^2(A_1 x + D_1) + 3\,c_2 \operatorname{sech}^2(A_2 x + D_2) \qquad (6.6.6)$$

where:

c_1 , c_2 , A_1 , A_2 , D_1 are given in problem (b)

$D_2 = -9.0$, $B_i = \varepsilon\, c_i A_i$, $i = 1,2$

the boundary conditions are chosen to be:

$$\left.\begin{aligned} u(0,t) &= u(4,t) = 0 \\ u_x(0,t) &= u_x(4,t) = 0 \end{aligned}\right\} \quad \text{for all time} \qquad (6.6.7)$$

Figure 6.4 shows that after the interaction of the two well separated solitons the large and small amplitudes change from their original values by a small amount (≈ 0.029% , ≈ 0.049% respectivly). Therefore we emphasise that after the interaction the amplitudes are virtually unchanged. From Figure 6.4 we see that the larger soliton is placed on and separated from the smaller one. As the time increases, the larger soliton catches up with the smaller when the time t = 3. The overlapping process continues and the larger soliton overtakes the smaller one at time t = 4. About time t = 6 the interaction process is complete and the larger soliton has separated completely from the smaller one:

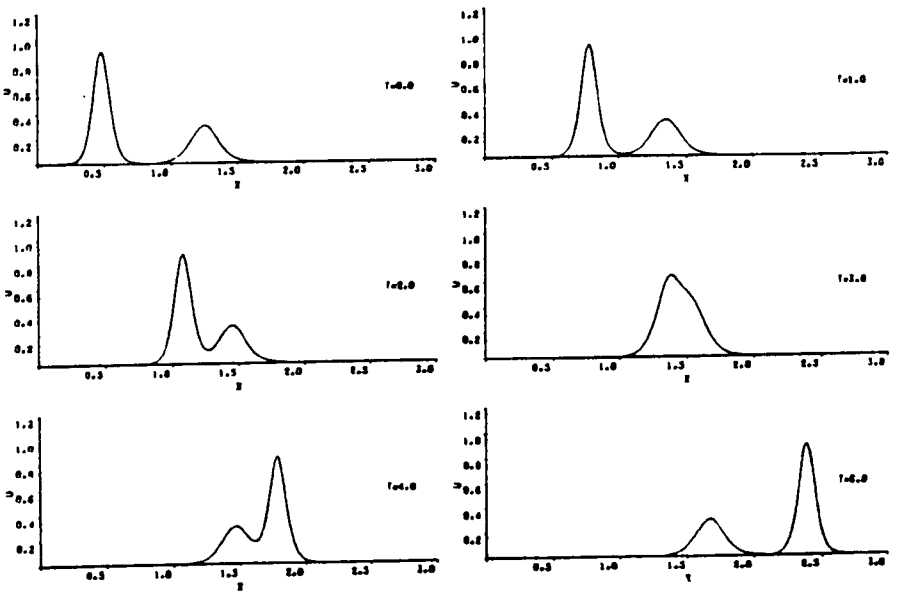

Figure 6.4 Problem (b1). The interaction of a two well separated solitons with h = 0.01 Δt = 0.005.

(c) Another interesting initial value problem for the KdV equation is given by the Gaussian distribution function:

$$u(x,0) = \exp(-x^2) \tag{6.6.8}$$

This is a typical symmetric function which tends to zero as $|x|$ tends to infinity. The finite boundary conditions imposed are:

$$\left.\begin{aligned} u(\mp 15, t) &= 0 \\ u_x(\mp 15, t) &= 0 \end{aligned}\right\} \text{ for all } t > 0 \tag{6.6.9}$$

We choose $\varepsilon = 1.0$ and we discuss each of the following cases:

(c1) $\mu = 0.04$, h = 0.1 , Δt = 0.01

(c2) $\mu = 0.01$, h = 0.1 , Δt = 0.01

(c3) $\mu = 0.001$, h = 0.025 , Δt = 0.005

(c4) $\mu = 0.0005$, h = 0.025 , Δt = 0.005

A comparison has been made with the work of Goda [59] in the first two cases. Figure 6.5 depicts the behaviour of numerical solutions of problem (c1) for times up to 12.5 . We see that the initial perturbation splits up into a soliton plus an oscillating

tail. The values of the analytic velocity $c_a \simeq 0.4$ and the numerical velocity $c_n \simeq 0.4$ agreed very well:

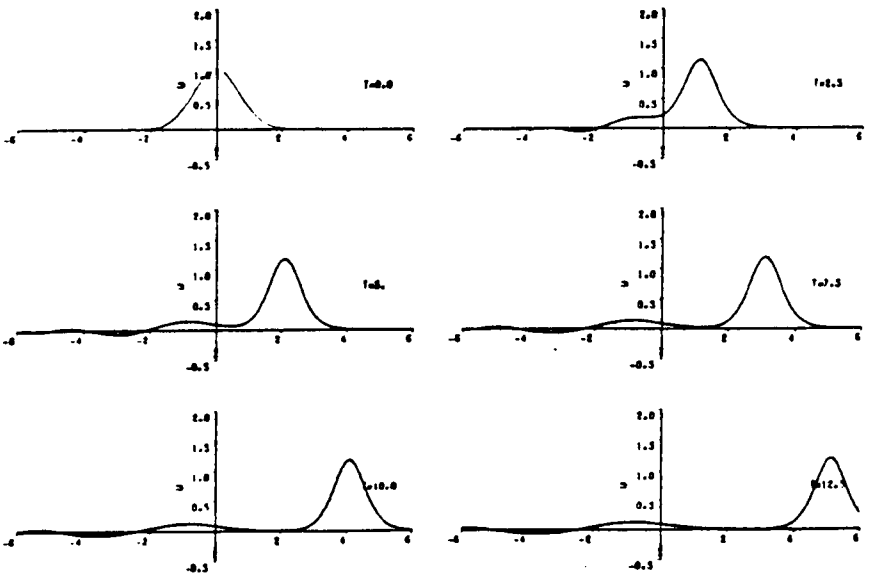

Figure 6.5 Problem (c1). A single soliton with oscillating tail for $\mu = 0.04$ $h = 0.1$ $\Delta t = 0.01$.

Figure 6.6 shows similar results for $\mu = 0.01$. We observe that the initial perturbation breaks up into three solitons. The graphs obtained by our algorithm in the cases (c1) and (c2) are identical with those given by Goda [59]. The agreement between the analytic velocity $c_a \simeq 0.5145$ and the observed velocity $c_n \simeq 0.52$ for the leading soliton was very satisfactorily:

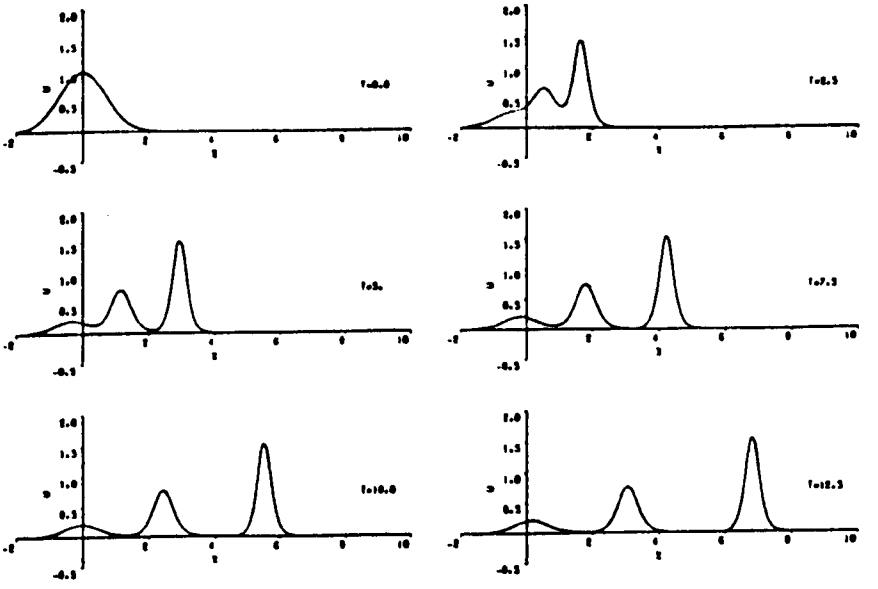

Figure 6.6 Problem (c2). The breakdown of the initial condition into 3 solitons when $\mu = 0.01$ $h = 0.1$ $\Delta t = 0.01$.

In Figure 6.7 for $\mu = 0.001$ we see that the initial perturbation breaks up into 9 solitons, whose magnitude decreases linearly to the left:

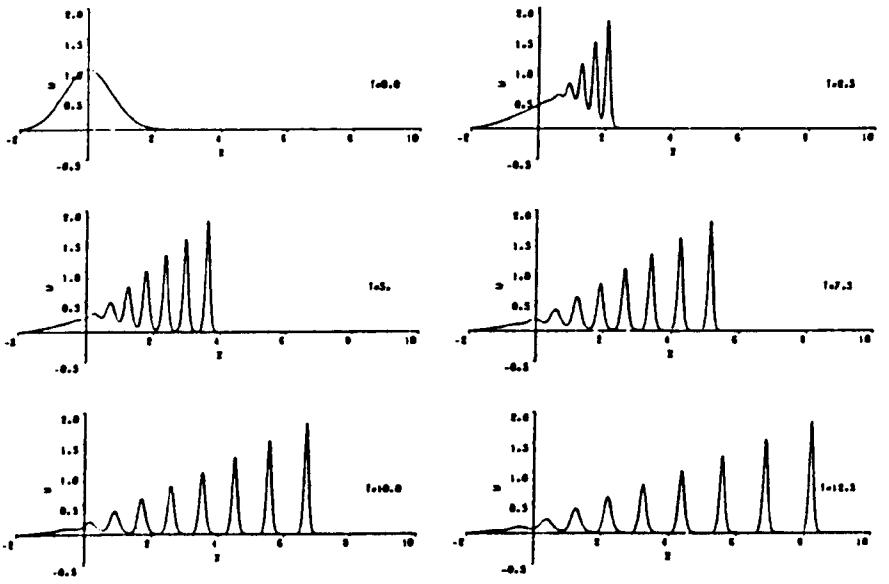

Figure 6.7 Problem (c3). The breakdown of the initial perturbation into 9 solitons with $\mu = 0.001$ $h = 0.1$ $\Delta t = 0.01$.

Figure 6.8 shows that for $\mu = 0.0005$ a train of solitons is generated when the initial perturbation splits itself into 12 solitons moving to the right:

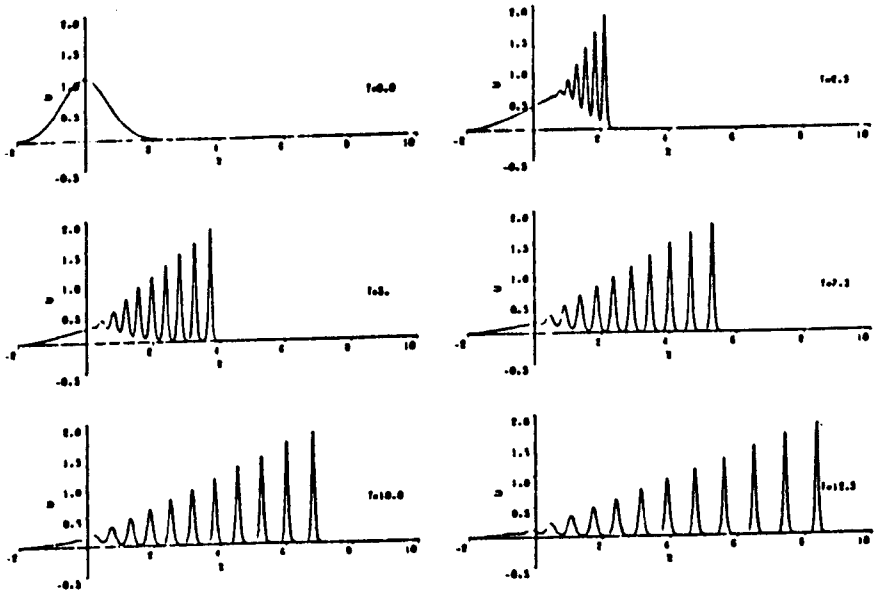

Figure 6.8 Problem (c4). The breakdown of the initial condition into 12 solitons when $\mu = 0.0005$ $h = 0.025$ $\Delta t = 0.005$.

It was found that the behaviour of the numerical solution depended upon the value of μ chosen. The initial perturbation breaks up into a train of solitons in the course of time, the actual number of solitons depending on the value of μ used. A comparison has been made of our results with those obtained by [15,39,80] and we found that there was agreement when $\mu = 0.04$, 0.01, but disagreement in other cases.

(d) As a final test example we shall consider the development of an undular bore in shallow water. This is represented by the initial condition:

$$\mathbf{u}(x,0) = \frac{1}{2}\left[1 - \tanh\left[\frac{x-25}{5}\right]\right] \qquad (6.6.10)$$

and the boundary conditions we impose are:

$$\left.\begin{aligned} \mathbf{u}(0,t) &= 1 \\ \mathbf{u}(50,t) &= 0 \\ \mathbf{u}_x(0,t) &= \mathbf{u}_x(50,t) = 0 \end{aligned}\right\} \quad \text{for all } t > 0 \qquad (6.6.11)$$

Let us consider the last example test:

$$(d1)\ \mathbf{u}(x,0) = \frac{1}{2}\left[1 - \tanh\left[\frac{|x|-25}{5}\right]\right] \qquad (6.6.12)$$

and the boundary conditions are chosen to be:

$$\left.\begin{aligned} \mathbf{u}(-50,t) &= \mathbf{u}(150,t) = 0 \\ \mathbf{u}_x(-50,t) &= \mathbf{u}_x(150,t) = 0 \end{aligned}\right\} \quad \text{for all } t > 0 \qquad (6.6.13)$$

For comparison with Vliegenthart [44] for problem (d) we have chosen $\varepsilon = 0.2$, $\mu = 0.1$ with $\Delta t = 0.05$ and $h = 0.4$. The computed solution reproduced in Figure 6.9 shows all the general features obtained by the earlier the solution [44]. However we cannot make a direct comparison with Vliegenthart [44] figures because the boundary conditions used by him are not given. We can, however, repeat his computations using his finite difference scheme and parameters together with our boundary conditions to produce comparable figures. If these are plotted also on Figure 6.9 the

graphs are indistinguishable with those obtained in the present study:

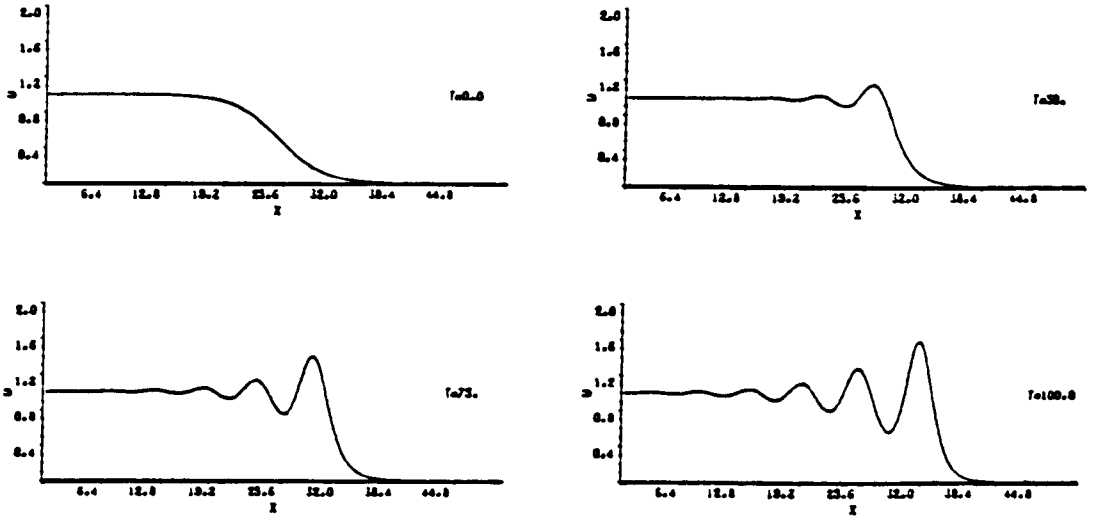

Figure 6.9 Problem (d). The solution graphs for various times with h = 0.4, Δt = 0.05, ε = 0.2, μ = 0.1.

The conservative quantities have been computed for problem (d) and it was found that these quantities varied somewhat with time. For this reason we have chosen the problem (d1). Figure 6.10 shows us that the initial perturbation problem (d1) has broken up into a regular sequence of solitons, which move steadily to the right with constant speeds whose magnitude depends upon their individual amplitude:

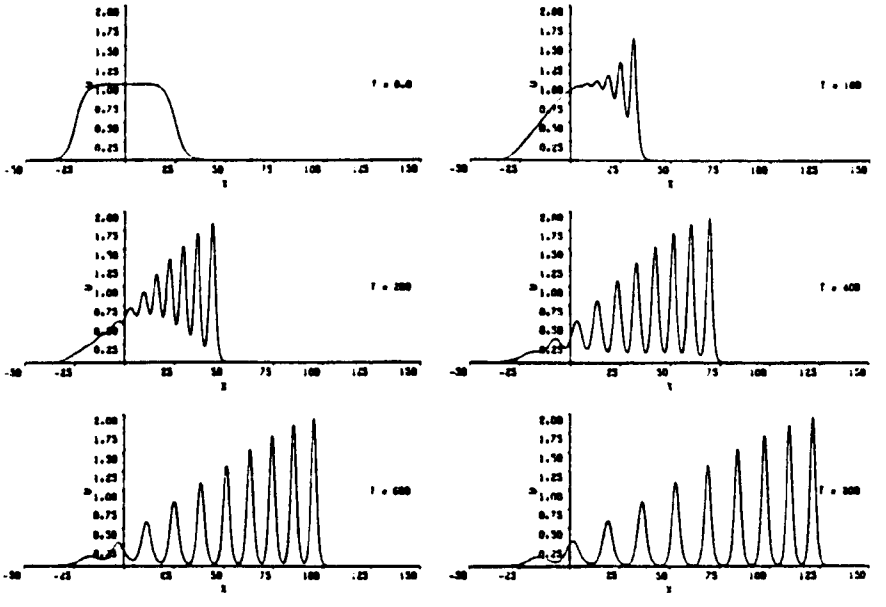

Figure 6.10 Problem (d1) . The solution graphs splits into a train of solitons with h = 0.4, Δt = 0.05, ε = 0.2, μ = 0.1.

We observe that the amplitude of the solitons vary approximately linerly. Physically these results represent among other things the development of an undular bore in shallow water and a collisionless shock in plasmas [44].

6.7 Discussion:

Any numerical scheme for computing the solution of the KdV equation must represent faithfully the amplitude and the position of the solution over many time steps with minimum errors [36], and also it should be conservative.

To examine the accuracy of our numerical scheme we have used the error in the form of the discrete L_2- and L_∞-error norms to compare the numerical and exact solutions. The L_2- and L_∞-error norms are defined by (4.5.1), (4.5.2) respectively. This error is used to compare 5 numerical methods in Tables 6.2, 6.3 for the single soliton solution [36]. We find from Table 6.2 that the Galerkin quadratic spline method compares well in accuracy with the best of the other methods. In fact, the L_2-error norm is factor 0.1 smaller than that of the modified Petrov-Galerkin method:

Table 6.2

The growth of the discrete L_2-error norm x 10^3 for single soliton

Time	Zabusky-Kruskal [27]	Hopscotch [26]	Petrov-Galerkin [36]	Modified P-G [36]	Galerkin Quadratic spline
$\Delta x = 0.05$		$\Delta t = 0.025$	$h = 0.05$	$\Delta t = 0.025$	
0.25	34.64	61.21	81.39	52.15	12.72
0.50	122.68	122.41	102.54	64.90	16.78
0.75	210.44	181.35	125.84	89.01	19.52
1.00	298.19	228.10	150.57	107.20	22.80
			$h = 0.033$	$\Delta t = 0.01$	
0.25			31.18	5.94	1.07
0.5			43.35	7.56	1.57
0.75			56.21	8.70	1.90
1.00			74.08	9.49	2.24
$\Delta x = 0.01$		$\Delta t = 0.0005$	$h = 0.01$	$\Delta t = 0.005$	
0.25	5.94	3.79	4.46	0.21	0.02
0.50	13.17	9.28	7.01	0.38	0.04
0.75	21.08	14.14	10.08	0.57	0.05
1.00	28.66	18.72	13.26	0.74	0.06

Table 6.3 shows us that the L_∞-error norm computed from our technique for a single soliton has been compared with all the method which are quoted in Table 6.3. We observe that the L_∞-error norm is greater than L_2-error norm which disagrrees with the authors [26,27,36]. Also we find that the value of L_∞-error norm using our method is smaller than even the best method (Modified Petrov-Galerkin) and in the worst case it has the same error of magnitude:

Table 6.3

The growth of the discrete L_∞-error norm x 10^3 for single soliton

Time	Zabusky-Kruskal [27]	Hopscotch [26]	Petrov-Galerkin [36]	Modified P-G [36]	Galerkin Quadratic spline
$\Delta x = 0.05$		$\Delta t = 0.025$	$h = 0.05$	$\Delta t = 0.025$	
0.25	19.4	32.7	42.18	30.22	35.79
0.50	63.5	67.4	51.85	22.85	42.68
0.75	122.4	99.3	87.60	35.86	45.92
1.00	161.4	141.6	100.41	39.39	38.94
			$h = 0.033$	$\Delta t = 0.01$	
0.25			14.27	2.80	2.96
0.5			21.65	4.53	3.24
0.75			29.78	4.85	3.79
1.00			39.37	5.85	5.85
$\Delta x = 0.01$		$\Delta t = 0.0005$	$h = 0.01$	$\Delta t = 0.005$	
0.25	2.05	1.11	1.21	0.07	0.06
0.50	4.22	2.14	2.15	0.11	0.10
0.75	6.36	3.54	3.09	0.17	0.13
1.00	8.13	4.91	3.83	0.21	0.15

Table 6.4 shows us that the error measured in terms of the discrete L_2- and L_∞-error norms is satisfactorily small for the motion of a single soliton even when the time is increased to $t = 3.0$:

Table 6.4

The growth of the error for a single soliton

Time	h = 0.05 Δt =0.025		h = 0.033 Δt =0.01		h = 0.01 Δt =0.005	
	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$
1.25	24.15	43.38	2.26	3.54	0.07	0.19
1.50	22.03	34.65	2.18	5.66	0.08	0.21
1.75	31.08	84.91	2.37	4.45	0.08	0.23
2.00	34.70	65.50	2.39	4.65	0.09	0.22
2.25	35.05	57.52	2.47	4.33	0.09	0.26
2.50	39.18	71.96	2.49	4.27	0.10	0.27
2.75	41.82	82.24	2.45	6.19	0.11	0.29
3.00	44.24	97.93	2.80	8.04	0.11	0.30

To show that our scheme is conservative we have computed the the three invariant quantities I_1 , I_2 , I_3 given by equations **(2.4.8)-(2.4.10)**, for the single soliton problem. Values are listed in Table 6.5 for the two cases h = 0.033 , Δt = 0.01 and h = 0.01 , Δt = 0.005:

Table 6.5

The computed values I_1 , I_2 , and I_3 for a single soliton

Time	I_1		I_2		I_3	
	h = .033 Δt = .01	h = .01 Δt =.005	h = .033 Δt = .01	h = .01 Δt =.005	h = .033 Δt = .01	h = .01 Δt=.005
0.0	0.144597	0.144598	0.086759	0.086759	0.046850	0.046850
0.50	0.144687	0.144598	0.086749	0.086761	0.045322	0.046735
1.00	0.144618	0.144602	0.086742	0.086763	0.045299	0.046737
1.50	0.144562	0.144604	0.086734	0.086765	0.045232	0.046739
2.00	0.144847	0.144606	0.086731	0.086767	0.045340	0.046740
2.50	0.144569	0.144607	0.086722	0.086769	0.045278	0.046742
3.00	0.144560	0.144610	0.086715	0.086771	0.045263	0.046744

From Table 6.5 we find that with our numerical scheme the three quantities I_1 , I_2 , I_3 are sensibly constant. Indeed even when fairly large time and space steps of h = 0.033 , Δt = 0.01 are used the changes are less than 0.173% , 0.051% , 3.454% respectively while for smaller values h = 0.01 , Δt = 0.005 the changes are 0.009% , 0.014% , 0.25% respectively during the computer run. A computer run on the single soliton solution with 200 nodes and 200 time steps took 13 secs of CPU time on a VAX 8650.

Further we have verified that our algorithm can adequately cope when two solitary waves coalesce for a short period and then separate with their original profiles intact, but their positions changed. By evaluating the quantities I_1 , I_2 , I_3 which are given in Table 6.6 . We find that these quantities are changed by less than 0.172% , 0.015% , 0.227% respectively during the computer run; therefore, we can consider them as relatively constant:

Table 6.6

The computed values I_1 , I_2 , I_3 for a two overlapping solitons with h = 0.01 Δt = 0.005

Time	I_1	I_2	I_3
0.0	0.228081	0.107062	0.053316
0.50	0.228124	0.107065	0.053253
1.00	0.227949	0.107068	0.053309
1.50	0.227740	0.107070	0.053262
2.00	0.227689	0.107073	0.053213
2.50	0.227732	0.107075	0.053198
3.00	0.227880	0.107078	0.053195

We have also determined the error and conserved quantities for the case when the two solitons are well separated. The three quantities I_1 , I_2 , I_3 are listed in Table 6.7:

Table 6.7

The computed values I_1 , I_2 , I_3 for a two well separated solitons with $h = 0.01$ $\Delta t = 0.005$

Time	I_1	I_2	I_3
0.0	0.228082	0.103466	0.049864
1.0	0.228088	0.103471	0.049750
2.0	0.228092	0.103476	0.049766
3.0	0.228098	0.103482	0.049864
4.0	0.228106	0.103487	0.049788
5.0	0.228112	0.103492	0.049767
6.0	0.228119	0.103496	0.049769
7.0	0.228124	0.103501	0.049773
8.0	0.228131	0.103506	0.049777

We observe that the three quantities I_1 , I_2 , I_3 have changed by less than 0.022% , 0.039% , 0.229% respectively during the computer run. So, we can consider them as constants; this is especially true for I_1 , I_2.

(b2) Let us consider the two soliton initial conditions which are determined from the analytic solution (2.3.4.1) when $t = 0.0$ in the following cases:

(i) problem (b) where $\alpha_i = \sqrt{c_i/\mu}$, $d_1 = -12$, $d_2 = -12 + \Delta$

(ii) problem **(b1)** where $\alpha_i = \sqrt{c_i/\mu}$, $d_1 = -12$, $d_2 = -18 + \Delta$

(iii) $\alpha_1 = 4.0$, $\alpha_2 = 2.0$, $d_1 = d_2 = 0.0$, $\varepsilon = 6$, $\mu = 1$. The boundary conditions are chosen as:

$$\left.\begin{aligned} u(\mp 12, t) &= 0 \\ u_x(\mp 12, t) &= 0 \end{aligned}\right\} \quad \text{for } -0.5 \le t \le 0.5 \qquad (6.7.1)$$

The first three conservative quantities I_i and the error for the problem (b2) case (i) are listed in Table 6.8:

Table 6.8

The computed vlues of the error x 10^3, I_1 , I_2 and I_3 for double soliton problem (b2) case (i) with h = 0.01, Δt = 0.005

Time	$L_2 \times 10^3$	$L_\infty \times 10^3$	I_1	I_2	I_3
0.0			0.228074	0.103456	0.049855
0.50	0.082	0.165	0.228065	0.103459	0.049798
1.00	0.085	0.148	0.228071	0.103462	0.049851
1.50	0.090	0.182	0.228081	0.103464	0.049813
2.00	0.096	0.209	0.228084	0.103467	0.049766
2.50	0.098	0.180	0.228086	0.103469	0.049751
3.00	0.093	0.199	0.228083	0.103472	0.049749

We see from this Table that the errors measured in terms of the discrete L_2- and L_∞-error norms are satisfactorily small for a two overlapping solitons even when the time is increased to t = 3 . Also the quantities I_i (i = 1,2,3) are changed by less than 0.006% , 0.016% , 0.213% respectively during the computer run. We note that the conservation of I_1 is better than that obtained in problem (b). Computationally the two overlapping solitons interact in the neighborhood of $x \approx 0.74$ at $t \approx 0.85$ which agrees with the analytic results given in equation (2.3.4.7).

After the interaction the position of the maxima of the larger and smaller solitons when t = 3 are 1.49 and 0.95 respectively which agree with the analytic values determined in equation (2.3.4.6b) . Also the larger and smaller amplitudes are changed from their original values by less than 0.792% and 0.367% respectively.

The L_2- and L_∞-error norms and the quantities I_i have also been determined for two well separated solitons problem (b2) case (ii) and are given in Table 6.9:

Table 6.9

The computed values of the error $x10^3$, I_1, I_2 and I_3 for double soliton probem cas (ii) with h = 0.01, Δt = 0.005

Time	$L_2 x10^3$	$L_\infty x10^3$	I_1	I_2	I_3
0.0			0.228082	0.103456	0.049855
1.0	0.063	0.163	0.228088	0.103461	0.049741
2.0	0.087	0.248	0.228093	0.103466	0.049757
3.0	0.068	0.206	0.228099	0.103471	0.049855
4.0	0.062	0.122	0.228107	0.103477	0.049778
5.0	0.090	0.204	0.228112	0.103481	0.049757
6.0	0.172	0.446	0.228118	0.103486	0.049760
7.0	0.293	0.781	0.228122	0.103491	0.049763
8.0	0.443	1.176	0.228129	0.103495	0.049767

Table 6.9 shows us that the L_2- and L_∞-error norms are satisfactorily small for two well separated solitons even with runs up to a time of t = 8 . The quantities I_i (i = 1,2,3) are changed by less than 0.021% , 0.038% , 0.229% respectively during the computer run. A comparison has been made between these quantities and those obtained from problem (b1), we find that they are the same.

The interaction of the two well separated solitons occurs in the neighborhood of $x \simeq 1.37$ when $t \simeq 2.95$ which agrees with the analytic results. After the interaction the two separated solitons reappeared with their original amplitudes, correct to a numerical error of less than $7.78x10^{-5}$% and 0.044% respectively.

When the two well separated solitons have large amplitudes problem (b2) case (iii), the values of the L_2- and L_∞-error norms and the first three invariant quantities are given in Table 6.10:

Table 6.10

The computed values of the error $x10^3$, I_1, I_2 and I_3 for double soliton problem (b2) case (iii) with h = 0.1, Δt = 0.0005, ε = 6, μ = 1

Time	$L_2 x10^3$	$L_\infty x10^3$	I_1	I_2	I_3
-0.5			11.99991	47.99998	211.2000
-0.4	0.908	1.103	11.99998	48.00003	209.8457
-0.3	1.255	1.512	11.99985	48.00008	209.8476
-0.2	1.637	1.948	12.00010	48.00021	209.8661
-0.1	1.918	2.121	12.00019	48.00054	210.0601
0.0	1.443	1.335	12.00034	48.00174	211.0252
0.1	1.861	1.857	12.00040	48.00139	210.4111
0.2	2.880	3.241	12.00038	48.00100	209.9110
0.3	3.482	3.890	12.00032	48.00102	209.8573
0.4	4.179	4.659	12.00041	48.00115	209.8533
0.5	4.778	5.217	12.00026	48.00126	209.8538

We observe that the method has coped well with this problem. From Table 6.10 we find that the L_2- and L_∞-error norms are small and the quantities I_i $(i = 1,2,3)$ are changed by less than 0.005% , 0.004% and 0.642% respectively during the computer run, therefore we consider the quantities I_1 and I_2 are virtually constants. After the interaction of these two well separated solitons (problem (b2) case (iii)) the larger and smaller amplitudes are changed from the original values by $\simeq$ 0.97% and $\simeq$ 0.006% respectively.

The forward and backward phase shifts defined by equation (2.3.4.8) have been calculated for problem (b), (b1), (b2(i-ii)) and we found that

$$\Delta_1 \simeq 0.11 \quad , \quad \Delta_2 \simeq -0.18$$

These agree with the values obtained analytically from equation (4.5.4) except in problem (b) where there is an error in the forward phase shift about 1%.

For problem (b2(iii))

$$\Delta_1 \simeq 0.50 \quad , \quad \Delta_2 \simeq -1.1$$

We observe that Δ_2 agrees with the value obtained analytically from equation (4.5.5) while Δ_1 does not.

The quantities I_1 , I_2 , I_3 of problems (c1) and (c2) are given in Table 6.11. We see that conservation is better in problem (c2) than in problem (c1), which may indicate that the conservation of energy could depend on the coefficient of the dispersive term (i.e. the value of μ):

Table 6.11

The computed values of I_1, I_2 , I_3 for $u(x,o) = \exp(-x^2)$

with $h = 0.1$ $\Delta t = 0.01$

	I_1		I_2		I_3	
Time	$\mu = .04$	$\mu = .01$	$\mu = .04$	$\mu = .01$	$\mu = .04$	$\mu = .01$
0.0	1.772454	1.772454	1.253314	1.253314	0.872929	0.985728
2.5	1.772532	1.772487	1.253379	1.253348	0.872261	0.982454
5.0	1.772677	1.772521	1.253445	1.253383	0.872220	0.981137
7.5	1.771568	1.772557	1.253512	1.253422	0.871163	0.981090
10.0	1.775931	1.772579	1.253580	1.253461	0.868774	0.981122
12.5	1.770864	1.772518	1.253645	1.253500	0.860989	0.981166

Table 6.11 shows that over the computer run the quantities I_1 , I_2 , I_3 are changed by less than 0.197% , 0.027% , 1.368% respectively when $\mu = 0.04$ and by 0.008% , 0.015% , 0.471% respectively when $\mu = 0.01$. So we can consider them as constants.

The total number of solitons which are generated from the Gaussian initial condition has been determined from equation (4.5.6) and we found that it agrees with those given above in Figures 6.5-6.8 .

The first three conservative quantities for problem (d1) boundary conditions(6.6.13)are listed in Table 6.12:

Table 6.12

The computed values I_1 , I_2 , I_3 for problem (dV boundary conditions(6.13)with h = 0.4, Δt = 0.05, ε = 0.2, μ = 0.1

Time	I_1	I_2	I_3
0.0	50.00010	45.00046	42.30069
100.0	50.00427	45.00789	42.25728
200.0	50.01289	45.01431	42.11004
300.0	50.01280	45.02113	42.04109
400.0	50.00404	45.02817	42.03289
500.0	49.98812	45.03519	42.03796
600.0	49.97722	45.04222	42.04924
700.0	49.98970	45.04924	42.05655
800.0	50.01744	45.05622	42.06376

We observe that the three quantities I_1 , I_2 , I_3 have changed by less than 0.046% , 0.124% , 0.634% respectively during the long time computer runs. So, we can consider them as relatively constant up to time t = 800. The values of the analytic and numerical velocities are computed to be $c_a \simeq 0.1305$, $c_n \simeq 0.128$ respectively and so gree with one another.

We conclude that Galerkin's method with quadratic spline interpolation polynomials is a useful technique for the computation of KdV solutions over long periods of time, with small space and time steps.

CHAPTER 7

COLLOCATION WITH QUINTIC SPLINES

7.1 Introduction:

So far the **KdV** equation has been solved numerically using Galerkin's method with Hermite cubic interpolation, cubic spline interpolation, and quadratic spline interpolation which have been described and studied in the previous chapters. The main disadvantage of these methods is the tedious calculations involved in the initialization and the complexity of the computations, especially that of the nonlinear term. For the above reasons we have searched for a more economic technique suitable for solving the **KdV** equation and have decided on the method of collocation using splines. The collocation approach to solving partial differential equations has two great benefits in that it does not involve integrations and it leads to banded matrices with a small band width.

At this stage we ask the question, "Is a cubic spline collocation technique suitable for solving the **KdV** equation ?" The answer is no because cubic spline interpolation has third order derivative discontinuity at the knots and so cannot represent a solution to the **KdV** equation.

We need a spline which has at least its third order derivative continuous, so we are lead to chose quintic splines which have up to the fourth order derivatives continuous.

In this chapter therefore a finite element solution of the **KdV** equation, using collocation with quintic splines as interpolation functions is set up.

7.2 The Governing Equation:

Again we are going to set up a numerical solution of the **KdV** equation:

$$u_t + \varepsilon\, u\, u_x + \mu\, u_{xxx} = 0 \qquad a \le x \le b \qquad (7.2.1)$$

where; ε and μ are positive parameters. Appropriate boundary conditions will be chosen from the following:

$$\left.\begin{aligned} u(a,t) &= \beta_1 \\ u(b,t) &= \beta_2 \\ u_x(a,t) &= u_x(b,t) = 0 \\ u_{xx}(a,t) &= u_{xx}(b,t) = 0 \end{aligned}\right\} \qquad (7.2.2)$$

and the initial conditions to be used will be prescribed later, in section 7.6.

7.3 The Collocation Solution [81,82]:

Quintic splines will be used to approximate the solution $u(x,t)$. Let π: $a = x_0 < x_1 < \ldots < x_N = b$ be a partition of $[a,b]$ by the knots x_i, and let $\phi_i(x)$ be those quintic splines with knots at the points of π. Now

$$x_N = \text{span}\ \{\, \phi_{-2}\ ,\ \phi_{-1}\ ,\ \phi_0\ ,\ \ldots\ ,\ \phi_N\ ,\ \phi_{N+1}\ ,\ \phi_{N+2}\,\}$$

form a basis for functions defined over $[a,b]$. Our task is to find out the approximate solution $u_N(x,t)$ to the solution $u(x,t)$ which is given by:

$$u_N(x,t) = \delta_{-2}(t)\,\phi_{-2}(x) + \delta_{-1}(t)\,\phi_{-1}(x) + \ldots + \delta_{N+2}(t)\,\phi_{N+2}(x)$$

$$= \sum_{i=-2}^{N+2} \delta_i(t)\,\phi_i(x)\ , \qquad (7.3.1)$$

where the δ_i are unknown time dependent parameters to be determined using the boundary conditions:

$$\left.\begin{aligned} u_N(a,t) &= \beta_1 \\ u_{Nx}(a,t) &= 0 \\ u_N(b,t) &= \beta_2 \\ u_{Nx}(b,t) &= 0 \end{aligned}\right\} \qquad (7.3.2)$$

and the collocation conditions given by:

$$u_{Nt}(x_j,t) + \varepsilon\, u_N(x_j,t)\, u_{Nx}(x_j,t) + \mu\, u_{Nxxx}(x_j,t) = 0 \quad .$$

$$j = 0,1, \ldots ,N \quad . \qquad (7.3.3)$$

The quintic spline $\phi_i(x)$ is defined by the relationships [36,73]:

$$\phi_i(x)=\frac{1}{h^5}\begin{cases} (x-x_{i-3})^5 & [x_{i-3},x_{i-2}] \\ (x-x_{i-3})^5 - 6(x-x_{i-2})^5 & [x_{i-2},x_{i-1}] \\ (x-x_{i-3})^5 - 6(x-x_{i-2})^5 + 15(x-x_{i-1})^5 & [x_{i-1},x_i] \\ (x-x_{i-3})^5 - 6(x-x_{i-2})^5 + 15(x-x_{i-1})^5 - 20(x-x_i)^5 & [x_i,x_{i+1}] \\ (x-x_{i-3})^5 - 6(x-x_{i-2})^5 + 15(x-x_{i-1})^5 - 20(x-x_i)^5 + 15(x-x_{i+1})^5 & [x_{i+1},x_{i+2}] \\ (x-x_{i-3})^5 - 6(x-x_{i-2})^5 + 15(x-x_{i-1})^5 - 20(x-x_i)^5 + 15(x-x_{i+1})^5 - 6(x-x_{i+2})^5 & [x_{i+2},x_{i+3}] \\ 0 & \text{otherwise} \end{cases} , \qquad (7.3.4)$$

where; $h = (x_i - x_{i-1})$ for all i, implying that all intervals $[x_{i-1},x_i]$ are of equal size.

The quintic spline $\phi_i(x)$ and its three principle derivatives vanish outside the interval $[x_{i-3},x_{i+3}]$. In Table 7.1 the values of $\phi_i(x)$ and its principle derivatives at the relevant knots are listed for convenience:

Table 7.1

x	x_{i-3}	x_{i-2}	x_{i-1}	x_i	x_{i+1}	x_{i+2}	x_{i+3}
$\phi_i(x)$	0	1	26	66	26	1	0
$\phi_i'(x)$	0	$\frac{5}{h}$	$\frac{50}{h}$	0	$-\frac{50}{h}$	$-\frac{5}{h}$	0
$\phi_i''(x)$	0	$\frac{20}{h^2}$	$\frac{40}{h^2}$	$-\frac{120}{h^2}$	$\frac{40}{h^2}$	$\frac{20}{h^2}$	0
$\phi_i'''(x)$	0	$\frac{60}{h^3}$	$-\frac{120}{h^3}$	0	$\frac{120}{h^3}$	$-\frac{60}{h^3}$	0

Substituting (7.3.1) into (7.3.3) leads to the equation:

$$\sum_{i=-2}^{N+2} \phi_i(x_j)\,\dot{\delta}_i + \varepsilon \sum_{i=-2}^{N+2} \phi_i'(x_j)\,\delta_i \sum_{k=-2}^{N+2} \phi_k(x_j)\,\delta_k$$

$$+ \mu \sum_{i=-2}^{N+2} \phi_i'''(x_j)\,\delta_i = 0 \qquad j = 0, \ldots, N \quad (7.3.5)$$

Suppose that δ_i is linearly interpolated between two time levels n and $n+1$ by:

$$\delta_i = (1-\theta)\delta_i^n + \theta\,\delta_i^{n+1} \quad (7.3.6)$$

where $0 \le \theta \le 1$ and δ_i^n are the parameters at the time $n\Delta t$. The time derivative is discretised using the standard finite difference formula:

$$\frac{d\delta}{dt}_i = \frac{1}{\Delta t}\,(\delta_i^{n+1} - \delta_i^n) \quad (7.3.7)$$

Hence equation (7.3.5) can be written as:

$$\sum_{i=-2}^{N+2} \left[\phi_i + \theta\,\Delta t(\varepsilon\,\phi_i' \sum_{k=-2}^{N+2} \phi_k\,\delta_k^n + \mu\,\phi_i''') \right] \delta_i^{n+1}$$

$$= \sum_{i=-2}^{N+2} \left[\phi_i - (1-\theta)\Delta t(\varepsilon\,\phi_i' \sum_{k=-2}^{N+2} \phi_k\,\delta_k^n + \mu\,\phi_i''') \right] \delta_i^n \quad (7.3.8)$$

where the basis functions and their derivatives are evaluated at the $N+1$ knots x_j , $j = 0,1, \ldots ,N$.

Giving the parameter θ the values 0 , $\frac{1}{2}$, 1 produces explicit, Crank-Nicolson and backward difference schemes respectively.

Now assume $\theta = \frac{1}{2}$ then the equation (7.3.8) takes the form:

$$\sum_{i=-2}^{N+2} \left[\phi_i + \frac{\Delta t}{2} \left(\varepsilon \phi_i' \sum_{k=-2}^{N+2} \phi_k \delta_k^n + \mu \phi_i''' \right) \right] \delta_i^{n+1}$$

$$= \sum_{i=-2}^{N+2} \left[\phi_i - \frac{\Delta t}{2} \left(\varepsilon \phi_i' \sum_{k=-2}^{N+2} \phi_k \delta_k^n + \mu \phi_i''' \right) \right] \delta_i^n \qquad (7.3.9)$$

Using the values given in Table 7.1 equation (7.3.9) can be calculated at the knots x_j , $j = 0,1, \ldots ,N$, so that at $x = x_0$ equation (7.3.9) gives:

$$\alpha_{01} \delta_{-2}^{n+1} + \alpha_{02} \delta_{-1}^{n+1} + \alpha_{03} \delta_0^{n+1} + \alpha_{04} \delta_1^{n+1} + \alpha_{05} \delta_2^{n+1} =$$

$$\alpha_{05} \delta_{-2}^n + \alpha_{04} \delta_{-1}^n + \alpha_{03} \delta_0^n + \alpha_{02} \delta_1^n + \alpha_{01} \delta_2^n \qquad (7.3.10)$$

where:

$$\alpha_{01} = 1 - R_1 Z_{-2} - R_2 \ , \quad \alpha_{02} = 26 - 10R_1 Z_{-2} + 2R_2 \ , \quad \alpha_{03} = 66 \ ,$$

$$\alpha_{04} = 26 + 10R_1 Z_{-2} - 2R_2 \ , \quad \alpha_{05} = 1 + R_1 Z_{-2} + R_2$$

$$Z_{-2} = \delta_{-2} + 26\delta_{-1} + 66\delta_0 + 26\delta_1 + \delta_2 \ , \quad R_1 = \frac{5}{2h} \varepsilon \Delta t \ , \quad R_2 = \frac{30}{h^3} \mu \Delta t \qquad (7.3.11)$$

at $x = x_1$ equation (7.3.9) becomes:

$$\alpha_{11} \delta_{-1}^{n+1} + \alpha_{12} \delta_0^{n+1} + \alpha_{13} \delta_1^{n+1} + \alpha_{14} \delta_2^{n+1} + \alpha_{15} \delta_3^{n+1} =$$

$$\alpha_{15} \delta_{-1}^n + \alpha_{14} \delta_0^n + \alpha_{13} \delta_1^n + \alpha_{12} \delta_2^n + \alpha_{11} \delta_3^n \qquad (7.3.12)$$

where:

$$\alpha_{11} = 1 - R_1 Z_{-1} - R_2 \quad , \quad \alpha_{12} = 26 - 10R_1 Z_{-1} + 2R_2 \quad , \quad \alpha_{13} = 66,$$

$$\alpha_{14} = 26 + 10R_1 Z_{-1} - 2R_2 \quad , \quad \alpha_{15} = 1 + R_1 Z_{-1} + R_2,$$

$$Z_{-1} = \delta^n_{-1} + 26\delta^n_0 + 66\delta^n_1 + 26\delta^n_2 + \delta^n_3 \qquad (7.3.13)$$

.

.

.

at $x = x_N$ equation (7.3.9) becomes:

$$\alpha_{N1}\,\delta^{n+1}_{N-2} + \alpha_{N2}\,\delta^{n+1}_{N-1} + \alpha_{N3}\,\delta^{n+1}_{N} + \alpha_{N4}\,\delta^{n+1}_{N+1} + \alpha_{N5}\,\delta^{n+1}_{N+2} =$$

$$\alpha_{N5}\,\delta^{n}_{N-2} + \alpha_{N4}\,\delta^{n}_{N-1} + \alpha_{N3}\,\delta^{n}_{N} + \alpha_{N2}\,\delta^{n}_{N+1} + \alpha_{N1}\,\delta^{n}_{N+2} \qquad (7.3.14)$$

where:

$$\alpha_{N1} = 1 - R_1 Z_{N-2} - R_2 \quad , \quad \alpha_{N2} = 26 - 10R_1 Z_{N-2} + 2R_2 \quad , \quad \alpha_{N3} = 66,$$

$$\alpha_{N4} = 26 + 10R_1 Z_{N-2} - 2R_2 \quad , \quad \alpha_{N5} = 1 + R_1 Z_{N-2} + R_2 \quad ,$$

$$Z_{N-2} = \delta^n_{N-2} + 26\delta^n_{N-1} + 66\delta^n_N + 26\delta^n_{N+1} + \delta^n_{N+2} \qquad (7.3.15)$$

Generally, these equations can be written as a recurrence relationship:

$$\alpha_{i1}\,\delta^{n+1}_{i-2} + \alpha_{i2}\,\delta^{n+1}_{i-1} + \alpha_{i3}\,\delta^{n+1}_{i} + \alpha_{i4}\,\delta^{n+1}_{i+1} + \alpha_{i5}\,\delta^{n+1}_{i+2} =$$

$$\alpha_{i5}\,\delta^{n}_{i-2} + \alpha_{i4}\,\delta^{n}_{i-1} + \alpha_{i3}\,\delta^{n}_{i} + \alpha_{i2}\,\delta^{n}_{i+1} + \alpha_{i1}\,\delta^{n}_{i+2} \qquad (7.3.16)$$

where:

$i = 0 , 1 , \ldots , N$

$$\alpha_{i1} = 1 - R_1 Z_{i-2} - R_2 \quad , \quad \alpha_{i2} = 26 - 10R_1 Z_{i-2} + 2R_2 \quad , \quad \alpha_{i3} = 66,$$

$$\alpha_{i4} = 26 + 10R_1 Z_{i-2} - 2R_2 \quad , \quad \alpha_{i5} = 1 + R_1 Z_{i-2} + R_2,$$

$$Z_{i-2} = \delta^n_{i-2} + 26\delta^n_{i-1} + 66\delta^n_i + 26\delta^n_{i+1} + \delta^n_{i+2},$$

$$R_1 = \frac{5}{2h}\varepsilon\Delta t \quad , \quad R_2 = \frac{30}{h^3}\mu\Delta t \tag{7.3.17}$$

The system (7.3.16) consists of N+1 linear equations in N+5 unknowns $(\delta_{-2}, \delta_{-1}, \delta_0, \ldots, \delta_{N+1}, \delta_{N+2})^T$. To obtain a unique solution to this system we need 4 additional constraints. These are obtained from the boundary conditions (7.3.2) which require that:

$$\left.\begin{aligned} \delta_{-2} + 26\delta_{-1} + 66\delta_0 + 26\delta_1 + \delta_2 &= \beta_1 \\ -5\delta_{-2} - 50\delta_{-1} + 50\delta_1 + 5\delta_2 &= 0 \\ \delta_{N-2} + 26\delta_{N-1} + 66\delta_N + 26\delta_{N+1} + \delta_{N+2} &= \beta_2 \\ -5\delta_{N-2} - 50\delta_{N-1} + 50\delta_{N+1} + 5\delta_{N+2} &= 0 \end{aligned}\right\} \tag{7.3.18}$$

By solving the first two equations of (7.3.18) simultaneously in δ_{-2} and δ_{-1}, we obtain:

$$\left.\begin{aligned} \delta_{-2} &= -\frac{5}{8}\beta_1 + \frac{165}{4}\delta_0 + \frac{65}{2}\delta_1 + \frac{9}{4}\delta_2 \\ \delta_{-1} &= \frac{1}{16}\beta_1 - \frac{33}{8}\delta_0 - \frac{9}{4}\delta_1 - \frac{1}{8}\delta_2 \end{aligned}\right\} \tag{7.3.19}$$

Similarly, solve the last two equations of (7.3.18) simultaneously for δ_{N+1}, δ_{N+2}, to get:

$$\left.\begin{aligned} \delta_{N+2} &= -\frac{5}{8}\beta_2 + \frac{165}{4}\delta_N + \frac{65}{2}\delta_{N-1} + \frac{9}{4}\delta_{N-2} \\ \delta_{N+1} &= \frac{1}{16}\beta_2 - \frac{33}{8}\delta_N - \frac{9}{4}\delta_{N-1} - \frac{1}{8}\delta_{N-2} \end{aligned}\right\} \tag{7.3.20}$$

Eliminating δ_{-2} and δ_{-1} from the first two equations of the system (7.3.16) using equations (7.3.19) to obtain:

$$-49.5\delta_0^{n+1} - 39\delta_1^{n+1} - 1.5\delta_2^{n+1} = 49.5\delta_0^n + 39\delta_1^n + 1.5\delta_2^n - 1.5\beta_1 \qquad (7.3.21)$$

$$s_1\,\delta_0^{n+1} + s_2\,\delta_1^{n+1} + s_3\,\delta_2^{n+1} + s_4\,\delta_3^{n+1} = s_5\,\delta_0^n + s_6\,\delta_1^n + s_7\,\delta_2^n + s_8\,\delta_3^n + \beta_1^*$$

where:

$$s_1 = \frac{1}{8}(175 - 47R_1Z_{-1} + 49R_2)\ ,\quad s_2 = \frac{1}{4}(255 + 9(R_1Z_{-1} + R_2))$$

$$s_3 = \frac{1}{8}(207 + 81R_1Z_{-1} - 15R_2)\ ,\quad s_4 = 1 + R_1Z_{-1} + R_2$$

$$s_5 = \frac{1}{8}(175 + 47R_1Z_{-1} + -49R_2)\ ,\quad s_6 = \frac{1}{4}(255 - 9R_1Z_{-1} - 9R_2)$$

$$s_7 = \frac{1}{8}(207 - 81R_1Z_{-1} + 15R_2)\ ,\quad s_8 = 1 - R_1Z_{-1} - R_2$$

$$\beta_1^* = \frac{\beta}{8}1(R_1Z_{-1} + R_2) \qquad (7.3.22)$$

Similarly, eliminating δ_{N+1} and δ_{N+2} from the last two equations of (7.3.16) and using equations (7.3.20) to obtain:

$$y_1\,\delta_{N-3}^{n+1} + y_2\,\delta_{N-2}^{n+1} + y_3\,\delta_{N-1}^{n+1} + y_4\,\delta_N^{n+1} = y_5\,\delta_{N-3}^n + y_6\,\delta_{N-2}^n + y_7\,\delta_{N-1}^n + y_8\,\delta_N^n - \beta_2^*$$

where:

$$y_8 = \frac{1}{8}(175 - 47R_1Z_{N-3} + 49R_2)\ ,\quad y_7 = \frac{1}{4}(255 + 9(R_1Z_{N-3} + R_2))$$

$$y_6 = \frac{1}{8}(207 + 81R_1Z_{N-3} - 15R_2)\ ,\quad y_5 = 1 + R_1Z_{N-3} + R_2$$

$$y_4 = \frac{1}{8}(175 + 47R_1Z_{N-3} + -49R_2)\ ,\quad y_3 = \frac{1}{4}(255 - 9R_1Z_{N-3} - 9R_2)$$

$$y_2 = \frac{1}{8}(207 - 81R_1Z_{N-3} + 15R_2)\ ,\quad y_1 = 1 - R_1Z_{N-3} - R_2$$

$$\beta_2^* = \frac{\beta}{8}2(R_1Z_{N-3} + R_2) \qquad (7.3.23)$$

$$1.5\delta_{N-2}^{n+1} + 39\delta_{N-1}^{n+1} + 49.5\delta_N^{n+1} = -1.5\delta_{N-2}^n - 39\delta_{N-1}^n - 49.5\delta_N^n + 1.5\beta_2 \qquad (7.3.24)$$

The equation (7.3.21)-(7.3.24) together with the third through (N-1)th equations of (7.3.16) give N+1 equations in N+1 unknowns $(\delta_0\ ,\ \delta_1\ ,\ \delta_3\ ,\ \ldots\ ,\ \delta_N)^T$ which can be written in a matrix form as:

$$A(\delta^n)\,\delta^{n+1} = B(\delta^n)\delta^n + r \qquad (7.3.25)$$

where; $A(\delta^n)$, and $B(\delta^n)$ are penta-diagonal $(N+1)x(N+1)$ matrices and r is $N+1$ vector:

$$r = \left[-1.5\beta_1, \beta_1^*, 0, 0, \ldots, -\beta_2^*, 1.5\beta_2\right] \qquad (7.3.26)$$

Since the matrices $A(\delta^n)$, and $B(\delta^n)$ depend on δ^n, the matrix equation (7.3.25) is nonlinear. We handle the problem by solving not equation (7.3.25) directly but by setting up and solving an equivalent system [50,68]. Such a system is:

$$A(\delta^n)\,\hat{\delta}^{n+1} = B(\delta^n)\delta^n + r \qquad (7.3.27a)$$

$$A\left(\frac{\hat{\delta}^{n+1} + \delta^n}{2}\right)\delta^{n+1} = B\left(\frac{\hat{\delta}^{n+1} + \delta^n}{2}\right)\delta^n + r \qquad (7.3.27b)$$

where; equation (7.3.27a) predicts the first approximation $\hat{\delta}^{n+1}$ then equation (7.3.27b) corrects iteratively the improved approximation.

Our approach to the solution of the nonlinear system (7.3.27) is to store the pentadiagonal matrices $A(\delta^n)$ and $B(\delta^n)$ in rectangular form $(N+1)x5$ and then use the penta-diagonal algorithm (see Appendix A2) to solve the system (7.3.27). The boundary parameters δ_{-2} , δ_{-1} , δ_{N+1} , and δ_{N+2} can be computed at each time step from equations (7.3.19)-(7.3.20).

To start the iterative procedure (7.3.27) an initial vector δ^0 must be determined from the initial condition on $u(x,t)$. Once the parameters δ have been determined at a specified time then we can compute the solution at the required knots from the formula

$$u_N(x_i, n\Delta t) = \delta_{i-2}^n + 26\delta_{i-1}^n + 66\delta_i^n + 26\delta_{i+1}^n + \delta_{i+2}^n \qquad (7.3.28)$$

$$i = 0, 1, \ldots, N$$

7.4. The Initial State:

From the initial condition $\mathbf{u}(x,0)$ on the function $\mathbf{u}(x,t)$ we must determine the initial vector $\underset{\sim}{\delta}^0$ in order that the determination of the time evolution of $\underset{\sim}{\delta}$ and hence $\mathbf{u}$ can be started.

We firstly rewrite equation (7.3.1) for the initial condition:

$$u_N(x,0) = \sum_{i=-2}^{N+2} \delta_i^0 \phi_i(x_i) \quad , \tag{7.4.1}$$

where; δ_i^0 are unknown parameters to be determined. To do this we require $u_N(x,0)$ to satisfy the following constraints:

(a) It must agree with the initial condition $\mathbf{u}(x,0)$ at the knots; leading to N+1 conditions, and

(b) The first and second derivatives of the approximate initial condition shall agree with those of the exact initial condition at both ends of the range: 4 further conditions.

These two conditions (a) and (b) can be expressed as:

$$\left.\begin{aligned} u_N'(x_0,0) &= 0 \\ u_N''(x_0,0) &= 0 \\ u_N(x_i,0) &= u(x_i,0) \qquad i = 0,1,\ldots,N \\ u_N''(x_N,0) &= 0 \\ u_N'(x_N,0) &= 0 \end{aligned}\right\} \tag{7.4.2}$$

From Table 7.1 the system (7.4.2) can be reduced to:

$$\left.\begin{aligned}
&-5\delta_{-2} - 50\delta_{-1} + 50\delta_{1} + 5\delta_{2} = 0\\
&20\delta_{-2} + 40\delta_{-1} - 120\delta_{0} + 40\delta_{1} + 20\delta_{2} = 0\\
&\delta_{i-2} + 26\delta_{i-1} + 66\delta_{i} + 26\delta_{i+1} + \delta_{i+2} = u(x_i,0)\\
&20\delta_{N-2} + 40\delta_{N-1} - 120\delta_{N} + 40\delta_{N+1} + 20\delta_{N+2} = 0\\
&-5\delta_{N-2} - 50\delta_{N-1} + 50\delta_{N+1} + 5\delta_{N+2} = 0
\end{aligned}\right\} \qquad (7.4.3)$$

The equations (7.4.3) can be written as a matrix equation of the form:

$$A\ \underset{\sim}{\delta}^{0} = \underset{\sim}{b} \qquad (7.4.4)$$

where:

$$A = \begin{bmatrix}
-5 & -50 & 0 & 50 & 5 & & & & & \\
20 & 40 & -120 & 40 & 20 & & & & & \\
1 & 26 & 66 & 26 & 1 & & & & & \\
 & 1 & 26 & 66 & 26 & 1 & & & & \\
 & & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\
 & & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\
 & & & & & & & & & \\
 & & & & & & & & & \\
 & & & & & & 1 & 26 & 66 & 26 & 1 \\
 & & & & & & 20 & 40 & -120 & 40 & 20 \\
 & & & & & & -5 & -50 & 0 & 50 & 5
\end{bmatrix}$$

and $\underset{\sim}{\delta}^{0} = \left[\, \delta^{0}_{-2}\ ,\ \delta^{0}_{-1}\ ,\ \ldots\ ,\ \delta^{0}_{N+1}\ ,\ \delta^{0}_{N+2} \,\right]^{T}$. While the vector $\underset{\sim}{b}$ has the form

$$\underset{\sim}{b} = \left[0\ ,\ 0\ ,\ u(x_0,0)\ ,\ u(x_1,0)\ ,\ \ldots\ ,\ u(x_N,0)\ ,\ 0\ ,\ 0\right]^{T} \qquad (7.4.5)$$

To solve the system (7.4.4), reduce the matrix A to penta-diagonal form by the following steps:

(1) Solve the first two equations of the system (7.4.4) simultaneously in δ_{-2} and δ_{-1} to obtain:

$$\left.\begin{aligned} \delta_{-2} &= 7.5\delta_0 - 5\delta_1 - 1.5\delta_2 \\ \delta_{-1} &= -0.75\delta_0 + 1.5\delta_1 + 0.25\delta_2 \end{aligned}\right\} \qquad (7.4.6)$$

Eliminating δ_{-2} and δ_{-1} from the third and fourth equations of (7.4.4) gives:

$$\left.\begin{aligned} 54\delta_0 + 60\delta_1 + 6\delta_2 &= u(x_0,0) \\ 25.25\delta_0 + 67.5\delta_1 + 26.25\delta_2 + \delta_3 &= u(x_1,0) \end{aligned}\right\} \qquad (7.4.7)$$

(2) Similarly, by solving the last two equations of the system (7.4.4) simultaneously we get:

$$\left.\begin{aligned} \delta_{N+2} &= 7.5\delta_N - 5\delta_{N-1} - 1.5\delta_{N-2} \\ \delta_{N+1} &= -0.75\delta_N + 1.5\delta_{N-1} + 0.25\delta_{N-2} \end{aligned}\right\} \qquad (7.4.8)$$

Eliminating δ_{N+1} and δ_{N+2} from the (N+1)th and N-th equations of the system (7.4.4) gives:

$$\left.\begin{aligned} \delta_{N-3} + 26.25\delta_{N-2} + 67.5\delta_{N-1} + 25.25\delta_N &= u(x_{N-1},0) \\ 6\delta_{N-2} + 60\delta_{N-1} + 54\delta_N &= u(x_N,0) \end{aligned}\right\} \qquad (7.4.9)$$

Hence the system (7.4.4) is reduced to penta-diagonal (N+1)x(N+1) form. To solve that system apply the penta-diagonal algorithm (see Appendix A2) to obtain the computed solution $(\delta_0, \delta_1, \ldots, \delta_N)^T$ and hence compute δ_{-2}, δ_{-1}, δ_{N+1}, and δ_{N+2} from equations (7.4.6) and (7.4.8). So the initial vector $\underset{\sim}{\delta}^0$ is determined.

7.5 The Stability Analysis:

The investigation of the stability of the **KdV** equation will be based on the von Neuman theory in which the growth factor of a

typical Fourier mode defined as:

$$\delta_j^n = \hat{\delta}^n \, e^{ijkh} \tag{7.5.1}$$

where; k is the mode number and h is the element size, is determined from the numerical scheme (7.3.16)-(7.3.17).

The nonlinear term $\mathbf{u}\,\mathbf{u}_x$ of the **KdV** equation cannot be handled by the Fourier mode method. Therefore we tackle this problem by linearising this term [26,32,35,53]. We assume that the quantity $\mathbf{u}$ in the nonlinear term $\mathbf{u}\,\mathbf{u}_x$ is locally constant which is equivalent to supposing that in equation (7.3.17) all the δ_j^n are equal to a local constant d, so that the equation (7.3.16) can now be written as:

$$\alpha_1 \, \delta_{j-2}^{n+1} + \alpha_2 \, \delta_{j-1}^{n+1} + \alpha_3 \, \delta_j^{n+1} + \alpha_4 \, \delta_{j+1}^{n+1} + \alpha_5 \, \delta_{j+2}^{n+1} =$$

$$\alpha_5 \, \delta_{j-2}^{n} + \alpha_4 \, \delta_{j-1}^{n} + \alpha_3 \, \delta_j^{n} + \alpha_2 \, \delta_{j+1}^{n} + \alpha_1 \, \delta_{j+2}^{n} \tag{7.5.2}$$

where:

$j = 0, 1, \ldots, N$

$$\alpha_1 = 1 - R_1^* - R_2, \quad \alpha_2 = 26 - 10R_1^* + 2R_2, \quad \alpha_3 = 66,$$

$$\alpha_4 = 26 + 10R_1^* - 2R_2, \quad \alpha_5 = 1 + R_1^* + R_2,$$

$$R_1^* = \frac{5}{2h} \varepsilon \Delta t (120d), \quad R_2 = \frac{30}{h^3} \mu \Delta t \tag{7.5.3}$$

If we insert the Fourier mode (7.5.1) in equation (7.5.2) we obtain:

$$\hat{\delta}^{n+1}\left[\alpha_1 e^{-2ikh} + \alpha_2 e^{-ikh} + \alpha_3 + \alpha_4 e^{ikh} + \alpha_5 e^{2ikh}\right] = \hat{\delta}^n\left[\alpha_5 e^{-2ikh} + \right.$$

$$\left. \alpha_4 e^{-ikh} + \alpha_3 + \alpha_2 e^{ikh} + \alpha_1 e^{2ikh} \right] \tag{7.5.4}$$

Rewrite this equation in simple form:

$$(a + ib)\hat{\delta}^{n+1} = (a - ib)\hat{\delta}^{n} \qquad (7.5.5)$$

where:

$$\left.\begin{aligned} i &= \sqrt{-1} \\ a &= 33 + \cos(2kh) + 26\cos(kh) \\ b &= (R_1^* + R_2)\sin(2kh) + (10R_1^* - 2R_2)\sin(kh) \end{aligned}\right\} \qquad (7.5.6)$$

Let $\hat{\delta}^{n+1} = g\,\hat{\delta}^{n}$ where g is the amplification factor for the mode and substitute in (7.5.5) to get:

$$g = \frac{a - ib}{a + ib} \qquad (7.5.7)$$

Taking the modulus of this equation gives:

$$|g| = \sqrt{g\bar{g}} = 1 ,$$

Therefore the linearised numerical scheme is unconditionally stable.

7.6 The Test Problems:

The principal purpose of the work reported in this section is the thorough testing of the collocation quintic spline algorithm based on the method which has been described in this chapter.

For the tests we shall compute the numerical solution of the KdV equation with different initial and boundary conditions which are chosen as follows.

(a) The initial condition which represents the motion of a single soliton given by:

$$u(x, 0) = 3\,c\,\mathrm{sech}^2(A_1 x + D_1) \qquad (7.6.1)$$

where; A_1 , D_1 and c are given constants, together with the boundary conditions:

$$\left.\begin{aligned} u(0,t) &= u(2,t) = 0 \\ u_x(0,t) &= u_x(2,t) = 0 \end{aligned}\right\} \text{ for all time} \qquad (7.6.2)$$

Since the KdV equation has an analytic solution of the form [26]:

$$u(x,t) = 3\,c\,\mathrm{sech}^2(A_1 x - B_1 t + D_1) \qquad (7.6.3)$$

provided:

$$A_1 = \tfrac{1}{2}(\varepsilon\, c/\mu)^{1/2} \quad \text{and} \quad B_1 = \varepsilon\, c\, A_1 \qquad (7.6.4)$$

equation (7.6.1) is a possible initial condition if $A_1 = \frac{1}{2}(\varepsilon\, c/\mu)^{1/2}$ and in fact represents a single soliton moving to the right.

To make comparison with the work of Sanz-Serna and Christie [36] we choose $\varepsilon = 1$, $\mu = 4.84\times10^{-4}$, $c = 0.3$, $D_1 = -6$, $h = 0.05,\ 0.033,\ 0.01$, and $\Delta t = 0.025,\ 0.01,\ 0.005$. We observe in Figure 7.1, the computed solution. The soliton moves to the right at constant speed with unchanged amplitude for times from $t = 0.0$ to $t = 3.0$. When the exact solution (7.6.3) is plotted on the same figure, we find that the curves are indistinguishable. These graphs have been compared exactly with those of Greig and Morris [26] for corresponding times and the agreement is also excellent:

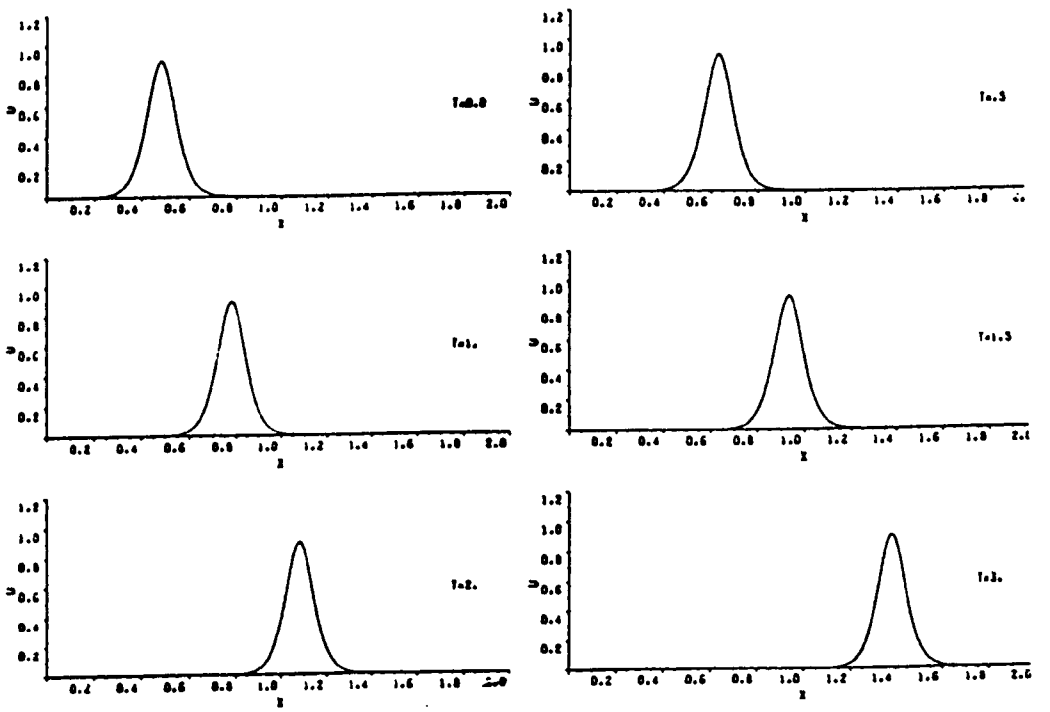

Figure 7.1 Problem (a). The motion of single soliton with $h = 0.01$ $\Delta t = 0.005$.

(b) The interaction of two overlapping solitons with initial condition given by:

$$u(x,o) = 3\,c_1 \operatorname{sech}^2(A_1 x + D_1) + 3\,c_2 \operatorname{sech}^2(A_2 x + D_2) \qquad (7.6.5)$$

together with the boundary conditions:

$$\left.\begin{aligned} u\,(o,t) &= u(2,t) = o \\ u_x(o,t) &= u_x(2,t) = o \end{aligned}\right\} \quad \text{for all time} \qquad (7.6.6)$$

These conditions represent two solitons, one with amplitude $3c_1$ sited initially at $x = - D_1/A_1$ and a second with amplitude $3c_2$ placed at $x = - D_2/A_2$. As is well known that the velocity of a soliton depends directly upon its amplitude. So choosing $c_1 > c_2$ and $D_1/A_1 > D_2/A_2$ ensures that these solitary waves interact with increasing time. For comparison with Greig and Morris [26] solution we have chosen $c_1 = 0.3$, $c_2 = 0.1$, $D_1 = D_2 = -6.$, $h = 0.01$, $\Delta t = 0.005$ and $A_j = \frac{1}{2}\left(\frac{\varepsilon c}{\mu}j\right)^{1/2}$, $j=1,2$:

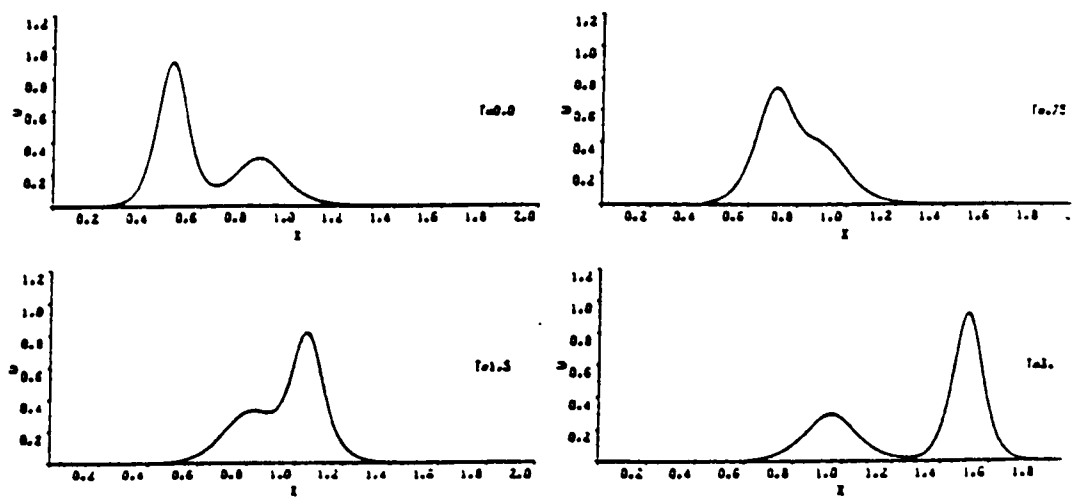

Figure 7.2 Problem (b). The interaction of the two overlapping solitons with $h = 0.01$ $\Delta t = 0.005$.

From Figure 7.2 we observe that the two solitons are placed where the larger one on the left. As the time increases, the larger soliton catches up with the smaller one at time t = 0.75. At time t = 1.5, the larger soliton, overtakes the smaller one and is in the process of separating. Around time t = 3, the interaction process is complete and the larger soliton has sparated completely from the smaller one.

The interaction of two overlapping solitons observed in our computations is shown in Figure 7.2 and agrees well with those obtained by other authors [26,35]. We see that the solitons emerge from the interaction with large and small amplitudes slightly changed from the original by $\approx 0.99\%$, $\approx 0.28\%$ respectively. The agreement with Greig and Morris [26] is very satisfactory.

(b1) Consider the motion of two well separated solitons as an initial condition:

$$u(x,0) = 3\, c_1 \mathrm{sech}^2(A_1 x + D_1) + 3\, c_2 \mathrm{sech}^2(A_2 x + D_2) \qquad (7.6.7)$$

where:

c_1 , c_2 , A_1 , A_2 , D_1 are given in problem (b)

$D_2 = -9.0$, $B_i = \varepsilon\, c_i A_i$, $i = 1$, 2 ,

The boundary conditions are:

$$\left.\begin{aligned} u(0,t) &= u(4,t) = 0 \\ u_x(0,t) &= u_x(4,t) = 0 \end{aligned}\right\} \quad \text{for all time} \qquad (7.6.8)$$

Figure 7.3 shows us that the two solitons with the larger on the left. As the time increases, the larger soliton catches up with the smaller until, at time t = 3. Around time t = 4, the larger soliton has overtaken the smaller one and is in the process of separating. By time t = 6, the interaction process is complete and the larger soliton has separated completely from the smaller

one. Also we see that after the interaction of the two well separated solitons the larger and smaller amplitudes alter from the originals by very small amounts of 0.002%, 0.037% respectively. Therefore we emphasise that after the interaction the amplitudes are unchanged as required by theory:

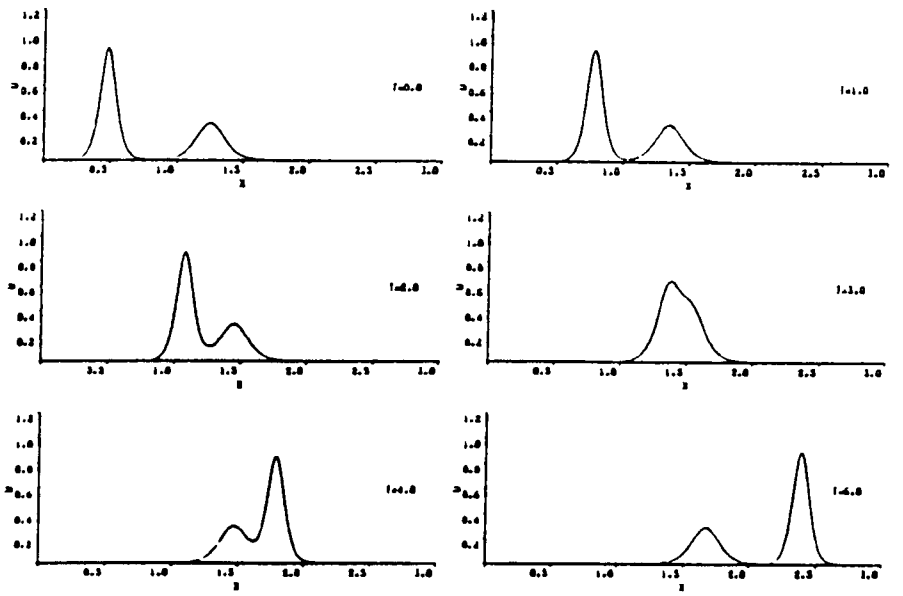

Figure 7.3 Problem (b1). The interaction of two well separated solitons with h = 0.01 Δt = 0.005.

(c) Another interesting initial value problem for the KdV equation is given by using the Gaussian distribution function as the initial condition:

$$u(x,0) = \exp(-x^2) \tag{7.6.9}$$

This is a typical symmetric function which tends zero as $|x|$ tends to infinity. The boundary conditions imposed are:

$$\left.\begin{aligned} u(\mp 15,t) &= 0 \\ u_x(\mp 15,t) &= 0 \end{aligned}\right\} \text{ for all } t > 0 \tag{7.6.10}$$

We choose $\varepsilon = 1.0$ and we discuss each of the following cases:

(c1) $\mu = 0.04$, $h = 0.1$, $\Delta t = 0.01$

(c2) $\mu = 0.005$, $h = 0.05$, $\Delta t = 0.01$

(c3) $\mu = 0.001$, $h = 0.025$, $\Delta t = 0.005$

(c4) $\mu = 0.0005$, $h = 0.025$, $\Delta t = 0.005$

A comparison has been made with the work of Goda [59]. Figure 7.4 depicts the behaviour of numerical solution of problem (c1) for times up to 12.5 . We see that the initial perturbation splits itself into a soliton plus an oscillating tail. The values of the analytic and numerical velocities are $c_a \simeq 0.401$, $c_n \simeq 0.4$ respectively, so they agree:

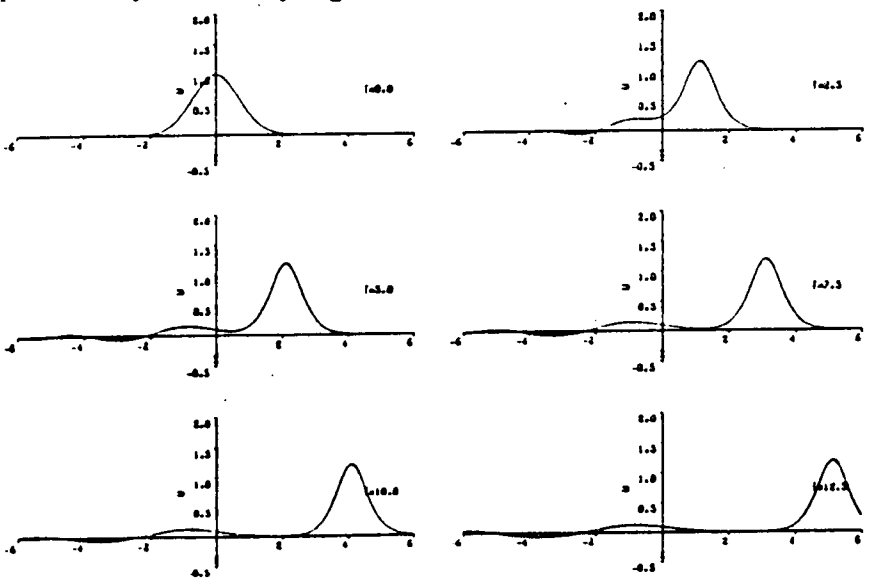

Figure 7.4 Problem (c1). A single soliton with oscillating tail for $\mu = 0.04$.

Figure 7.5 shows similar results for $\mu = 0.005$. We observe that the initial perturbation breaks up into four solitons. The graphs obtained by our algorithm for the case (c1) is identical with that given by Goda [59]. The agreement between the analytic velocity $c_a \simeq 0.5589$ and the observed velocity $c_n \simeq 0.558$ for the leading soliton is very satisfactorily:

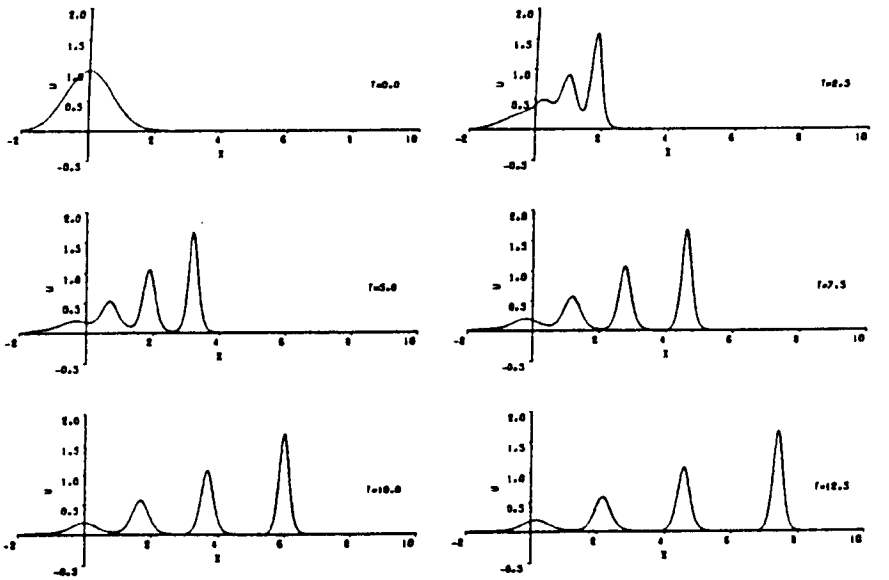

Figure 7.5 Problem (c2). The breakdown of the initial condition into 4 soliton when $\mu = 0.005$.

In Figure 7.6, for $\mu = 0.001$, we see that the initial perturbation breaks up into 9 solitons moving to the right. The agreement between the analytic velocity $c_a \simeq 0.616$ and the observed velocity $c_n \simeq 0.62$ for the leading soliton is very satisfactorily:

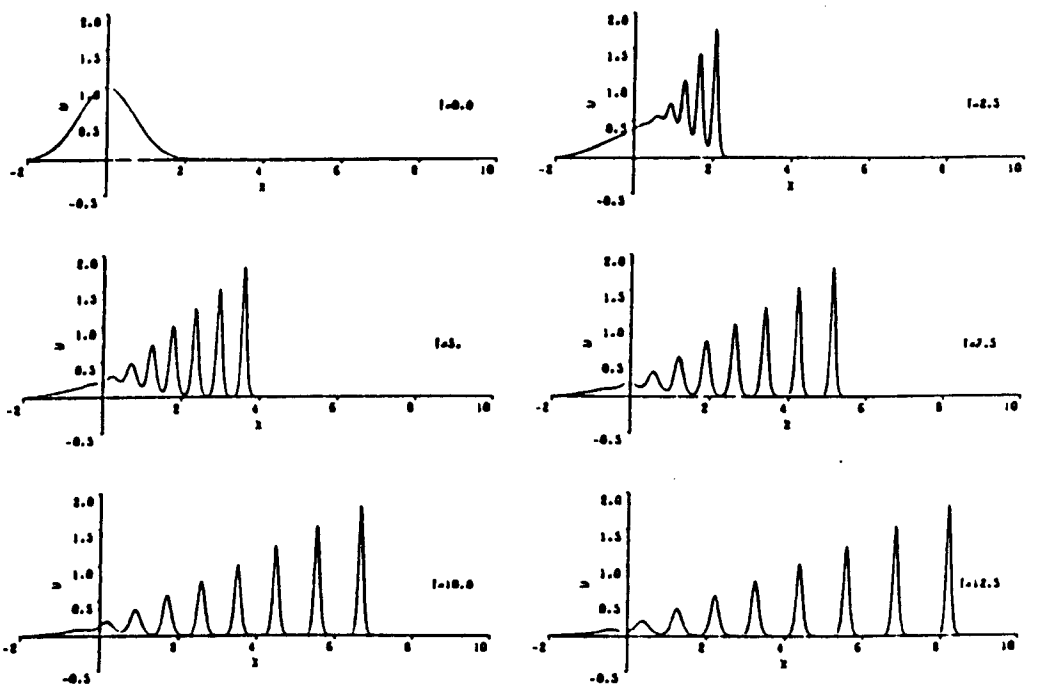

Figure 7.6 Problem (c3). The breakdown of the initial perturbation into 9 solitons with $\mu = 0.001$.

Figure 7.7, for $\mu = 0.0005$, shows another train of solitons generated when the initial perturbation splits into 12 solitons moving to the right with constant velocity, for the leading soliton this is $c_n \simeq 0.64$ and amplitude ($\simeq 1.91$). The analytic velocity $c_a \simeq 0.637$ agrees very well with the numerical one c_n:

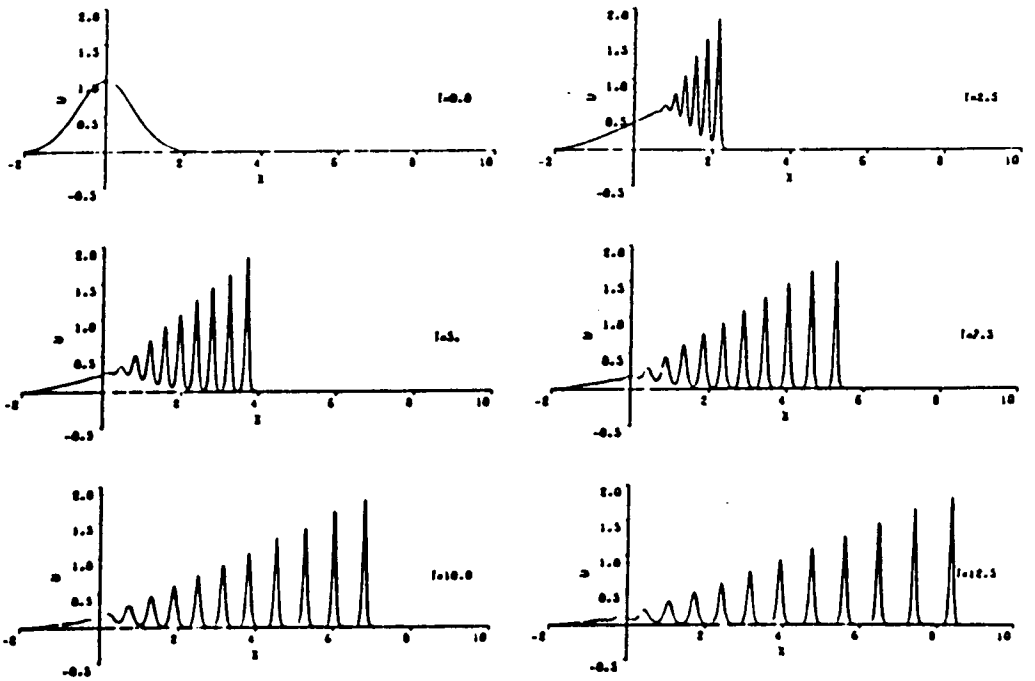

Figure 7.7 Problem (c4). The breakdown of the initial condition into 12 solitons when $\mu = 0.0005$.

It was found that the behaviour of the numerical solutions differed according to the value of μ . The initial perturbation breaks up into a number of solitons in the course of time depending on the value of μ chosen. So, if we decrease the value of μ then the number of solitons, the amplitude, and the velocity increase. Also. it appears that the amplitudes vary approximately linearly.

(d) As a final test example we shall as take initial condition:

$$u(x,0) = \frac{1}{2}\left[1 - \tanh\left[\frac{x - 25}{5}\right]\right] \qquad (7.6.11)$$

and as boundary conditions:

$$\left.\begin{aligned} u(0,t) &= 1 \\ u(50,t) &= 0 \\ u_x(0,t) &= u_x(50,t) = 0 \end{aligned}\right\} \text{ for all } t > 0 \qquad (7.6.12)$$

(d1) Let us consider the symetric initial condition as given by:

$$u(x,0) = \frac{1}{2}\left[1 - \tanh\left[\frac{|x| - 25}{5}\right]\right] \qquad (7.6.13)$$

and the boundary conditions are impose:

$$\left.\begin{aligned} u(-150,t) &= u(150,t) = 0 \\ u_x(-150,t) &= u_x(150,t) = 0 \end{aligned}\right\} \text{ for all } t > 0 \qquad (7.6.15)$$

To allow comparison with Vliegenthart [44] for problem (d) we have chosen $\varepsilon = 0.2$, $\mu = 0.1$ and used $\Delta t = 0.05$ and $h = 0.4$. The solution we compute, reproduced in Figure 7.8 shows us all the general features obtained in the earlier solution [44]. We cannot make a direct comparison with Vliegenthart's [44] figures because the boundary conditions used are not given. We can, however, repeat his computations using his finite difference scheme and parameters together with our boundary conditions to produce comparable figures. If these are plotted also on Figure 7.8 the graphs are indistinguishable with those obtained in the present study:

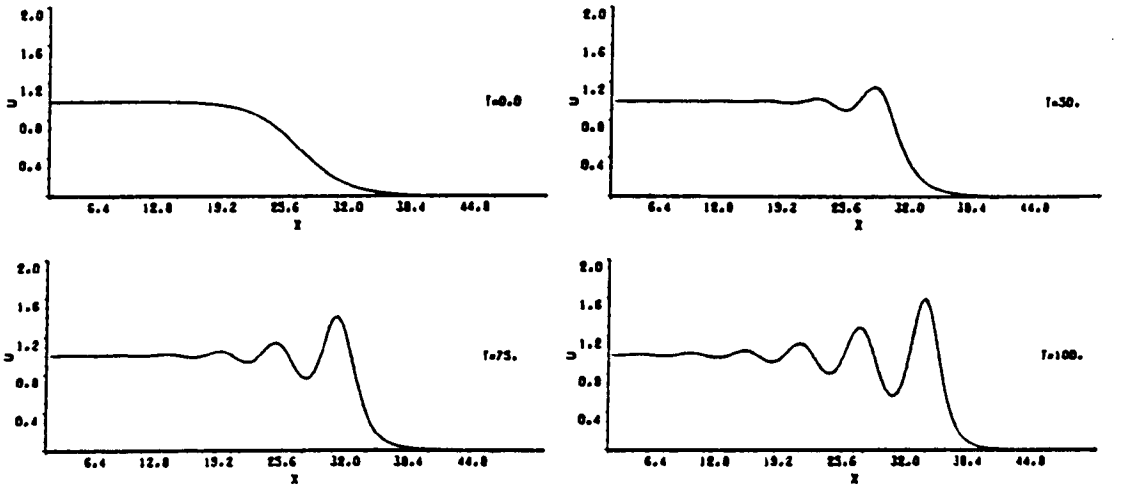

Figure 7.8 Problem (d). The solution graphs for varoius times with h = 0.4, Δt = 0.05, μ = 0.1, ε = 0.2.

We found that his numerical velocities for the solitons were greater than ours and also that the amplitudes differed. Also, the conservative quantities varied somewhat. We suspected the boundary conditions. So, we decided to chose alternative problem (d1). The behaviour of this solution is given in Figure 7.9:

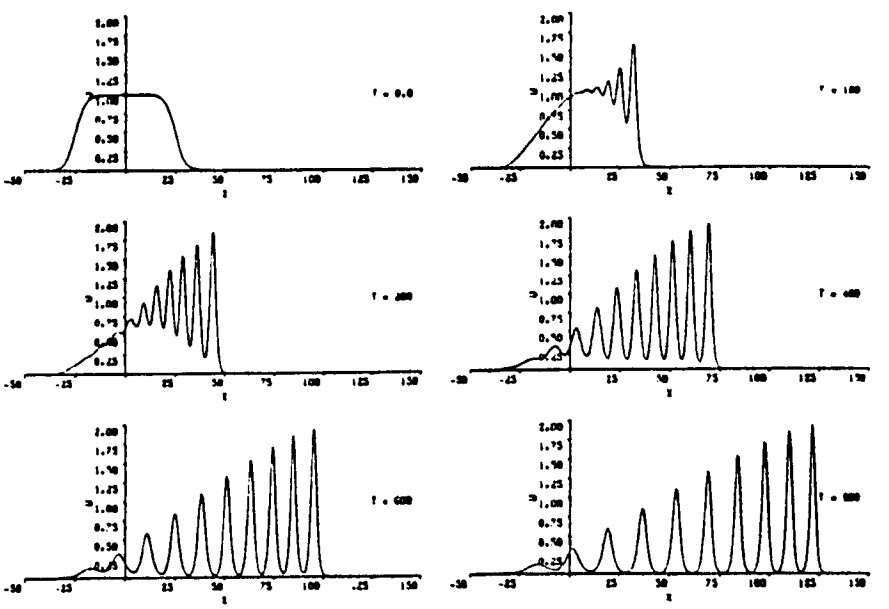

Figure 7.9 Problem (d1).The initial perturbation splits into train of solitons with ε = 0.2, μ = 0.1, h = 0.4, Δt = 0.05.

It is observed from Figure 7.9 that the initial perturbation problem (d1) as broken up into a train of solitons, which move steadily to the right with constant speeds whose magnitude depends upon their individual amplitude. It appears that the amplitudes of the solitons vary approximately linearly. The agreement between the values of the analytic velocity $c_a \simeq 0.1302$ and the numerical velocity $c_n \simeq 0.128$ for the leading soliton are very satisfactory; especially with these long time and large space steps.

7.7 Discussion:

Any numerical scheme for computing the solution of the **KdV** equation must represent faithfully the amplitude and the position of solution over many time steps with minimum errors [36], and also it should be conservative.

To examine the accuracy of our numerical scheme we have used the L_2- and L_∞-error norms to compare the numerical and exact solutions. The L_2-error is used to compare 5 numerical methods in Table 7.2 for the single soliton problem [36]. We find that the collocation quintic spline method compares well in the accuracy with the best of the other methods. In fact, the error in the single soliton solution is less than that of the modified Petrov-Galerkin method by a factor 10, when $h = 0.01$, $\Delta t = 0.005$.

Table 7.2

The growth of the discrete L_2-error norm x 10^3 for single soliton

Time	Zabusky-Kruskal [27]	Hopscotch [26]	Petrov-Galerkin [36]	Modified P-G [36]	Collocation Quintic [present]
$\Delta x = 0.05$		$\Delta t = 0.025$	$h = 0.05$	$\Delta t = 0.025$	
0.25	34.64	61.21	81.39	52.15	37.39
0.50	122.68	122.41	102.54	64.90	38.97
0.75	210.44	181.35	125.84	89.01	52.97
1.00	298.19	228.10	150.57	107.20	63.72
			$h = 0.033$	$\Delta t = 0.01$	
0.25			31.18	5.94	2.35
0.5			43.35	7.56	3.25
0.75			56.21	8.70	2.23
1.00			74.08	9.49	2.88
$\Delta x = 0.01$		$\Delta t = 0.0005$	$h = 0.01$	$\Delta t = 0.005$	
0.25	5.94	3.79	4.46	0.21	0.022
0.50	13.17	9.28	7.01	0.38	0.041
0.75	21.08	14.14	10.08	0.57	0.054
1.00	28.66	18.72	13.26	0.74	0.067

Table 7.3 shows us that the L_∞-error norm computed from our technique for a single soliton using the definition (4.5.2) has been compared with all the method which are quoted in Table 7.3. We observe that the L_∞-error norm is greater than L_2-error norm which disagrrees with the authors [26,27,36]. Also we find that the value of L_∞-error norm using our method is greater than Modified Petrov-Galerkin when $h = 0.05$, $\Delta t = 0.025$ and it has the same error of magnitude for $h = 0.033$, $\Delta t = 0.01$ and $h = 0.01$, $\Delta t = 0.005$:

Tal 7.3

The growth of the discrete L_∞-error norm x 10^3 for single soliton

Time	Zabusky-Kruskal [27]	Hopscotch [26]	Petrov-Galerkin [36]	Modified P-G [36]	Collocation Quintic [present]
Δx = 0.05		Δt = 0.025	h = 0.05	Δt = .025	
0.25	19.4	32.7	42.18	30.22	57.54
0.50	63.5	67.4	51.85	22.85	65.26
0.75	122.4	99.3	87.60	35.86	76.69
1.00	161.4	141.6	100.41	39.39	100.96
			h = 0.033	Δt = .01	
0.25			14.27	2.80	4.80
0.5			21.65	4.53	4.83
0.75			29.78	4.85	4.83
1.00			39.37	5.85	4.83
Δx = 0.01		Δt = 0.0005	h = 0.01	Δt = .005	
0.25	2.05	1.11	1.21	0.07	0.07
0.50	4.22	2.14	2.15	0.11	0.11
0.75	6.36	3.54	3.09	0.17	0.16
1.00	8.13	4.91	3.83	0.21	0.20

Further results for times up to t = 3.0 are listed in Table 7.4:

Table 7.4

errors x 10^3 for a single soliton

T	1.25	1.50	1.75	2.00	2.25	2.50	2.75	3.00
L_2 h = 0.033 Δt = 0.01	3.62	3.31	2.55	2.35	2.10	2.89	3.11	2.94
L_∞	6.47	8.14	8.14	8.14	8.14	8.14	8.14	8.14
L_2 h = .01 Δt=0.005	0.08	0.09	0.09	0.10	0.12	0.13	0.14	0.15
L_∞	0.22	0.24	0.25	0.28	0.32	0.33	0.37	0.41

Table 7.4 shows us that the error is still small even when the time is increased up to t = 3.0.

The **KdV** equation has an infinite number of conservative quantities. For this reason it is important for any proposed numerical scheme for solving the **KdV** equation to have at least the lower order quantities conserved. We will study how the four quantities I_i $(i = 1,\ldots,4)$ defined by equations (2.4.8)-(2.4.11) respectively behave.

We have computed the first four invariant quantities for the single soliton solution. These are given in Tables 7.5 and 7.6:

Table 7.5

The computed value I_1 , I_2 , I_3 for a single soliton

	I_1		I_2		I_3	
Time	h = .033 Δt = .01	h = .01 Δt =.005	h = .033 Δt = .01	h = .01 Δt =.005	h = .033 Δt = .01	h = .01 Δt=.005
0.0	0.144597	0.144598	0.086759	0.086759	0.046850	0.046850
0.50	0.144599	0.144602	0.086785	0.086761	0.046871	0.046851
1.00	0.144562	0.144601	0.086794	0.086762	0.046878	0.046853
1.50	0.144584	0.144603	0.086788	0.086764	0.046876	0.046854
2.00	0.144570	0.144604	0.086767	0.086765	0.046861	0.046855
2.50	0.144635	0.144605	0.086771	0.086767	0.046864	0.046857
3.00	0.144604	0.144606	0.086783	0.086768	0.046873	0.046858

In Table 7.5 we give the values of I_1 , I_2 , I_3 . The change in these quantities during the computer run are less than 0.027% , 0.041% , 0.060% , respectively for h = 0.033 , Δt = 0.01 and 0.006% , 0.011% , 0.018% respectively for h = 0.01 , Δt = 0.005 . We observe that they are satisfactorily constant even when h, Δt are relatively large.

Table 7.6 shows us that I_4 is almost constant when h = 0.01 , Δt = 0.005:

Table 7.6

The numerical value of I_4 for a single soliton

T	0.0	.50	1.00	1.50	2.00	2.50	3.00
h =0.033 Δt =0.01	.024089	.025466	.024955	.025234	.024438	.024693	.024475
h = 0.01 Δt=0.005	.024094	.024101	.024101	.024098	.024099	.024100	.024104

We find that the change in I_4 is less than 5.72% when h = 0.033 Δt = 0.01 and 0.042% when h = 0.01 Δt = 0.005 during the computer run.

A computer run on the single soliton solution with 200 nodes and 200 time steps took 4 secs of CPU time on a VAX 8650.

With example (b) we have verified that our numerical method can adequately cope when two overlapping solitary waves coalesce for a short period and then separate with their original profiles intact but their large and small amplitudes affected by 0.99% , 0.28% respectively and their relative positions changed. The quantities I_1 , I_2 , I_3 and I_4 have been computed for problem (b) and are listed in Table 7.7:

Table 7.7

The computed values I_1 , I_2 , I_3 , and I_4 for two overlapping solitons with h = 0.01 Δt = 0.005

Time	I_1	I_2	I_3	I_4
0.0	0.228081	0.107062	0.053316	0.027083
0.50	0.228081	0.107064	0.053317	0.030863
1.00	0.228262	0.107074	0.053321	0.034886
1.50	0.228222	0.107075	0.053323	0.031270
2.00	0.228037	0.107070	0.053323	0.027651
2.50	0.227816	0.107073	0.053325	0.026696
3.00	0.227658	0.107079	0.053328	0.026292

Also Table 7.7 indicates to us that the quantities I_i $(i = 1,\ldots,4)$ are changed about 0.186% , 0.016% , 0.023% and 28.82% respectively during the computer run.

The discrete L_2-error norm is evaluated for the two well separated solitons problem (b1) up to t = 2.0 and is given in Table 7.8:

Table 7.8

The growth of the discrete L_2-error norm x 10^3 for two well separated solitons with h = 0.01 Δt = 0.005

T	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00
$L_2 \times 10^3$	0.029	0.090	0.195	0.421	0.883	1.824	3.740	7.598

We have also calculated the first four conservative quantities for problem (b1) which are recorded in Table 7.9:

Table 7.9

The computed values I_1 , I_2 , I_3 , and I_4 for two well separated solitons with h = 0.01 Δt = 0.005

Time	I_1	I_2	I_3	I_4
0.0	0.228082	0.103466	0.049864	0.024616
1.0	0.228085	0.103467	0.049865	0.024617
2.0	0.228086	0.103468	0.049865	0.024617
3.0	0.228088	0.103470	0.049866	0.024628
4.0	0.228091	0.103473	0.049869	0.024620
5.0	0.228092	0.103474	0.049870	0.024642
6.0	0.228093	0.103475	0.049870	0.024622
7.0	0.288094	0.103476	0.049870	0.024628
8.0	0.228095	0.103476	0.049870	0.024648

Table 7.9 shows us that the quantities I_i $(i = 1,\ldots,4)$ change by less than 0.006% , 0.01% , 0.013% , 0.13% respectively during the computer run. So they can be considered as constant.

Using equation (2.3.4.8) the forward and backward phase shifts have been evaluated numerically and obtained as

$$\Delta_1 = 0.11 , \quad \text{and} \quad \Delta_2 = -0.18$$

which agree with the analytic results equation (4.5.7) .

(b2) Let us study the two solitary waves initial conditions which are followed from the analytic solutions (2.3.4.1) when t = 0.0 in the following cases:

(i) problem (b) where $\alpha_i = \sqrt{c_i/\mu}$, $d_1 = -12$, $d_2 = -12 + \Delta$

(ii) problem **(b1)** where $\alpha_i = \sqrt{c_i/\mu}$, $d_1 = -12$, $d_2 = -18 + \Delta$

(iii) $\alpha_1 = 4.0$, $\alpha_2 = 2.0$, $d_1 = d_2 = 0.0$. The boundary conditions are chosen as

$$\left.\begin{aligned} u(\mp 12, t) &= 0 \\ u_x(\mp 12, t) &= 0 \end{aligned}\right\} \quad \text{for } -0.5 \leq t \leq 0.5 \qquad (7.7.1)$$

The present technique has been used to compute the errors in the solution using the discrete L_2- and L_∞-error norms and also the firs four conservative quantities which are given in Table 7.10:

Table 7.10

The computed values of the errors $x10^3$, and I_1 to I_4 for two overlapping soliton problem (b1) case (i) with h = 0.01, Δt =0.005

Time	$L_2 x10^3$	$L_\infty x10^3$	I_1	I_2	I_3	I_4
0.0			0.228074	0.103456	0.049855	0.024610
0.50	0.217	0.309	0.228096	0.103457	0.049856	0.025336
1.00	0.081	0.309	0.228097	0.103459	0.049857	0.024655
1.50	0.114	0.309	0.228101	0.103461	0.049859	0.024743
2.00	0.100	0.309	0.228099	0.103462	0.049860	0.024619
2.50	0.128	0.309	0.228100	0.103463	0.049860	0.024689
3.00	0.172	0.452	0.228094	0.103463	0.049860	0.024699

Table 7.10 shows us that the errors measured in terms of the discrete L_2- and L_∞-error norms are satisfactorily small even when the time achieves 3. The quantities I_i $(i = 1,\ldots,4)$ are changed by less than 0.012% , 0.0068% , 0.011% , 0.541% respectively during the computer run. These quantities are virtually constants especially I_1, I_2, I_3. After the collision the larger and smaller amplitudes are changed from their original by values less than 0.8% and 0.4% respectively.

The L_2- and L_∞-error norms and the first four conservative quantities have been computed for two well separated solitons problem (b2) case (ii) and are listed in Table 7.11:

Table 7.11

The computed values of the errors x 10^3, and I_1 to I_4 for a double soliton problem(b2) case (ii) with h=.01, Δt = 0.005

Tim	$L_2 \times 10^3$	$L_\infty \times 10^3$	I_1	I_2	I_3	I_4
0.0			0.228082	0.103456	0.049855	0.024610
1.0	0.078	0.225	0.228084	0.103457	0.049856	0.024612
2.0	0.142	0.375	0.228086	0.103458	0.049856	0.024611
3.0	0.141	0.375	0.228089	0.103460	0.049857	0.024617
4.0	0.238	0.621	0.228093	0.103463	0.049860	0.024619
5.0	0.321	0.799	0.228095	0.103464	0.049861	0.024618
6.0	0.386	1.016	0.228096	0.103465	0.049861	0.024632
7.0	0.448	1.170	0.228097	0.103466	0.049861	0.024614
8.0	0.512	1.332	0.228098	0.103466	0.049861	0.024619

Table 7.11 shows us that the L_2- and L_∞-error norms are satisfactorily small for two well separated solitons even with runs up to a time of t = 8. The quantities I_i $(i = 1,\ldots,4)$ are changed by less than 0.0071%, 0.0097%, 0.0121%, 0.0894% respectively during the computer run. These quantities are virtually constants. After the interaction the larger and smaller amplitudes are altered from their original by values less than 0.008% and 0.045% respectively.

For the two well separated solitons with large amplitudes problem (b2) case (iii), the values of the L_2- and L_∞-error norms and the first four conservative quantities are given in Table 7.12:

Table. 7. 12

The computed values of the errors x 10^3, and I_1 to I_4 for a double soliton problem (b2) case (iii) with h = 0. 1, Δt =. 0005

Time	$L_2 x10^3$	$L_\infty x10^3$	I_1	I_2	I_3	I_4
-0. 5			11. 99991	47. 99998	211. 2000	943. 5421
-0. 4	1. 937	1. 470	12. 00011	48. 00021	211. 2014	944. 0864
-0. 3	1. 520	1. 814	11. 99994	48. 00042	211. 2023	943. 7606
-0. 2	1. 812	1. 814	11. 99981	48. 00063	211. 2036	944. 0063
-0. 1	2. 890	1. 816	11. 99982	48. 00094	211. 2062	944. 9952
0. 0	1. 292	1. 816	12. 00005	48. 00158	211. 2134	943. 7791
0. 1	1. 395	1. 816	12. 00029	48. 00239	211. 2170	943. 7485
0. 2	2. 088	1. 975	12. 00035	48. 00280	211. 2181	943. 8973
0. 3	2. 559	2. 743	12. 00041	48. 00303	211. 2190	943. 8876
0. 4	3. 205	3. 400	12. 00068	48. 00323	211. 2200	944. 1334
0. 5	3. 704	4. 437	12. 00060	48. 00343	211. 2209	943. 9015

We see that over the computer run the method has coped very well with this problem. Table 7.12 shows us that the L_2- and L_∞-error norms are satisfactorily with large amplitudes. Also the quantities I_i $(i = 1, \ldots, 4)$ are changed by less than 0.007%, 0.0072%, 0.01, 0.155% respectively. Therefore, we conclude that these quantities are virtually constant, particularly I_1, I_2, I_3. After the collision of these two well separated solitons (problem (b2) case (iii)) the larger and smaller amplitudes are changed from their original values by 0.96% and 0.006% respectively.

Similar results are given in Table 7.13 for the conservative quantities I_1, I_2, I_3 of problems (c1) and (c2). We found that each of the quantities I_i are very satisfactorly constants:

Table 7.13

The computed values of I_1 , I_2 and I_3 for $u(x,0) = \exp(-x^2)$

Time	I_1		I_2		I_3	
	μ =.04	μ = .005	μ = .04	μ =.005	μ =.04	μ = .005
0.0	1.772454	1.772454	1.253314	1.253314	0.872929	1.004527
2.5	1.772475	1.772475	1.253344	1.253345	0.872955	1.004562
5.0	1.772488	1.772496	1.253371	1.253376	0.872984	1.004594
7.5	1.772537	1.772518	1.253400	1.253407	0.873015	1.004635
10.0	1.772525	1.772542	1.253424	1.253438	0.873046	1.004677
12.5	1.772536	1.772561	1.253481	1.253469	0.873082	1.004719

Table 7.13 shows that the quantities I_i $(i = 1,\ldots,4)$ are changed by less than 0.005% , 0.014% , 0.018% respectively when μ = 0.04 and 0.006% , 0.0124% , 0.0192% respectively when μ = .005 during the computer run. Therefore they can be considered as invariant.

Table 7.13a gives the numerical value of I_4:

Table 7.13a

The computed value of I_4 for $u(x,0) = \exp(-x^2)$

T	0.0	2.50	5.00	7.50	10.00	12.50
μ = .04	0.602077	0.602117	0.603916	0.606560	0.609052	0.617424
μ = .005	0.845971	0.846186	0.846324	0.846377	0.846432	0.846548

Table 7.13a shows us that the change in the quantity I_4 is less than 2.55% when μ = 0.04 and 0.0682% when μ = .005 during the computer run.

From Tables 7.13 , 7.13a we observe that the four computed conserved quantities are constants and have magnitude dependent on the coefficient of the dispersive term (i.e. the value of μ).

The total number of solitons which are generated from a Gaussian initial condition has been determined using

equation (4.5.6) for different values of μ and we found an agreement with those given in the above Figures 7.4 - 7.7 .

The first four conservative quantities for problem (d1) with boundary conditions (7.6.14) are given in Table 7.14 up to time t = 800:

Table 7.14

The computed values I_1 , I_2 , I_3 , and I_4 for problem (d1) boundary conditions (7.6.14) with h = 0.4, Δt = 0.05, ε = 0.2, μ = 0.1

Time	I_1	I_2	I_3	I_4
000.0	50.00022	45.00045	42.30069	40.44194
100.0	50.00456	45.00827	42.31154	40.45628
200.0	50.00883	45.01594	42.32193	40.47181
300.0	50.01307	45.02349	42.33296	40.48888
400.0	50.01719	45.03096	42.34436	40.51368
500.0	50.02153	45.03846	42.35592	40.57836
600.0	50.02568	45.04593	42.36752	40.57206
700.0	50.03011	45.05344	42.37922	40.61868
800.0	50.03298	45.06098	42.39087	41.20851

Table 7.14 shows us that the quantities I_i $(i = 1,\ldots,4)$ change by less than 0.066% , 0.135% , 0.214% , 1.896% respectively during this long computer run. So they can be considered as relatively constant.

Finally, we conclude that the collocation method using quintic polynomial spline interpolation functions is a suitable technique for the computation of KdV equation solutions over long periods of time with small space and time steps.

CHAPTER 8

A COLLOCATION METHOD FOR THE GENERALISED EQUATION

8.1 Introduction:

The generalised **KdV** equation has been studied analytically by several authors [6,32] and numerical solutions using finite difference methods [6,32,80,88], a fourier or psedospectral method [32] and Galerkin methods have been presented [35,48].

Our aim in this chapter is to compute the finite element solution of the generalised **KdV** equation using collocation with quintic splines as interpolation functions.

8.2 The Governing Equation:

We seek to solve numerically the generalised Korteweg-de Vries equation, in the normalised form:

$$u_t + \varepsilon u^p u_x + \mu u_{xxx} = 0 \qquad a \le x \le b \qquad (8.2.1)$$

where; p ($p = 1,2,\ldots$) is positive integer, ε and μ are positive parameters and the subscripts t and x denote differentiation. Appropriate boundary conditions will be chosen from the following:

$$\left.\begin{aligned} u(a,t) &= \beta_1 \\ u(b,t) &= \beta_2 \\ u_x(a,t) &= u_x(b,t) = 0 \\ u_{xx}(a,t) &= u_{xx}(b,t) = 0 \end{aligned}\right\} \text{ for all } t > 0 \qquad (8.2.2)$$

and the initial conditions to be used will be prescribed later, in section 8.6 .

8.3 The Collocation Solution [52,82,84,85,86,87,88]:

We intend to use quintic **B**-splines to approximate the solution $u(x,t)$ of equation (8.2.1). If we apply the collocation method to equation (8.2.1), we obtain:

$$u_{Nt}(x_j,t) + \varepsilon\, u_N^p(x_j,t)\, u_{Nx}(x_j,t) + \mu\, u_{Nxxx}(x_j,t) = 0$$

$$j = 0,1, \ldots ,N \qquad (8.3.1)$$

Using the definition and properties of quintic B-splines described in section 7.3, equation (8.3.1) becomes:

$$\sum_{i=-2}^{N+2} \phi_i(x_j)\, \dot{\delta}_i + \varepsilon \sum_{i=-2}^{N+2} \phi_i'(x_j)\, \delta_i \left[\sum_{k=-2}^{N+2} \phi_k(x_j)\, \delta_k \right]^p + \mu \sum_{i=-2}^{N+2} \phi_i'''(x_j)\, \delta_i = 0 \qquad j = 0, \ldots ,N \qquad (8.3.2)$$

Suppose that δ_i is linearly interpolated between two time levels n and $n+1$ by:

$$\delta_i = (1-\theta)\delta_i^n + \theta\, \delta_i^{n+1} \qquad (8.3.3)$$

where $0 \le \theta \le 1$ and δ_i^n are the parameters at the time $n\Delta t$. The time derivative is discretised using the standard finite difference formula:

$$\frac{d\delta}{dt}_i = \frac{1}{\Delta t}\,(\delta_i^{n+1} - \delta_i^n) \qquad (8.3.4)$$

Hence equation (8.3.2) can be written as:

$$\sum_{i=-2}^{N+2} \left[\phi_i + \theta\, \Delta t(\varepsilon\, \phi_i' \left\{ \sum_{k=-2}^{N+2} \phi_k\, \delta_k^n \right\}^p + \mu\, \phi_i''') \right] \delta_i^{n+1}$$

$$= \sum_{i=-2}^{N+2} \left[\phi_i - (1-\theta)\Delta t(\varepsilon\, \phi_i' \left\{ \sum_{k=-2}^{N+2} \phi_k\, \delta_k^n \right\}^p + \mu\, \phi_i''') \right] \delta_i^n \qquad (8.3.5)$$

where the basis functions and their derivatives are evaluated at the $N+1$ knots x_j , $j = 0,1, \ldots ,N$. Giving the parameter θ the values 0 , $\frac{1}{2}$, 1 produces explicit, Crank-Nicolson and backward difference schemes respectively.

In the present analysis we will take $\theta = \frac{1}{2}$ so that equation (8.3.5) takes the particular form:

$$\sum_{i=-2}^{N+2} \left[\phi_i + \frac{\Delta t}{2} \left(\varepsilon \phi_i' \left\{ \sum_{k=-2}^{N+2} \phi_k \delta_k^n \right\}^p + \mu \phi_i''' \right) \right] \delta_i^{n+1}$$

$$= \sum_{i=-2}^{N+2} \left[\phi_i - \frac{\Delta t}{2} \left(\varepsilon \phi_i' \left\{ \sum_{k=-2}^{N+2} \phi_k \delta_k^n \right\}^p + \mu \phi_i''' \right) \right] \delta_i^n \qquad (8.3.6)$$

Using the properties of quintic splines, equation (8.3.6) can be calculated at the knots x_j , $j = 0,1, \ldots ,N$, so that at $x = x_0$ we have:

$$\alpha_{01} \delta_{-2}^{n+1} + \alpha_{02} \delta_{-1}^{n+1} + \alpha_{03} \delta_0^{n+1} + \alpha_{04} \delta_1^{n+1} + \alpha_{05} \delta_2^{n+1} =$$

$$\alpha_{05} \delta_{-2}^n + \alpha_{04} \delta_{-1}^n + \alpha_{03} \delta_0^n + \alpha_{02} \delta_1^n + \alpha_{01} \delta_2^n \qquad (8.3.7)$$

where:

$$\alpha_{01} = 1 - R_1 Z_{-2}^p - R_2 \ , \quad \alpha_{02} = 26 - 10R_1 Z_{-2}^p + 2R_2 \ , \quad \alpha_{03} = 66$$

$$\alpha_{04} = 26 + 10R_1 Z_{-2}^p - 2R_2 \ , \quad \alpha_{05} = 1 + R_1 Z_{-2}^p + R_2$$

$$Z_{-2} = \delta_{-2} + 26\delta_{-1} + 66\delta_0 + 26\delta_1 + \delta_2 \ , \ R_1 = \frac{5}{2h} \varepsilon \Delta t \ , \ R_2 = \frac{30}{h^3} \mu \Delta t \qquad (8.3.8)$$

and at $x = x_1$ equation (8.3.6) becomes:

$$\alpha_{11} \delta_{-1}^{n+1} + \alpha_{12} \delta_0^{n+1} + \alpha_{13} \delta_1^{n+1} + \alpha_{14} \delta_2^{n+1} + \alpha_{15} \delta_3^{n+1} =$$

$$\alpha_{15} \delta_{-1}^n + \alpha_{14} \delta_0^n + \alpha_{13} \delta_1^n + \alpha_{12} \delta_2^n + \alpha_{11} \delta_3^n \qquad (8.3.9)$$

where:

$$\alpha_{11} = 1 - R_1 Z_{-1}^p - R_2 \ , \quad \alpha_{12} = 26 - 10R_1 Z_{-1}^p + 2R_2 \ , \quad \alpha_{13} = 66$$

$$\alpha_{14} = 26 + 10R_1 Z_{-1}^p - 2R_2 \ , \quad \alpha_{15} = 1 + R_1 Z_{-1}^p + R_2$$

$$Z_{-1} = \delta_{-1}^n + 26\delta_0^n + 66\delta_1^n + 26\delta_2^n + \delta_3^n \qquad (8.3.10)$$

.

.

.

at $x = x_N$ equation (8.3.6) becomes:

$$\alpha_{N1}\,\delta_{N-2}^{n+1} + \alpha_{N2}\,\delta_{N-1}^{n+1} + \alpha_{N3}\,\delta_{N}^{n+1} + \alpha_{N4}\,\delta_{N+1}^{n+1} + \alpha_{N5}\,\delta_{N+2}^{n+1} =$$

$$\alpha_{N5}\,\delta_{N-2}^{n} + \alpha_{N4}\,\delta_{N-1}^{n} + \alpha_{N3}\,\delta_{N}^{n} + \alpha_{N2}\,\delta_{N+1}^{n} + \alpha_{N1}\,\delta_{N+2}^{n} \qquad (8.3.11)$$

where:

$$\alpha_{N1} = 1 - R_1\,Z_{N-2}^{p} - R_2\,, \quad \alpha_{N2} = 26 - 10R_1\,Z_{N-2}^{p} + 2R_2\,, \quad \alpha_{N3} = 66$$

$$\alpha_{N4} = 26 + 10R_1\,Z_{N-2}^{p} - 2R_2\,, \quad \alpha_{N5} = 1 + R_1\,Z_{N-2}^{p} + R_2$$

$$Z_{N-2} = \delta_{N-2}^{n} + 26\delta_{N-1}^{n} + 66\delta_{N}^{n} + 26\delta_{N+1}^{n} + \delta_{N+2}^{n} \qquad (8.3.12)$$

Generally, these equations can be written as a recurrence relationship:

$$\alpha_{i1}\,\delta_{i-2}^{n+1} + \alpha_{i2}\,\delta_{i-1}^{n+1} + \alpha_{i3}\,\delta_{i}^{n+1} + \alpha_{i4}\,\delta_{i+1}^{n+1} + \alpha_{i5}\,\delta_{i+2}^{n+1} =$$

$$\alpha_{i5}\,\delta_{i-2}^{n} + \alpha_{i4}\,\delta_{i-1}^{n} + \alpha_{i3}\,\delta_{i}^{n} + \alpha_{i2}\,\delta_{i+1}^{n} + \alpha_{i1}\,\delta_{i+2}^{n} \qquad (8.3.13)$$

where:

$i = 0\,,\ 1\,,\ \ldots\,,\ N$

$$\alpha_{i1} = 1 - R_1\,Z_{i-2}^{p} - R_2\,, \quad \alpha_{i2} = 26 - 10R_1\,Z_{i-2}^{p} + 2R_2\,, \quad \alpha_{i3} = 66$$

$$\alpha_{i4} = 26 + 10R_1\,Z_{i-2}^{p} - 2R_2\,, \quad \alpha_{i5} = 1 + R_1\,Z_{i-2}^{p} + R_2$$

$$Z_{i-2} = \delta_{i-2}^{n} + 26\delta_{i-1}^{n} + 66\delta_{i}^{n} + 26\delta_{i+1}^{n} + \delta_{i+2}^{n}$$

$$R_1 = \frac{5}{2h}\,\varepsilon\Delta t\,, \quad R_2 = \frac{30}{h^3}\,\mu\Delta t \qquad (8.3.14)$$

The system (8.3.13) consists of $N+1$ linear equations in $N+5$ unknowns $(\delta_{-2}, \delta_{-1}, \delta_0, \ldots, \delta_{N+1}, \delta_{N+2})^T$. To obtain a unique solution to this system we need 4 additional constraints. These are obtained from the boundary conditions (8.2.2) which require that:

$$\left.\begin{aligned} \delta_{-2} + 26\delta_{-1} + 66\delta_0 + 26\delta_1 + \delta_2 &= \beta_1 \\ -5\delta_{-2} - 50\delta_{-1} + 50\delta_1 + 5\delta_2 &= 0 \\ \delta_{N-2} + 26\delta_{N-1} + 66\delta_N + 26\delta_{N+1} + \delta_{N+2} &= \beta_2 \\ -5\delta_{N-2} - 50\delta_{N-1} + 50\delta_{N+1} + 5\delta_{N+2} &= 0 \end{aligned}\right\} \qquad (8.3.15)$$

By solving the first two equations of (8.3.15) simultaneously for δ_{-2} and δ_{-1}, we obtain:

$$\left.\begin{aligned} \delta_{-2} &= -\frac{5}{8}\beta_1 + \frac{165}{4}\delta_0 + \frac{65}{2}\delta_1 + \frac{9}{4}\delta_2 \\ \delta_{-1} &= \frac{1}{16}\beta_1 - \frac{33}{8}\delta_0 - \frac{9}{4}\delta_1 - \frac{1}{8}\delta_2 \end{aligned}\right\} \qquad (8.3.16)$$

Similarly, solving the last two equations of (8.3.15) simultaneously for δ_{N+1}, δ_{N+2}, gives:

$$\left.\begin{aligned} \delta_{N+2} &= -\frac{5}{8}\beta_2 + \frac{165}{4}\delta_N + \frac{65}{2}\delta_{N-1} + \frac{9}{4}\delta_{N-2} \\ \delta_{N+1} &= \frac{1}{16}\beta_2 - \frac{33}{8}\delta_N - \frac{9}{4}\delta_{N-1} - \frac{1}{8}\delta_{N-2} \end{aligned}\right\} \qquad (8.3.17)$$

Eliminating δ_{-2} and δ_{-1} from the first two equations of the system (8.3.13) using equations (8.3.16) we obtain:

$$-49.5\delta_0^{n+1} - 39\delta_1^{n+1} - 1.5\delta_2^{n+1} = 49.5\delta_0^n + 39\delta_1^n + 1.5\delta_2^n - 1.5\beta_1 \qquad (8.3.18)$$

$$s1\ \delta_0^{n+1} + s2\ \delta_1^{n+1} + s3\ \delta_2^{n+1} + s4\ \delta_3^{n+1} = s5\ \delta_0^n + s6\ \delta_1^n + s7\ \delta_2^n + s8\ \delta_3^n + \beta_1^*$$

where

$$s1 = \frac{1}{8}(175 - 47R_1Z_{-1}^p + 49R_2)\ ,\quad s2 = \frac{1}{4}(255 + 9(R_1Z_{-1}^p + R_2))$$

$$s3 = \frac{1}{8}(207 + 81R_1Z_{-1}^p - 15R_2)\ ,\quad s4 = 1 + R_1Z_{-1}^p + R_2$$

$$s5 = \frac{1}{8}(175 + 47R_1Z_{-1}^p + -49R_2)\ ,\quad s6 = \frac{1}{4}(255 - 9R_1Z_{-1}^p - 9R_2)$$

$$s7 = \frac{1}{8}(207 - 81R_1Z_{-1}^p + 15R_2)\ ,\quad s8 = 1 - R_1Z_{-1}^p - R_2$$

$$\beta_1^* = \frac{\beta}{8}1(R_1Z_{-1}^p + R_2) \qquad (8.3.19)$$

Similarly, eliminating δ_{N+1} and δ_{N+2} from the last two equations of (8.3.13) and using equations (8.3.17) we obtain:

$$y1\ \delta_{N-3}^{n+1} + y2\ \delta_{N-2}^{n+1} + y3\ \delta_{N-1}^{n+1} + y4\ \delta_N^{n+1} = y5\ \delta_{N-3}^n + y6\ \delta_{N-2}^n + y7\ \delta_{N-1}^n + y8\ \delta_N^n - \beta_2^*$$

where:

$$y8 = \frac{1}{8}(175 - 47R_1Z_{N-3}^p + 49R_2)\ ,\quad y7 = \frac{1}{4}(255 + 9(R_1Z_{N-3}^p + R_2))$$

$$y6 = \frac{1}{8}(207 + 81R_1Z_{N-3}^p - 15R_2)\ ,\quad y5 = 1 + R_1Z_{N-3}^p + R_2$$

$$y4 = \frac{1}{8}(175 + 47R_1Z_{N-3}^p + -49R_2)\ ,\quad y3 = \frac{1}{4}(255 - 9R_1Z_{N-3}^p - 9R_2)$$

$$y2 = \frac{1}{8}(207 - 81R_1Z_{N-3}^p + 15R_2)\ ,\quad y1 = 1 - R_1Z_{N-3}^p - R_2$$

$$\beta_2^* = \frac{\beta}{8}2(R_1Z_{N-3}^p + R_2) \qquad (8.3.20)$$

$$1.5\delta_{N-2}^{n+1} + 39\delta_{N-1}^{n+1} + 49.5\delta_N^{n+1} = -1.5\delta_{N-2}^n - 39\delta_{N-1}^n - 49.5\delta_N^n + 1.5\beta_2 \qquad (8.3.21)$$

The equations (8.3.18)-(8.3.21) together with the third to the (N-1)th equation of (8.3.13) give N+1 equations in the N+1 unknowns $(\delta_0\ ,\ \delta_1\ ,\ \delta_3\ ,\ \ldots\ ,\ \delta_N)^T$ which can be written in a matrix form as:

$$A(\delta^n)\ \delta^{n+1} = B(\delta^n)\delta^n + r \qquad (8.3.22)$$

where; $A(\delta^n)$, and $B(\delta^n)$ are penta-diagonal $(N+1)x(N+1)$ matrices and r is $N+1$ vector:

$$r = \left[-1.5\beta_1,\ \beta_1^*,\ 0, 0,\ \ldots,\ 0,\ -\beta_2^*, 1.5\beta_2\right] \qquad (8.3.23)$$

Since the matrices $A(\delta^n)$, and $B(\delta^n)$ depend on δ^n the matrix equation (8.3.22) is nonlinear. We handle the problem by solving not equation (8.3.22) directly but by setting up an equivalent system [50,68]. Such a system is:

$$A(\delta^n)\ \hat{\delta}^{n+1} = B(\delta^n)\ \delta^n + r \qquad (8.3.24a)$$

$$A\left[\frac{\hat{\delta}^{n+1} + \delta^n}{2}\right]\delta^{n+1} = B\left[\frac{\hat{\delta}^{n+1} + \delta^n}{2}\right]\delta^n + r \qquad (8.3.24b)$$

where equation (8.3.24a) predicts the first approximation $\hat{\delta}^{n+1}$ then equation (8.3.24b) corrects iteratively the improved approximation.

To solve the nonlinear system (8.3.24) we store the penta-diagonal matrices $A(\delta^n)$ and $B(\delta^n)$ in rectangular form $(N+1)x5$ and then use the penta-diagonal algorithm (see Appendix A2) to obtain the solution. The boundary parameters δ_{-2} , δ_{-1} , δ_{N+1} , and δ_{N+2} can be computed at each time step from equations (8.3.16)-(8.3.17).

To start the iterative procedure (8.3.24) a starting vector δ^0 must be determined from the initial condition on $u(x,t)$. Once the parameters δ^n have been determined at a specified time we can compute the solution at the required knots from the formula:

$$u_N(x_i,\ n\Delta t) = \delta_{i-2}^n + 26\delta_{i-1}^n + 66\delta_i^n + 26\delta_{i+1}^n + \delta_{i+2}^n \qquad (8.3.25)$$

$$i = 0, 1,\ \ldots\ , N$$

8.4. The Initial State:

From the initial condition $u(x,0)$ on the function $u(x,t)$ we determine the initial vector $\underset{\sim}{\delta}^0$ so that the evaluation of the time development of $\underset{\sim}{\delta}^n$ and hence u can be undertaken. For more details see section 7.4 .

8.5 The Stability Analysis:

The investigation of the stability of the algorithm will be based on the von Neuman theory in which the growth factor of a typical Fourier mode defined as:

$$\delta_j^n = \hat{\delta}^n \, e^{ijkh} \quad , \tag{8.5.1}$$

where k is the mode number and h is the element size, is determined from the numerical scheme (8.3.13)-(8.3.14) .

The nonlinear term $u^p u_x$ of the KdV equation is not easy to handle it by Fourier method, therefore we linearise it [26,32,35,44]. To do this assume that the quantity u in the nonlinear term $u^p u_x$ is locally constant. This is equivalent to assuming that in (8.3.14) all the δ_j^n are equal to a local constant d, so that $Z_{i-2} = 120d$. Hence equation (8.3.13) can be written:

$$\alpha_1 \, \delta_{j-2}^{n+1} + \alpha_2 \, \delta_{j-1}^{n+1} + \alpha_3 \, \delta_j^{n+1} + \alpha_4 \, \delta_{j+1}^{n+1} + \alpha_5 \, \delta_{j+2}^{n+1} =$$

$$\alpha_5 \, \delta_{j-2}^{n} + \alpha_4 \, \delta_{j-1}^{n} + \alpha_3 \, \delta_j^{n} + \alpha_2 \, \delta_{j+1}^{n} + \alpha_1 \, \delta_{j+2}^{n} \tag{8.5.2}$$

where:

$$j = 0 \, , \, 1 \, , \, \ldots \, , \, N$$

$$\alpha_1 = 1 - R_1^* - R_2 \, , \quad \alpha_2 = 26 - 10R_1^* + 2R_2 \, , \quad \alpha_3 = 66$$

$$\alpha_4 = 26 + 10R_1^* - 2R_2 \, , \quad \alpha_5 = 1 + R_1^* + R_2$$

$$R_1^* = \frac{5}{2h}\varepsilon\Delta t(120d)^p \quad , \quad R_2 = \frac{30}{h^3}\mu\Delta t \tag{8.5.3}$$

If we insert the Fourier mode (8.5.1) into equation (8.5.2), we obtain:

$$\hat{\delta}^{n+1}\left[\alpha_1 e^{-2ikh} + \alpha_2 e^{-ikh} + \alpha_3 + \alpha_4 e^{ikh} + \alpha_5 e^{2ikh}\right] = \hat{\delta}^n\left[\alpha_5 e^{-2ikh} + \alpha_4 e^{-ikh} + \alpha_3 + \alpha_2 e^{ikh} + \alpha_1 e^{2ikh}\right] \tag{8.5.4}$$

This equation has the simple form:

$$(a + ib)\hat{\delta}^{n+1} = (a - ib)\hat{\delta}^n \tag{8.5.5}$$

where:

$$\left.\begin{aligned} i &= \sqrt{-1} \\ a &= 33 + \cos(2kh) + 26\cos(kh) \\ b &= (R_1^* + R_2)\sin(2kh) + (10R_1^* - 2R_2)\sin(kh) \end{aligned}\right\} \tag{8.5.6}$$

Let $\hat{\delta}^{n+1} = g\,\hat{\delta}^n$, where g is the amplification factor, and substitute in (8.5.5) to get:

$$g = \frac{a - ib}{a + ib} \tag{8.5.7}$$

Taking the modulus of this equation gives:

$$|g| = \sqrt{g\bar{g}} = 1 \ ,$$

Therefore the linearised numerical scheme (8.5.2) is unconditionally stable.

8.6 The Test Problems:

The principal purpose of the work reported in this section is the testing of the collocation quintic spline algorithm based on the method which has been investigated in this chapter. For the

testing we shall compute the numerical solution of the generalised **KdV** equation for $p = 1, 2,$ and 3 using different initial and boundary conditions.

As a first test we investigate how well the numerical scheme determines the motion of a single soliton. It is well known that when $p = 1$ the **KdV** has the single soliton analytic solution:

$$u(x,t) = c\ \mathrm{sech}^2(A_1 x - B_1 t + D_1) \tag{8.6.1}$$

provided:

$$A_1 = \frac{1}{2}(\varepsilon\ c/3\ \mu)^{1/2} \quad \text{and} \quad B_1 = \frac{\varepsilon\ c}{3} A_1 \tag{8.6.2}$$

We shall take the following numerical values in the test:

$$c = 1.3 \qquad A_1 = 0.5(1.3)^{1/2} \qquad D_1 = -15A_1$$

Similarly we find that if $p = 2$ the single soliton analytic solution is:

$$u(x,t) = c\ \mathrm{sech}\ (A_1 x - B_1 t + D_1) \tag{8.6.3}$$

provided:

$$A_1 = (\varepsilon\ c^2/6\ \mu)^{1/2} \quad \text{and} \quad B_1 = \frac{\varepsilon\ c^2}{6} A_1 \tag{8.6.4}$$

We will take:

$$c = 1.3 \qquad A_1 = 1.3/\sqrt{2} \qquad D_1 = -15A_1$$

While if $p = 3$ the analytic solution is:

$$u(x,t) = c\ \mathrm{sech}^{2/3}(A_1 x - B_1 t + D_1)\ , \tag{8.6.5}$$

provided:

$$A_1 = \frac{3}{2}(\ \varepsilon\ c^3/10\ \mu)^{1/2} \quad \text{and} \quad B_1 = \frac{\varepsilon c^3}{10} A_1\ . \tag{8.6.6}$$

This time we take:

$$c = 1.3 \qquad A_1 = 1.5(0.6591)^{1/2} \qquad D_1 = -15A_1$$

the boundary conditions for the cases $p = 1,2,3$ with $\varepsilon = 3.0$ and $\mu = 1.0$ are given by:

$$\left.\begin{aligned} u(0,t) &= u(200,t) = 0 \\ u_x(0,t) &= u_x(200,t) = 0 \end{aligned}\right\} \text{ for all time} \tag{8.6.7}$$

and the initial conditions can be obtained from these analytic solutions at t = 0 for the appropriate generalisation of the KdV equation. The solitary waves move steadily to the right unchanged in form. To examine the accuracy of the numerical method we have used the L_2- and L_∞-error norms to compare the numerical and exact solutions.

The L_2-and L_∞-error norms, defined by (4.5.1), (4.5.2) respectively, have been computed and are given in Table 8.1:

Table 8.1

The growth of the errors for single soliton

with $h = 0.2$, $\Delta t = 0.025$, $0 \le x \le 200$

Time	$L_2 \times 10^3$			$L_\infty \times 10^3$		
	p = 1	p = 2	p = 3	p = 1	p = 2	p = 3
1.0	0.159	0.250	0.396	0.091	0.099	0.191
2.0	0.286	0.352	0.699	0.154	0.168	0.425
3.0	0.352	0.390	1.148	0.206	0.251	0.831
4.0	0.435	0.507	2.034	0.259	0.357	1.425
5.0	0.519	0.750	3.260	0.308	0.514	2.159
6.0	0.623	1.019	4.747	0.368	0.673	3.140
7.0	0.712	1.315	6.542	0.415	0.849	4.211
8.0	0.799	1.664	8.669	0.457	1.069	5.559
9.0	0.862	2.032	11.114	0.494	1.296	6.977
10.0	0.925	2.449	13.880	0.534	1.547	8.637

We find that the method has a small error even when p = 3. The L_∞-error norm is smaller than L_2-error norm.

Our second test will involve the interaction of two solitons with $\varepsilon = 3$, $\mu = 1$ and initial condition given for p = 1 by:

$$u(x,0) = c_1 \operatorname{sech}^2(A_1 x + D_1) + c_2 \operatorname{sech}^2(A_2 x + D_2) \qquad (8.6.8)$$

where:

$$c_1 = 1.3 \qquad A_1 = 0.5(1.3)^{1/2} \qquad D_1 = -15A_1$$

$$c_2 = 0.9 \qquad A_2 = 0.5(0.9)^{1/2} \qquad D_2 = -35A_2 ,$$

for p = 2:

$$\mathbf{u}(x,0) = c_1 \operatorname{sech}\,(A_1 x + D_1) + c_2 \operatorname{sech}\,(A_2 x + D_2) \qquad (8.6.9)$$

where:

$$c_1 = 1.3 \qquad A_1 = 1.3/\sqrt{2} \qquad D_1 = -15A_1$$
$$c_2 = 0.9 \qquad A_2 = 0.9/\sqrt{2} \qquad D_2 = -35A_2$$

and for p = 3:

$$\mathbf{u}(x,0) = c_1 \operatorname{sech}^{2/3}(A_1 x + D_1) + c_2 \operatorname{sech}^{2/3}(A_2 x + D_2) \qquad (8.6.10)$$

where:

$$c_1 = 1.3 \qquad A_1 = 1.5(0.3c_1^3)^{1/2} \qquad D_1 = -15A_1$$
$$c_2 = 0.9 \qquad A_2 = 1.5(0.3c_2^3)^{1/2} \qquad D_2 = -35A_2$$

together with the boundary conditions:

$$\left.\begin{aligned} \mathbf{u}\,(0,t) &= \mathbf{u}(200,t) = 0 \\ \mathbf{u}_x(0,t) &= \mathbf{u}_x(200,t) = 0 \end{aligned}\right\} \text{ for all time} \qquad (8.6.11)$$

These conditions represent two solitons, one with amplitude c_1 placed initially at $x = -D_1/A_1$ and a second with amplitude c_2 placed initially at $x = -D_2/A_2$.

All the waves move to the right with a velocity dependent upon their amplitude. To ensure interaction with increasing time we choose $c_1 > c_2$ and $D_1/A_1 > D_2/A_2$. The results of our computations for p = 1,2,3 are shown in Figures 8.1, 8.2 and 8.3; we see that in each of the 3 cases the solitons emerge from the interaction and resume their former shape, amplitude and velocity.

Figures 8.1, 8.2, and 8.3 show the two solitons with large amplitude on the left. As the time increases, the larger soliton catches up with the smaller until, at time t = 40, the smaller soliton is being absorbed. The overlapping process continues until, by time t = 60, the larger soliton has overtaken the smaller one and is in the process of separating. At time t = 100,

the interaction is complete and the larger soliton has separated completely from the smaller one:

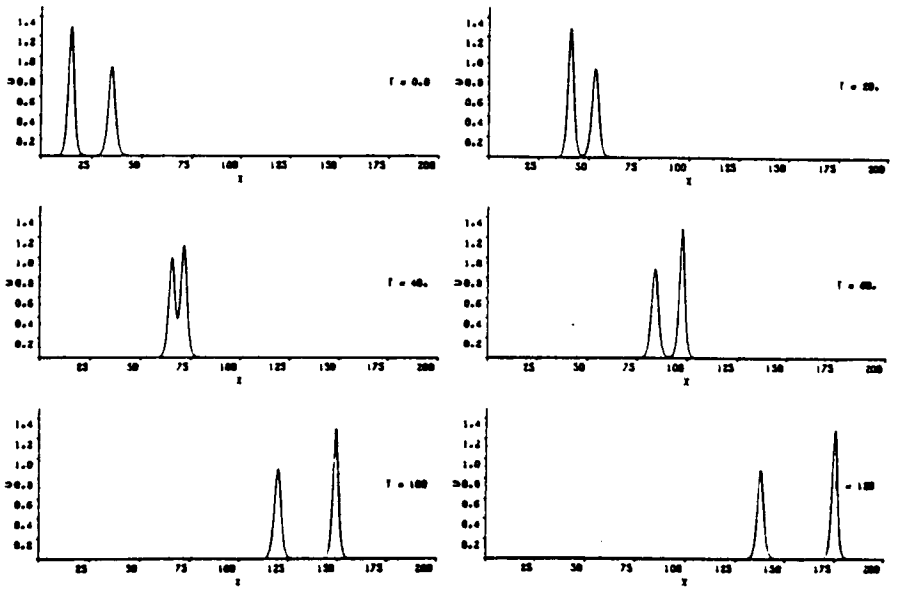

Figure 8.1 p = 1 The interaction of two solitons with $h = 0.2$, $\Delta t = 0.025$, $\varepsilon = 3.0$, $\mu = 1.0$.

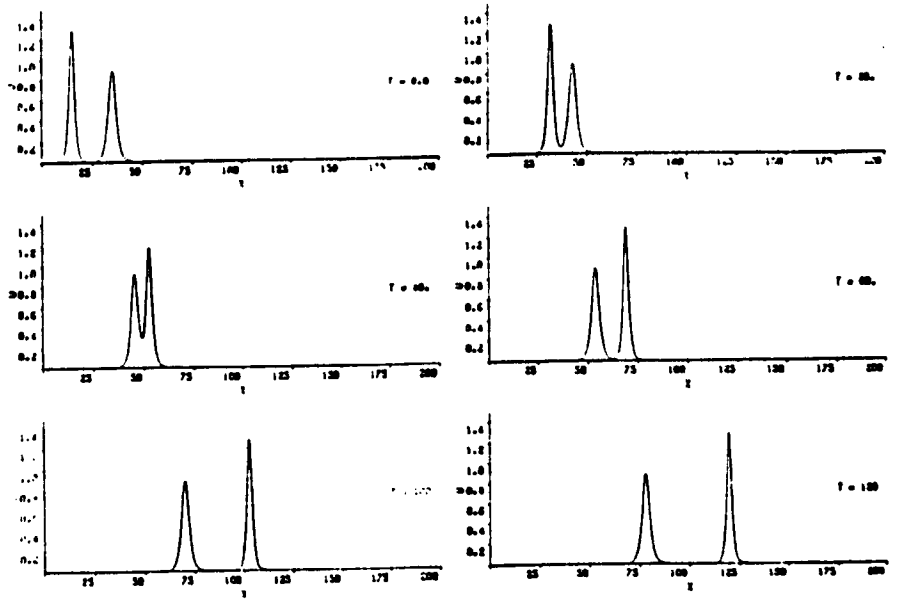

Figure 8.2 p = 2 The interaction of two solitons with $h = 0.2$, $\Delta t = 0.025$, $\varepsilon = 3.0$, $\mu = 1.0$.

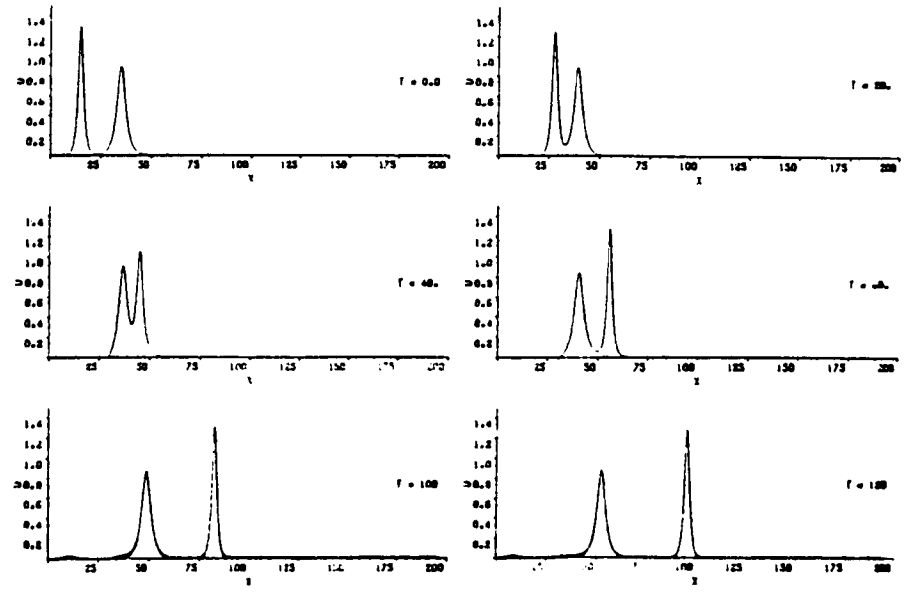

Figure 8.3 $p = 3$ The interaction of two solitons with $h = 0.2$, $\Delta t = 0.025$, $\varepsilon = 3.0$, $\mu = 1.0$.

The generalised KdV equation with $p = 2$ allows solitons of negative amplitude which also move steadily to the right since their velocity depends on their amplitude squared, equation (8.6.4). We have chosen the initial condition of the interaction of two solitons where the large amplitude has positive sign and the small amplitude has negative sign. For the initial condition we use equation (2.3.4.9) when $t = 0$ with $\alpha_1 = 1$, $\alpha_2 = -0.5$, $d_1 = 14\alpha_1$, $d_2 = 2\alpha_2$, $\varepsilon = 6$, $\mu = 1$. The results of the interaction are recorded in Figure 8.4. The waves behave exactly as solitons are expected too:

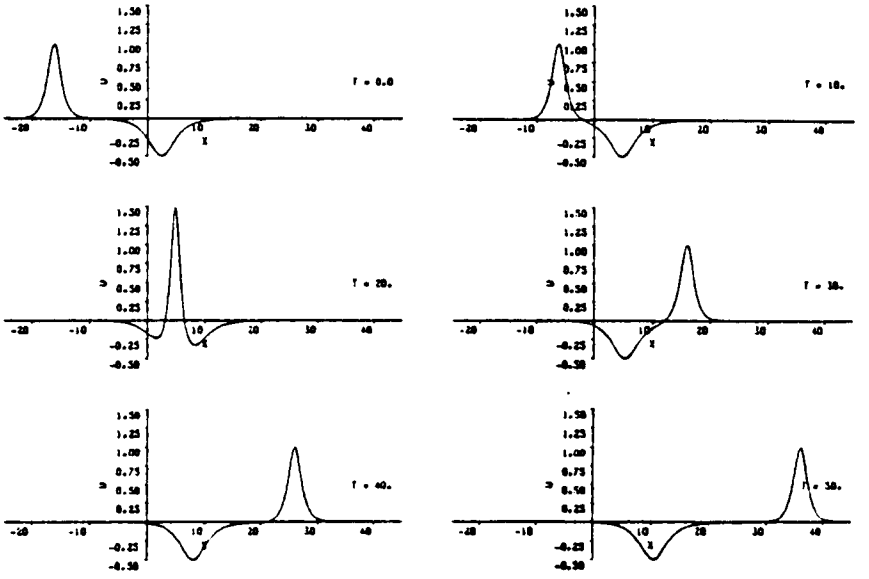

Figure 8.4 $p = 2$ The interaction of a positive and a negative soliton with $h = 0.2$, $\Delta t = 0.005$, $\varepsilon = 6.0$, $\mu = 1.0$.

From the graphs of Figure 8.4 we see that the two solitons are placed with the soliton of larger positive amplitude separated and on the left of the soliton of smaller negative amplitude. As the time increases, the larger soliton catches up with smaller around time t = 20. The merging process continues and around time t = 30, the larger soliton overtakes the smaller one and is in the process of separating. Around time t = 40, the interaction process is complete and the larger soliton has separated completely from the smaller one.

The third problem we shall consider has the initial condition:

$$\text{(i)} \qquad u(x,0) = \frac{1}{2}\left[\, 1 - \tanh\left[\, \frac{x - 25}{5} \,\right] \,\right] \tag{8.6.12}$$

and the boundary conditions we impose are:

$$\left.\begin{aligned} u(-50,t) &= 1 \\ u(\,50,t) &= 0 \\ u_x(-50,t) &= u_x(50,t) = 0 \end{aligned}\right\} \text{ for all } t > 0 \tag{8.6.13}$$

or (ii) $$u(x,0) = \frac{1}{2}\left[\, 1 - \tanh\left[\frac{|x| - 25}{5}\right] \,\right] \qquad (8.6.14)$$

and the boundary conditions are chosen to be:

$$\left.\begin{aligned} u(\mp 150, t) &= 0 \\ u_x(\mp 150, t) &= 0 \end{aligned}\right\} \text{ for all } t > 0 \qquad (8.6.15)$$

Physically this condition (i) can represent among other things, the development of an undular bore in shallow water and a collisionless shock in plasmas. To allow comparison of problem (i) with other authors we have taken $\varepsilon = 0.2$, $\mu = 0.1$ with $\Delta t = 0.05$ and $h = 0.4$. The numerical solution has been determined for the finite range $-50 \le x \le 50$ with the boundary conditions, given above, applied at $x = \mp 50$.

Snapshot solution curves that we have obtained using the present method for the cases $p = 2, 3$ ($p = 1$ has been discussed in section 7.6) are given in Figures 8.5 and 8.6 for times of $t = 0$, 50, 75, 100:

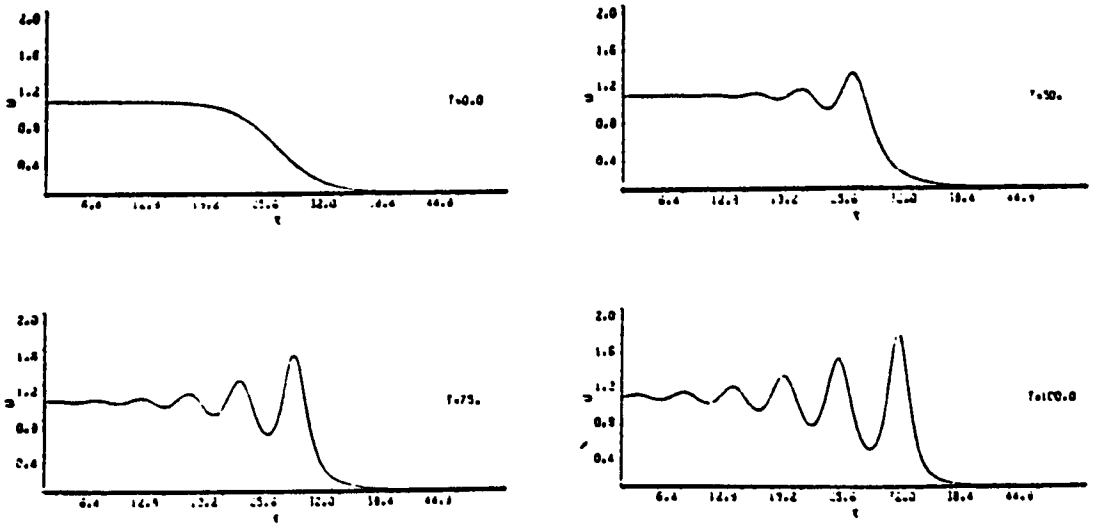

Figure 8.5 Problem (i) $p = 2$ The evolution of the tanh initial condition with $h = 0.4$, $\Delta t = 0.05$, $\varepsilon = 0.2$, $\mu = 0.1$.

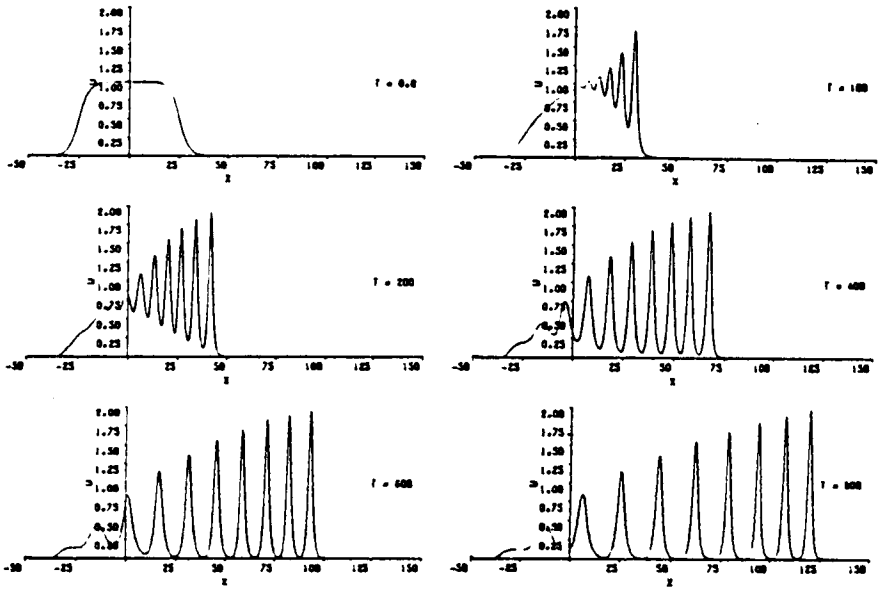

Figure 8.6 Problem (ii) $p = 2$ The evolution of the tanh initial condition with $h = 0.4$, $\Delta t = 0.05$, $\varepsilon = 0.2$, $\mu = 0.1$.

It is observed that the initial perturbation problem (i) has broken up into a regular sequence of waves all of which are moving steadily to the right with constant speeds whose magnitude depends upon their individual amplitude. We find that the conservative quantities vary as time increases. The problem (ii) has been used to satisfy the correct boundary conditions and we have used it to calculate the conservative quantities for $p = 2, 3$ as we did before for $p = 1$ in chapter 7. For $p = 2$ we observe that when the time reaches $t = 800$ the initial perturbation problem (ii) has broken up into a train of 9 solitons.

The final test problem arises from considering the Gaussian distribution function as the initial condition:

$$u(x\ ,\ 0) = \exp(-x^2) \tag{8.6.16}$$

The boundary conditions imposed are:

$$\left.\begin{aligned} u(\mp\ 15, t) &= 0 \\ u\ (\mp\ 15, t) &= 0 \end{aligned}\right\} \text{ for all } t > 0 \tag{8.6.17}$$

We choose $\varepsilon = 1.0$ and we discuss the following cases :

(i) $\mu = 0.04$, $h = 0.1$, $\Delta t = 0.01$

(ii) $\mu = 0.01$, $h = 0.1$, $\Delta t = 0.01$

(iii) $\mu = 0.005$, $h = 0.025$, $\Delta t = 0.005$

(iv) $\mu = 0.0025$, $h = 0.025$, $\Delta t = 0.005$

(v) $\mu = 0.001$, $h = 0.025$, $\Delta t = 0.005$

(vi) $\mu = 0.0005$, $h = 0.025$, $\Delta t = 0.005$

For the case $p = 1$, the problems (i) ,(iii), (v) and (vi) have been discussed in section 7.6. When $\mu = 0.01$, 0.0025 (problem (ii) and (iv)) the initial condition evolves into three and five solitons respectively which are shown in Figures 8.7-8.8:

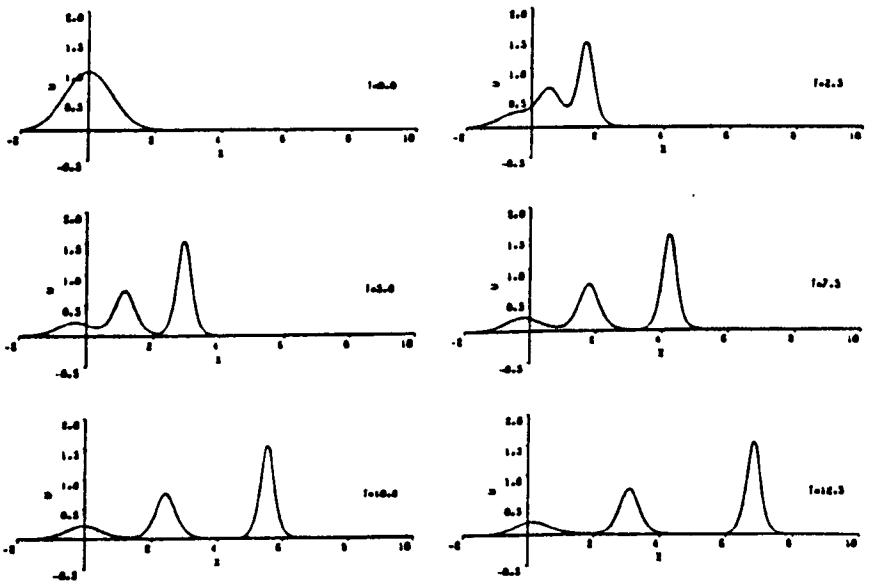

Figure 8.7 $p = 1$ The evolution of $u(x,0) = \exp(-x^2)$
$h = 0.1$, $\Delta t = 0.01$, $\varepsilon = 1.0$, $\mu = 0.01$.

Figure 8.7 shows us that the initial condition (ii) breaks down into three solitons which agrees with those produced by Goda [59]

and we find a very satisfactorily agreement. The agreement between the analytic velocity $c_a \simeq 0.514$, $a \simeq 1.542033$ and the numerical velocity $c_n \simeq 0.52$ for the leading soliton was very satisfactory:

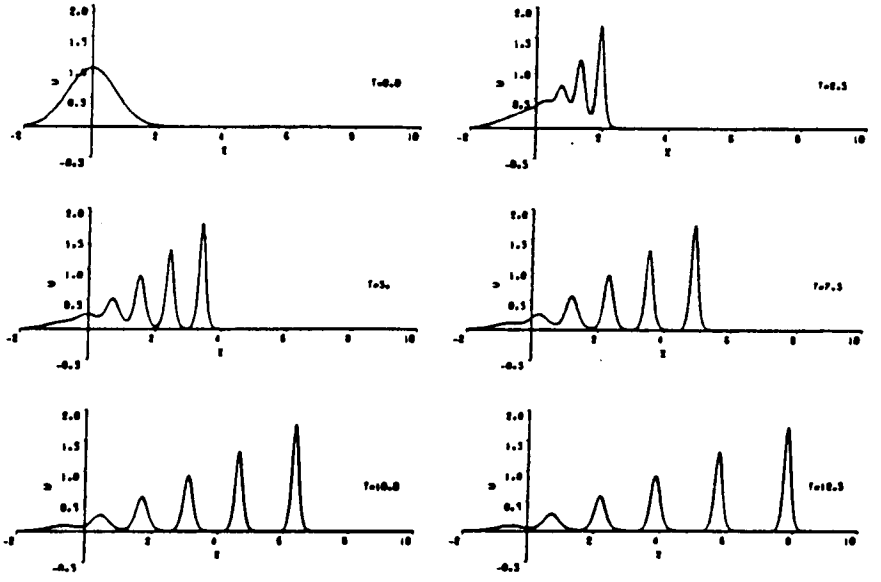

Figure 8.8 $p = 1$ The evolution of $u(x,0) = \exp(-x^2)$ with $h = 0.025$, $\Delta t = 0.005$, $\varepsilon = 1.0$, $\mu = 0.0025$.

From Figure 8.8 we observe that the initial condition (iv) splits into a train of 6 solitons. For the leading soliton the analytic velocity $c_a \simeq 0.589$, $a \simeq 1.767439$ and the observed velocity $c_n \simeq 0.588$ and so agree closely.

For $p = 2$ and 3 similar simulations are reported in Figures (8.9)-(8.15). These show substantially the same behaviour as for the case $p = 1$ simulations.

It has been shown theoretically that the break up of the initial condition (8.6.16) depends on the value of μ [15,39,80]. As μ is increased above $\mu = 0.04$ no solution breaks up into solitons at all, but the solutions for $\mu >> 0.04$ exhibit rapidly oscillating wave packets. When μ is decreased below $\mu = 0.04$ the

initial condition (8.6.16) evolves into more and more solitons. The relative spacing between the solitons increases as the value of μ is decreased:

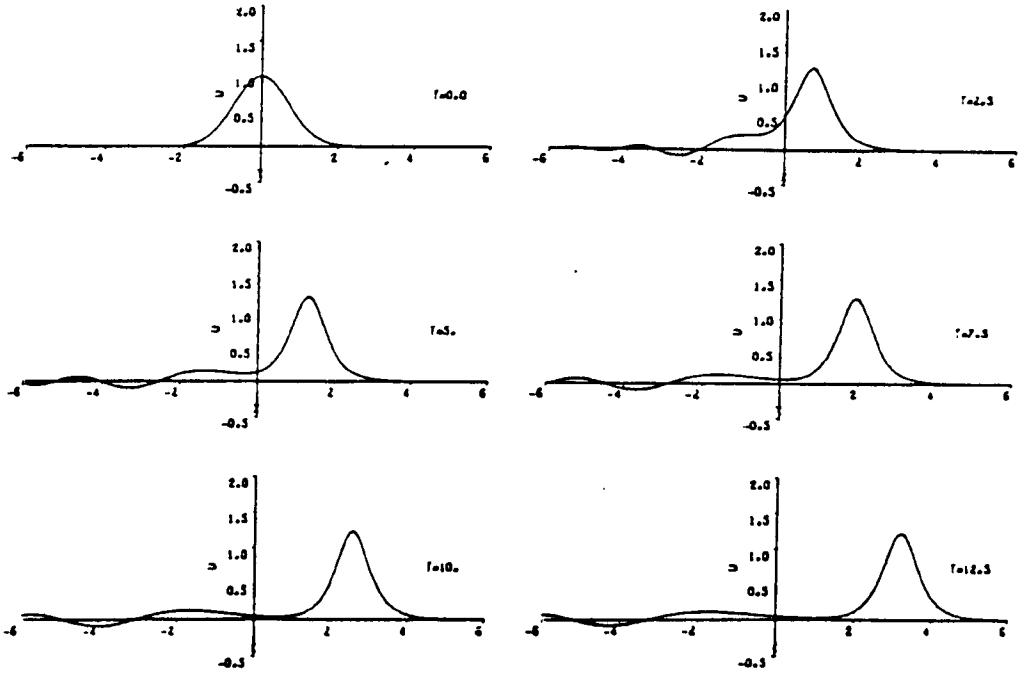

Figure 8.9 $p = 2$ The evolution of $u(x,0) = \exp(-x^2)$
$h = 0.1$, $\Delta t = 0.01$, $\varepsilon = 1.0$, $\mu = 0.04$.

From Figure 8.9 we see that the initial perturbation (i) for $p = 2$ breaks down into a soliton and tail. The analytic velocity for a soliton of mKdV equation is given by:

$$c_a = \varepsilon\, a^2/6 \text{ , a is the amplitude} \qquad (8.6.18)$$

The agreement between the analytic velocity $c_a \simeq 0.268$, derived from the solittons amplitude $a \simeq 1.268807$ and the numerical velocity $c_n \simeq 0.26$ for the leading soliton was very satisfactory. Also we observed that the velocity of the soliton in this case is smaller than that obtained from the case $p = 1$, $\mu = 0.04$.

Figure 8.10 shows that the initial condition (ii) $p = 2$ splits into three solitons. A comparison has been made between the analytic velocity $c_a \simeq 0.447$, derived from the amplitude $a \simeq 1.637368$ and the numerical velocity $c_n \simeq 0.44$: agreement is satisfactory. The soliton has large amplitude and the velocity of

the soliton is smaller than that obtained from case when $p = 1$, $\mu = 0.01$:

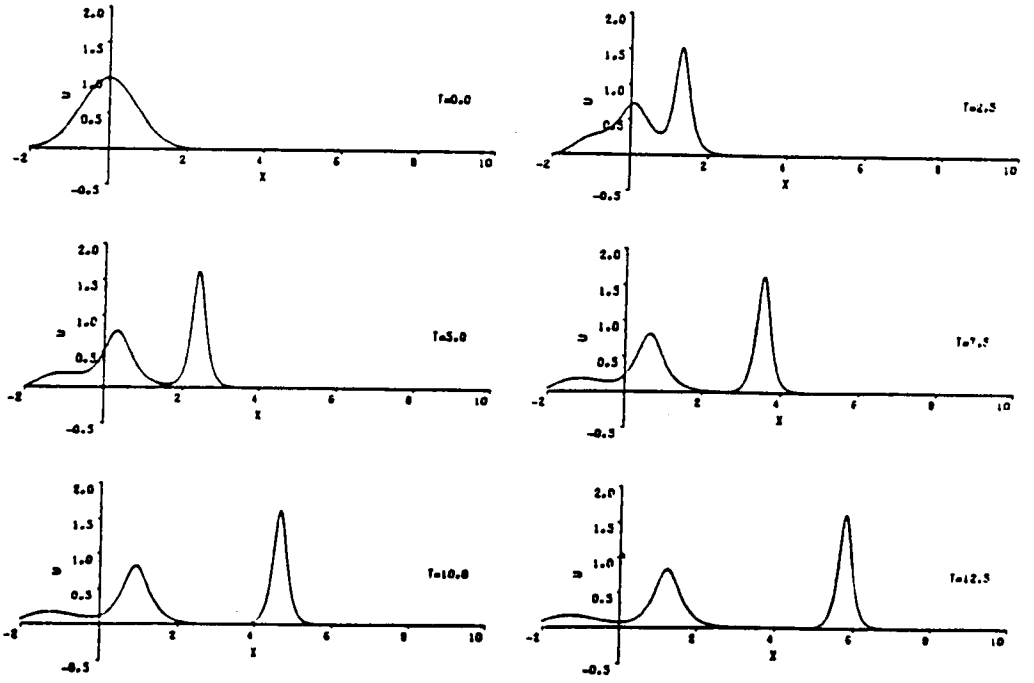

Figure 8.10 $p = 2$ The evolution of $u(x,0) = \exp(-x^2)$
$h = 0.1$, $\Delta t = 0.005$, $\varepsilon = 1.0$, $\mu = 0.01$.

Figure 8.11 shows that the initial perturbation (iii) for $p = 2$ breaks down into a train of 4 solitons. The leading soliton has analytic velocity $c_a \simeq 0.515$, calculated from its amplitude $a \simeq 1.757362$ and numerical velocity $c_n \simeq 0.52$:

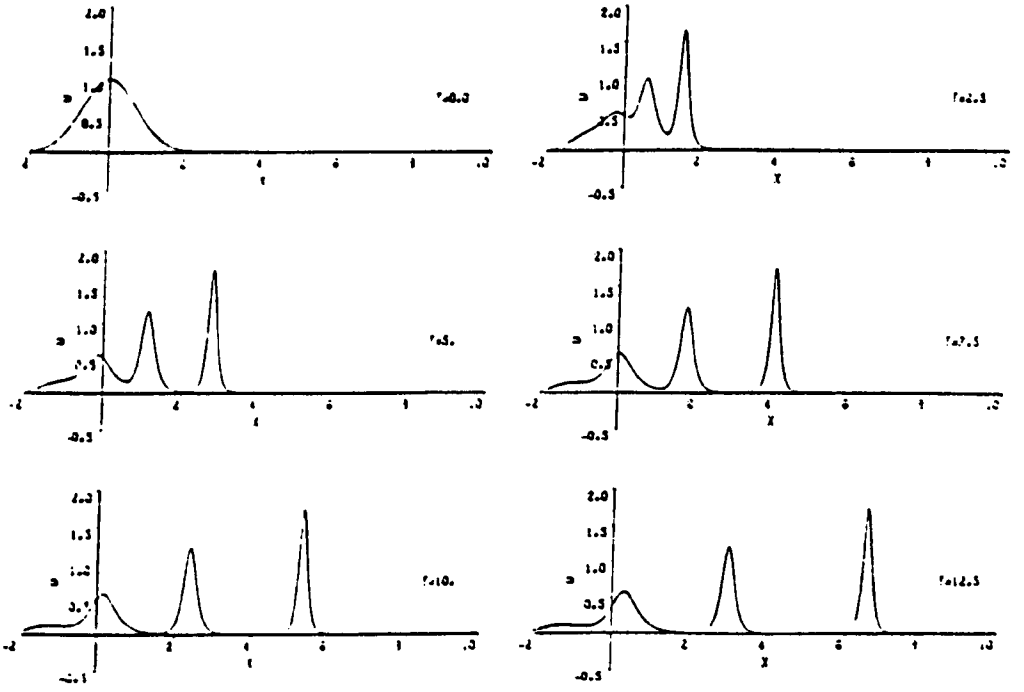

Figure 8.11 $p = 2$ The evolution of $u(x,0) = \exp(-x^2)$
with $h = 0.025$, $\Delta t = 0.005$, $\varepsilon = 1.0$, $\mu = 0.005$.

From Figure 8.12 we see that the initial condition (iv) splits into a train of 5 solitons. The analytic velocity of leading soliton $c_a \simeq 0.533$, calculated from its amplitude $a \simeq 1.788146$ and the observed velocity $c_n \simeq 0.535$ are consistent:

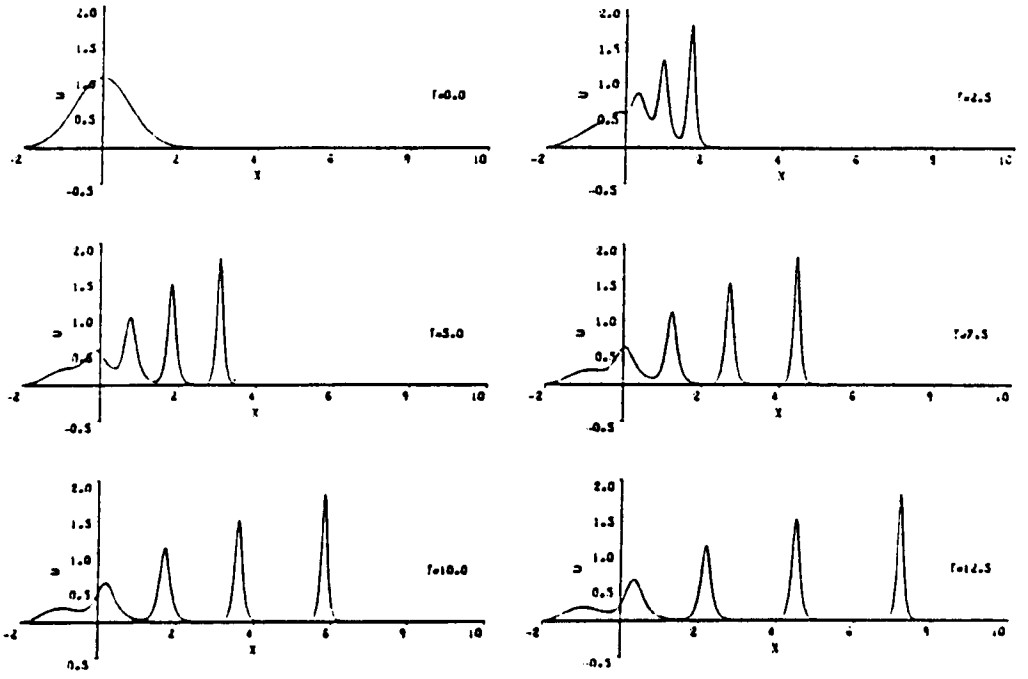

Figure 8.12 $p = 2$ The evolution of $u(x,0) = \exp(-x^2)$ with $h = 0.025$, $\Delta t = 0.005$, $\varepsilon = 1.0$, $\mu = 0.0025$.

The analytic velocity of the soliton produced from the solution of the generalised KdV ($p = 3$) equation is defined by:

$$c_a = \varepsilon a^3/10 \text{ , a is the amplitude} \qquad (8.6.19)$$

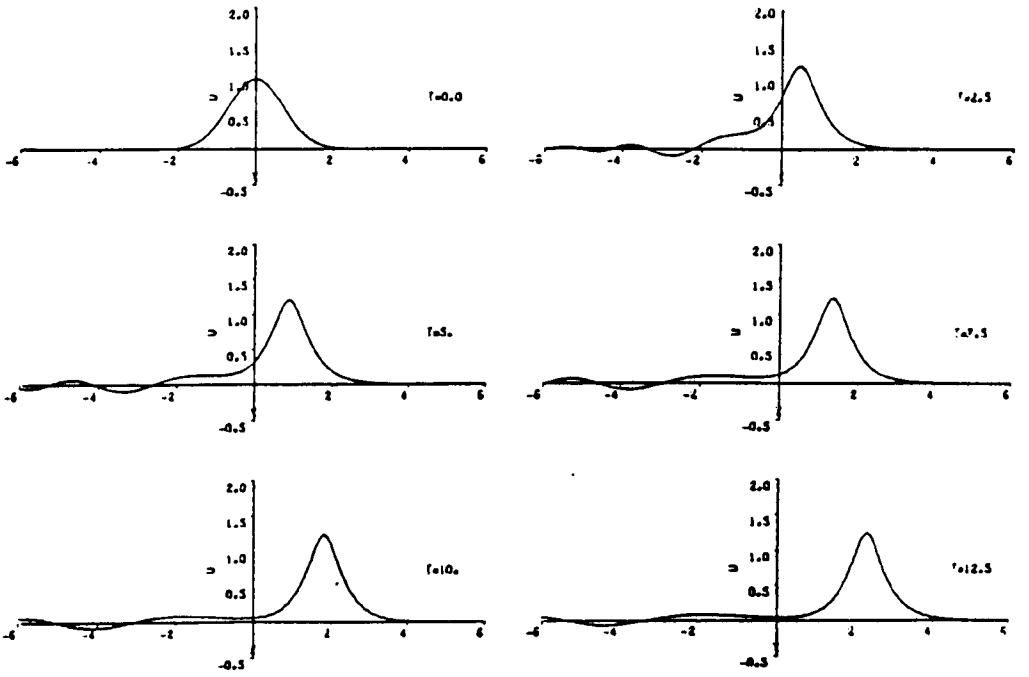

Figure 8.13 $p = 3$ The evolution of $u(x,0) = \exp(-x^2)$ with $h = 0.025$, $\Delta t = 0.005$, $\varepsilon = 1.0$, $\mu = 0.04$.

Figure 8.13 shows us that the initial condition (i) for $p = 3$ splits into a single soliton and a tail. This soliton has analytic velocity $c_a \simeq 0.220$, $(a \simeq 1.30141)$ and numerical velocity $c_n \simeq 0.20$:

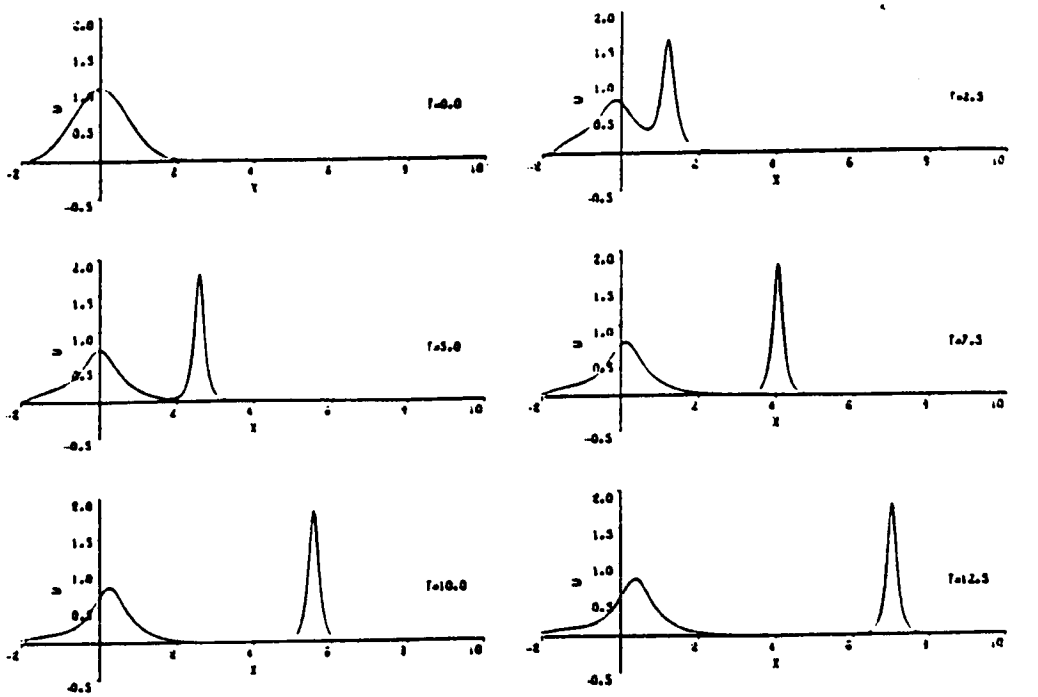

Figure 8.14 $p = 3$ The evolution of $u(x,0) = \exp(-x^2)$ with $h = 0.025$, $\Delta t = 0.005$, $\varepsilon = 1.0$, $\mu = 0.01$.

From Figure 8.14 we see that the initial condition (ii) for $p = 3$ breaks down into a two solitons. The leading soliton has analytic velocity $c_a \simeq 0.590$, $(a \simeq 1.807182)$ and numerical velocity $c_n \simeq 0.59$ which are consistent.

Figure 8.15 shows us that the initial condition (iii) for $p = 3$ splits into 3 solitons. The agreement between the analytic velocity $c_a \simeq 0.622$, $(a \simeq 1.838717)$ and the numerical velocity $c_n \simeq 0.62$ for the leading soliton is very satisfactory.

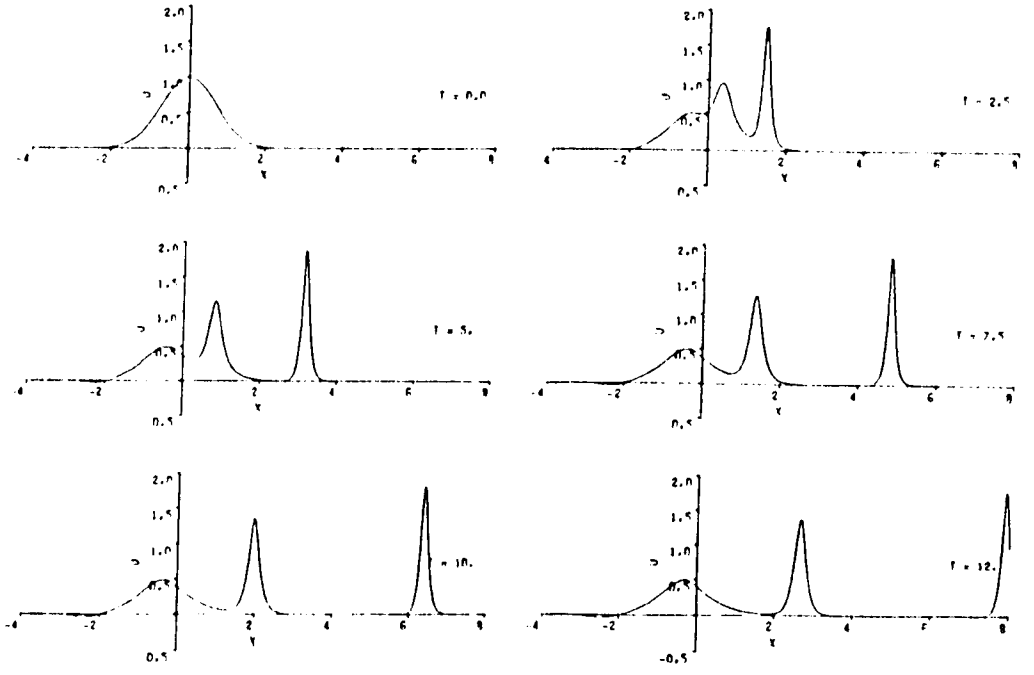

Figure 8.15 $p = 3$ The evolution of $u(x,0) = \exp(-x^2)$ with $h = 0.025$, $\Delta t = 0.005$, $\varepsilon = 1.0$, $\mu = 0.005$.

8.7 Discussion:

Any numerical scheme proposed for computing the solution of the generalised KdV equation should be capable of:

(a) following accurately the motion of a soliton, and

(b) representing accurately the interaction of solitons.

In addition since the KdV equation possesses an infinity of conservation properties, at least the lower order ones should also be exhibited by the recurrence relationship.

The accuracy of our numerical method for the first problem (single soliton) has been tested using the L_2- and L_∞-error norms and we found from Table 8.1 that the present method leads to acceptably low error magnitudes and that the error increases as p increases but remains acceptable.

The KdV equation and the mKdV $(p = 2)$ equation have an infinite number of conservation quantities. We will concentrate on the first four invariant quantities I_i $(i = 1,\ldots,4)$ which were defined in equations (2.4.8)-(2.4.11) respectively for the KdV equation, and (2.4.12)-(2.4.15) respectively for the mKdV equation. The generalised KdV $(p = 3)$ equation has only three conservation quantities I_1, I_2, I_3 which were defined in equations (2.4.16)-(2.4.18) respectively.

The first four conservative quantities I_1, I_2, I_3 and I_4 for the KdV and mKdV equations and the only first three conservative quantities I_1, I_2, I_3 for the generalised KdV $(p = 3)$ equation have been computed and are given in Tables 8.2 and 8.3 for times up to $t = 100$.

We have found from Table 8.2 over the computer runs that the quantities I_1 and I_2 for $p = 1,2,3$ have changed by less than 0.006%, 0.095%, 0.286% and 0.022%, 0.235%, 0.778% respectively which means that they are virtually constants:

Table 8.2

The computed values of the quantities I_1, I_2 for a single soliton with h = 0.2 , Δt = 0.025 $0 \leq x \leq 200$

Tim	I_1			I_2		
	p = 1	p = 2	p = 3	p = 1	p = 2	p = 3
0.0	4.560699	4.442879	4.490562	3.952607	3.676954	3.590333
10.0	4.560732	4.442445	4.488997	3.952693	3.676078	3.587124
20.0	4.560771	4.442041	4.487892	3.952788	3.675214	3.584017
30.0	4.560798	4.441658	4.486920	3.952876	3.674337	3.581008
40.0	4.560815	4.441242	4.485755	3.952958	3.673473	3.578097
50.0	4.560837	4.440826	4.484492	3.953033	3.672602	3.575284
60.0	4.560872	4.440398	4.483160	3.953126	3.671737	3.572557
70.0	4.560895	4.439972	4.481803	3.953209	3.670884	3.569919
80.0	4.560914	4.439530	4.480422	3.953305	3.670033	3.567347
90.0	4.560942	4.439094	4.479072	3.953386	3.669178	3.564854
100.0	4.560971	4.438664	4.477744	3.953467	3.668322	3.562426

Table 8.3

The computed values of the quantities I_3, I_4 for a single soliton with h = 0.2 , Δt = 0.025 $0 \leq x \leq 200$

Time	I_3			I_4	
	p = 1	p = 2	p = 3	p = 1	p = 2
0.0	3.083033	2.071351	1.126853	2.290254	1.050175
10.0	3.083146	2.069869	1.119820	2.290396	1.050754
20.0	3.083268	2.068411	1.113043	2.290540	1.048035
30.0	3.083383	2.066930	1.106503	2.290639	1.051200
40.0	3.083488	2.065473	1.100214	2.290997	1.048351
50.0	3.083587	2.064005	1.094163	2.290924	1.050258
60.0	3.083707	2.062545	1.088328	2.290988	1.054222
70.0	3.083816	2.061107	1.082707	2.291145	1.054919
80.0	3.083939	2.059675	1.077253	2.291444	1.045329
90.0	3.084045	2.058237	1.071990	2.291590	1.044967
100.0	3.084152	2.056797	1.066882	2.291479	1.043393

From Table 8.3 we observe over the computer runs that the quantity I_3 for $p = 1,2,3$ is changed by less than 0.037%, 0.703%, 5.322% respectively and so is relatively constant. While the quantity I_4 for $p = 1,2$ is changed by less than 0.059% and 0.646% respectively and so is satisfactorily constant.

From these tables we conclude that our method leads to very satisfactory results for the values of the conservation quantities for single soliton problem for the **KdV**, **mKdV**, and generalised **KdV** equations.

The L_2- and L_∞-error norms and also the conervative quantities I_i $(i = 1,\ldots,4)$ for the two solitons problem (8.6.8) of the **KdV** equation are listed in Table 8.4. To compute the errors we use the formula for the exact solution of the **KdV** equation for two solitons which is defined by (2.3.4.1):

Table 8.4

The computed values of the errors and I_i $(i=1,\ldots,4)$ of two soliton for equation (8.6.8) with h = 0.2, Δt = 0.025, ε = 3.0, μ = 1.

Time	$L_2 \times 10^3$	$L_\infty \times 10^3$	I_1	I_2	I_3	I_4
0.0			8.355434	6.229448	4.312527	2.922565
20.0	1.659	0.926	8.355734	6.229887	4.312990	2.923004
40.0	0.952	0.926	8.356101	6.230476	4.313579	2.923550
60.0	3.561	1.867	8.356467	6.231020	4.314230	2.924619
80.0	6.612	3.475	8.356751	6.231462	4.314695	2.924619
100.0	11.023	5.566	8.357059	6.231904	4.315172	2.925019
120.0	16.684	8.116	8.357321	6.232340	4.315627	2.925865

We observe over the computer runs that the errors are still acceptable at time t = 120. The quantities I_i $(i = 1,\ldots,4)$ have changed from their original values by less than 0.023%, 0.047%, 0.072% and 0.114% respectively. Hence we consider them as constants. After the interaction of the two solitons their large

and small amplitudes have slightly changed from their original values by less than 0.0011%, 0.005% respectively. So we can say that the amplitudes are virtually unchanged.

To compute the errors for the **mKdV** equation for two solitons we have used the formula for the exact solution which is defined by (2.3.4.9). The first four conservative quantities and the errors for the **mKdV** equation problem (8.6.9) are given in Table 8.5:

Table 8.5

The computed values of the errors and $I_{i\,(i=1,\ldots,4)}$ of two solitons for equation (8.6.9) with h = 0.2, Δt = 0.025, ε = 3.0, μ = 1.

Time	$L_2 \times 10^3$	$L_\infty \times 10^3$	I_1	I_2	I_3	I_4
0.0			8.885759	6.222641	2.758833	1.217337
20.0	8.727	5.281	8.885230	6.221246	2.756189	1.215616
40.0	12.994	7.908	8.885428	6.221223	2.755854	1.217136
60.0	46.377	27.450	8.885139	6.220260	2.753982	1.227959
80.0	86.531	51.160	8.884581	6.218845	2.751315	1.218175
100.0	142.530	83.472	8.883999	6.217449	2.748680	1.209365
120.0	214.213	125.743	8.883410	6.216068	2.746078	1.207956

Table 8.5 shows us that the errors are still acceptable up to time t = 120. The quantities $I_{i\ (i\,=\,1,\ldots,4)}$ have changed from their original values by less than 0.027% 0.106%, 0.463% and 0.771% respectively. Therefore we may consider them as relative constants. After the interaction of the two solitons the large and small amplitudes have changed from their original values by less than 0.18% and 0.02% respectively.

The **mKdV** equation has soliton solutions with both positive and negative amplitudes. If we make the smaller amplitude in the previous problem negative then the error and the first four conservative quantities are given in Table 8.6:

Table 8.6

The computed values of the errors and I_i $(i=1,\ldots,4)$ of two solitons for equation (8.6.9) with amplitudes 1 and -0.5 also $-25 \leq x \leq 45$

$h = 0.2$, $\Delta t = 0.005$, $\varepsilon = 6.0$, $\mu = 1$.

Time	$L_2 \times 10^3$	$L_\infty \times 10^3$	$I_1 \times 10^4$	I_2	I_3	I_4
0.0			-3.1027	3.000000	0.750000	0.206250
10.0	2.607	1.568	3.4736	3.000751	0.750562	0.223425
20.0	28.015	20.538	1.5000	3.001247	0.750439	0.226546
30.0	17.237	20.538	-1.7878	3.001531	0.750785	0.213519
40.0	22.590	20.538	-1.6846	3.002282	0.751352	0.209881
50.0	32.056	20.538	-7.0913	3.003035	0.751916	0.239530

From Table 8.6 we see that the errors are still acceptable up to time $t = 50$. The quantity I_1 has negative sign at the beginning and during the interaction the sign changes to positive and after the interaction changes to the negative. The quantities I_2, I_3, I_4 have changed from their original values by less than 0.101%, 0.256%, 16.136% respectively. After the interaction, of two solitons the positive large and negative small amplitudes have changed from their original values by less than 0.004% and 0.087% respectively. Therefore, we consider them as virtually conserved.

The computed values of the only three conservative quantities for the generalised **KdV** $(p = 3)$ equation (8.6.10) are recorded in Table 8.7:

Table 8.7

The computed quantities I_i $(i=1,\ldots,3)$ of two solitons for the generalised KdV equation with equation (8.6.10) with h =0.2, $\Delta t = 0.025$, $\varepsilon = 3.0$, $\mu = 1.0$

Time	I_1	I_2	I_3
0.0	9.887562	6.579399	1.441773
20.0	9.885179	6.574164	1.429899
40.0	9.886597	6.574096	1.429201
60.0	9.890670	6.573255	1.425755
80.0	9.874264	6.566907	1.412527
100.0	9.799949	6.640472	1.419819
120.0	9.832812	6.594644	1.396324

This Table indicates that over the computer runs the only three conservative quantities have changed from their original values by less than 0.886%, 0.929%, 3.153% respectively. We may consider these quantities as relatively constants. After the interaction of the two solitons the large and small amplitudes have changed from their original values by less than 1.12%, 2.19% respectively.

Using equation (2.3.4.8) the analytic values of the forward and backward phase shifts are given by

For the KdV ($p = 1$) equation:

$$\Delta_1 \simeq 4.19, \qquad \Delta_2 \simeq -5.04$$

For the mKdV ($p = 2$) equation:

$$\Delta_1 \simeq 3.71, \qquad \Delta_2 \simeq -5.36$$

The numerical values of the forward and backward phase shifts are obtained to be

For $p = 1$:

$$\Delta_1 \simeq 4.20, \qquad \Delta_2 \simeq -5.00$$

For $p = 2$:

$$\Delta_1 \simeq 3.60, \qquad \Delta_2 \simeq -5.40$$

The agreement between the analytic and numerical values of Δ_1 and Δ_2 is very satisfactory because of the space step h = 0.2.

The first four conservative quantities for the case (ii) of the KdV (p = 1) equation have been computed in chapter 7 (section 7.7). For the mKdV (p = 2) equation we compute the first four conservative quantities up to a time t = 800. These are given in Table 8.8:

Table 8.8

The computed quantities I_i (i=1,...,4) of the problem (8.6.14) for mKdV equation with h = 0.4, Δt = 0.05, ε = 0.2, μ = 0.1 case (ii)

Time	I_1	I_2	I_3	I_4
0.0	50.00022	45.00045	40.43423	38.00389
100.0	50.00456	45.00834	40.44621	38.08762
200.0	50.00867	45.01721	40.45752	40.69348
300.0	50.01367	45.02948	40.48748	45.55713
400.0	50.07191	45.04237	40.52482	43.25512
500.0	50.02417	45.06118	40.58363	44.64506
600.0	50.02927	45.06668	40.59357	62.91012
700.0	50.03747	45.08215	40.63359	117.2612
800.0	50.03485	45.09077	40.66845	48.27955

We have found over the computer runs that the quantities I_i (i = 1,...,3) have changed from their original values by less than 0.075%, 0.201%, and 0.580% respectively. Therefore we may consider them as relatively constants while the quantity I_4 does vary somewhat. The analytic velocity of the soliton in the mKdV equation is defined by $c_a = a^2\varepsilon/6$ where a is the amplitude. In this case a = 1.9884 , ε = 0.2 . Hence $c_a \simeq 0.1318$ while the numerical velocity is $c_n = 0.132$. Therefore we find that the analytic and numerical velocities are consistent.

The only three conservative quantities for the generalised

KdV ($p = 3$) equation are computed up to time $t = 100$ and shown in Table 8.9:

Table 8.9

The computed quantities I_i $(i=1,\ldots,3)$ of the prblem (ii) for generalised KdV equation with $h = 0.4$, $\Delta t = 0.05$, $\varepsilon = 0.2, \mu = 0.1$

Time	I_1	I_2	I_3
0.0	50.00022	45.00045	38.91780
25.0	50.00221	45.00401	38.92548
50.0	50.00424	45.00762	38.93293
75.0	50.00648	45.01146	38.93885
100.0	50.01132	45.04535	39.08813

We have observed over the computer runs that the quantities I_i $(i=1,\ldots,3)$ for the generalised KdV equation have changed from their original values by less than 0.023%, 0.1%, and 0.44% respectively. So we can consider them as relatively constants.

The conservative quantities of the generalised KdV equation with $p = 1,2,3$ for a problem with initial and boundary conditions given by the equations (8.6.16),(8.6.17) are listed in Tables 8.10 , 8.11:

Table 8.10

The computed values of I_1, I_2 for problem $\exp(-x^2)$ with $\varepsilon = 1.$ and $\mu = 0.01$

	I_1			I_2		
Time	p = 1 h = 0.05 Δt = 0.01	p = 2 h = 0.05 Δt=0.005	p = 3 h = 0.025 Δt=0.0025	p = 1 h = 0.05 Δt = 0.01	p = 2 h = 0.05 Δt=0.005	p = 3 h = 0.025 Δt=0.0025
0.0	1.772454	1.772454	1.772454	1.253314	1.253314	1.253314
2.5	1.772470	1.772490	1.772596	1.253350	1.253362	1.253509
5.0	1.772532	1.772496	1.772569	1.253395	1.253335	1.253233
7.5	1.772460	1.772486	1.772523	1.253440	1.253279	1.252510
10.0	1.772564	1.772550	1.771866	1.253483	1.253224	1.251853
12.5	1.772535	1.772536	1.772505	1.253526	1.253163	1.251124

From Table 8.10 we find that the quantities I_1 , I_2 have changed from their original values by less than 0.007%, 0.006%, 0.034%, 0.017%, 0.013%, 0.175% for p = 1,2,3 respectively. These quantities can be considered as conserved.

Table 8.11

The computed values of I_3 , I_4 for problem $\exp(-x^2)$ with $\varepsilon = 1.0$, $\mu = 0.01$

	I_3			I_4	
Time	p = 1 h = 0.05 Δt = 0.01	p = 2 h = 0.05 Δt=0.005	p = 3 h = 0.025 Δt=0.0025	p = 1 h = 0.05 Δt = 0.01	p = 2 h = 0.05 Δt=0.005
0.0	0.985728	0.811029	0.667334	0.807068	0.597435
2.5	0.985622	0.811017	0.667654	0.808045	0.597985
5.0	0.985606	0.810845	0.665598	0.811226	0.599993
7.5	0.985656	0.810648	0.660797	0.812439	0.613422
10.0	0.985709	0.810450	0.655996	0.812436	0.727596
12.5	0.985762	0.810253	0.651394	0.812494	0.628906

This Table shows us over the computer runs that the quantity I_3 for $p = 1,2,3$ has changed from its original value by less than 0.013%, 0.096%, and 2.389% respectively and we can consider it as constant while I_4 has changed from its original value by less than 0.673% for $p = 1$ but for $p = 2$ the variation was much larger (about 21%).

Finally, we conclude that the collocation method with quintic spline polynomial interpolation functions is useful technique for the computation of solutions to the generalised **KdV** equation over long period of time particularly when space and time steps are small.

CHAPTER 9

CONCLUSIONS

We have set up four finite element solutions to the **KdV** equation. Three of these are based on the Galerkin method but involve different trial functions. These are:

(a) Cubic Hermite polynomials,

(b) Cubic splines, and

(c) Quadratic splines.

A fourth method is based on collocation over finite elements using quintic spline trial functions.

This latter approach has also been used to construct a finite element solution to the generalised **KdV** equation. The cases $p = 2$ and 3 are discussed in detail.

It has been shown analytically that solutions of the **KdV** equation obey an infinity of conservation laws. It is therefore important that any numerical solution shall satisfy, at least, the lower order conservation laws. We choose to evaluate those appropriate to the trial functions being employed.

Solutions to the generalised **KdV** equation $(p > 1)$ obey different conservation laws and we have used the appropriate ones.

We have shown, in earlier chapters, that in all the simulations presented here these conservation laws are all satisfactorily obeyed.

Probably the important solutions of the **KdV** equation are the solitons. Any numerical scheme must be capable of accurately representing the position and amplitude of a soliton as it moves throughout a simulation. The interaction of solitons must also be well described. To evaluate how well our algorithms perform we have used the L_2- and L_∞- error norms. Again we have shown that throughout the simulations these error norms are satisfactorily small.

Taha and Ablowitz [58] have made a comparison of a large number of numerical algorithms and rated them for efficiency. We will now compare the algorithms presented in this thesis with the method rated the best by Taha and Ablowitz [58] call hereafter, the TA scheme:

The first test problem is the single soliton solution:

$$u(x,t) = 3c\ \mathrm{sech}^2(Ax - Bt + D) \tag{9.1}$$

of the **KdV** equation:

$$u_t + \varepsilon\, u\, u_x + \mu\, u_{xxx} = 0 \tag{9.2}$$

where:

$$\varepsilon = 6\ ,\quad \mu = 1\ ,\quad 3c = 2A^2\ ,\quad B = 4A^3\ ,\quad D = 0\ .$$

The initial condition is equation (9.1) with $t = 0$ and the boundary conditions are chosen from:

$$u(\mp 20, t) = u_x(\mp 20, t) = u_{xx}(\mp 20, t) = 0 \tag{9.3}$$

For a soliton with unit amplitude $3c = 1$ and we require the L_∞-error norm to remain below 5×10^{-3} throughout the simulation up until time $t = 1$, and evaluate the CPU time taken. The L_2- and L_∞-error norms and the relative errors in the conserved quantities I_2 and I_3 are recorded. The results are presented in Table 9.1.

It has been found that the speed of the VAX 8650 is 6 mps, and that of the IBM 4341 is 2 mps. Suppose that all the computations have been carried out on the same computer, say, the VAX 8650 then according to Table 9.1 the computing times would be 1.63 secs, 1.76 secs, 0.47 secs, 0.41 secs, and 2.33 secs for Galerkin cubic Hermite, Galerkin cubic spline, Galerkin quadratic spline, collocation quintic spline, and the TA scheme respectively:

Table 9.1

Comparison of the computing time which is required with an accuracy (L_∞) < 0.005 for the numerical methods in solving the KdV

Method	Mesh size	Time	$L_2 x10^3$	$L_\infty x10^3$	υ_1	υ_2	CPU secs
Hermite cubic VAX 8650	h =0.7 Δt = 0.025	0.25	1.22	0.97	$-1.34x10^{-4}$	$-3.09x10^{-3}$	1.63
		0.5	1.33	0.97	$-2.28x10^{-4}$	$-5.47x10^{-3}$	
		1.00	1.72	0.97	$-4.06x10^{-4}$	$-1.01x10^{-2}$	
cubic spline VAX 8650	h =0.5 Δt = 0.025	0.25	1.29	0.95	$-3.04x10^{-5}$	$1.01x10^{-3}$	1.76
		0.50	1.68	0.96	$-4.02x10^{-5}$	$9.95x10^{-4}$	
		1.00	2.16	1.02	$-6.02x10^{-5}$	$9.56x10^{-4}$	
quadratic spline VAX 8650	h =0.5 Δt=.05	0.25	1.50	0.80	$-4.39x10^{-4}$	$-2.30x10^{-2}$	0.47
		0.50	1.70	0.68	$-6.57x10^{-4}$	$-2.34x10^{-2}$	
		1.00	2.30	1.46	$-1.09x10^{-3}$	$-2.41x10^{-2}$	
quintic spline VAX 8650	h =0.4 Δt = 0.025	0.25	0.91	0.62	$-1.97x10^{-5}$	$-2.21x10^{-5}$	0.41
		0.50	0.98	0.62	$-4.00x10^{-5}$	$-5.58x10^{-5}$	
		1.00	1.30	0.68	$-8.07x10^{-5}$	$-1.24x10^{-4}$	
TA scheme IBM 4341	Δx=.16 Δt = 0.125	0.25		1.46	$5.00x10^{-5}$	$4.13x10^{-3}$	7
		0.5		1.62	$7.00x10^{-5}$	$4.19x10^{-3}$	
		1.00		1.73	$1.10x10^{-4}$	$4.26x10^{-3}$	

where:

$\upsilon_1 = (I_2 - I_{20})/I_{20}$, $\upsilon_2 = (I_3 - I_{30})/I_{30}$,

I_{20} = the exact value of $\int u^2\, dx$,

I_{30} = the exact value of $\int (2u^3 - u_x^2)\, dx$,

I_2 , I_3 are the second and the third conservative quantities defined by equations (2.4.9), (2.4.10) respectively.

From Table 9.1 we see that the collocation method is roughly four times faster than the Galerkin cubic Hermite, four times faster than the Galerkin cubic spline, slightly faster than the Galerkin quadratic spline, and six times faster than the TA scheme [58]. We find that all our methods are more accurate and more efficient than the TA scheme and we conclude that the best method

to chose according to efficiency and accuracy is the collocation method with quintic splines as shape functions.

We have made a comparison based on accuracy and efficiency for a single soliton solution with various amplitudes of the **KdV** equation (9.2) between the following numerical methods:

(1) the TA scheme (finite difference) [58],

(2) the BDK methods (a finite element fully discrete Galerkin method) [90] are based on a standard semi-discretisation in the spatial variable x using smooth splines over uniform mesh. For the temporal discretisation various procedures are proposed, mainly second and third order accurate Runge-Kutta methods coupled with Newton's method to handle the nonlinear systems arising from the nolinear term at each time step, and

(3) the finite element methods presented in this thesis.

If it is assumed that all calculations had been executed on the same computer (VAX 8650) and we evaluate the computing time need to attain an accuracy (L_∞) of less than $5x10^{-3}$, $1x10^{-2}$, and $2.2x10^{-2}$ for solitons of amplitudes 1, 2, and 4 respectively throughout a run up to time t = 1, we obtain the results given in Table 9.2 for the time t = 1:

Table 9.2

A comparison between several numerical methods based on the accuracy and efficiency

Method	TA [58]	BDK [90]	Galerki Cubic Hermite	Galerki Cubic Spline	Galerki Quadrat Spline	Collocat Quintic Spline	Ampl
Mesh size	$\Delta x=.16$ $\Delta t=.125$	$\Delta x=1/96$ $\Delta t=.04$	$h = 0.7$ $\Delta t=.025$	$h = 0.5$ $\Delta t=.025$	$h = 0.5$ $\Delta t=.05$	$h = 0.4$ $\Delta t=.025$	1
$L_\infty x10^3$	1.73	1.78	0.97	1.02	1.46	0.68	
CPU secs	2.33	2	1.63	1.76	0.47	0.41	
Mesh size	$\Delta x=.1$ $\Delta t=.1$	$\Delta x=1/144$ $\Delta t=1/45$	$h = 0.4$ $\Delta t=.005$	$h = .35$ $\Delta t=.005$	$h = 0.2$ $\Delta t=.01$	$h = 0.25$ $\Delta t=.005$	2
$L_\infty x10^3$	3.32	2.88	1.75	2.10	2.10	1.07	
CPU secs	7.67	5.67	14.37	12.62	4.63	3.03	
Mesh size	$\Delta x=.05$ $\Delta t=.0275$	$\Delta x=1/172$ $\Delta t=1/140$	$h = 0.3$ $\Delta t = 0.0025$	$h = .25$	$h =.225$ $\Delta t = 0.0025$	$h =.225$	4
$L_\infty x10^3$	17.47	17.10	15.02	10.07	13.19	7.33	
CPU secs	46.67	20.33	38.79	34.83	20.00	6.34	

From Table 9.2 we see that for a single soliton with amplitude one the collocation method is roughly six times faster than the TA scheme, five times faster than the BDK method, four times faster than Galerkin cubic Hermite, four times fatser than Galerkin cubic spline, and slightly faster than Galerkin quadratic spline. We conclude that the methods prsented here are more accurate and efficient than the others and also that the collocation method using quintic splines is the most accurate and efficient method of all.

For a soliton with amplitude two the collocation method is roughly two and half times faster than the TA scheme, two times faster than the BDK method, five times faster than Galerkin cubic Hermite, four times faster than Galerkin cubic spline, and one and

half times faster than Galerkin quadratic spline. Again we reach the same conclusions.

For a soliton with amplitude four the collocation method is roughly seven times faster than the TA scheme, three times faster than the BDK method, six times faster than Galerkin cubic Hermite, five times faster than Galerkin cubic spline, and three times faster than Galerkin quadratic spline. One more the same conclusions are reached.

We have also made a comparison between the following numerical methods:

-Zabusky and Kruskal scheme [27] (finite difference),

-Self-Adaptive conservative scheme SACS [91] (finite difference), and

-The finite element methods presented here for the single soliton solution (9.1) of the **KdV** equation (9.2) with $\varepsilon = 1$, $\mu = 0.000484$, $c = 0.3$, $A = \frac{1}{2}(\frac{\varepsilon\, c}{\mu})^{1/2}$, $B = \varepsilon\, c\, A$, $D = -0.55A$. The initial condition is equation (9.1) with $t = 0$, and the boundary conditions are $\mathbf{u}(0,t) = \mathbf{u}(2,t) = 0$, $\mathbf{u}_x(0,t) = \mathbf{u}_x(2,t) = 0$. The L_∞-error norms have been computed and are listed in Table 9.3:

Table 9.3

Growth of the $L_\infty x10^3$ for several numerical methods for a single soliton

Time	Zabusky Kruskal h = 0.01 Δt=.0005 [27]	SASC h = 0.01 Δt=.0008 [91]	Cubic Hermite h = .033 Δt=.0125	Cubic Spline h = .033 Δt=.0125	Quadratic Spline h = .033 Δt=.0125	Quintic Spline h = .033 Δt=.0125
1.5	13.8	13.2	3.8	6.44	4.7	5.5
2.0	17.9	16.9	5.7	6.8	7.9	5.6
2.5	21.8	20.1	7.1	9.1	6.5	6.0
3.0	26.4	24.2	9.6	9.8	6.4	7.6

Table 9.3 shows that the accuracy of our methods is roughly about

three times better than that of the Zabusky and Kruskal method [27] and of the SASC method [91] even with space and time steps about three and twenty five times larger than those used for the Zabusky and Kruskal [27] and the SASC method [91].

We have also made a comparison of methods for the interaction of two solitons. The initial condition is determined from the analytic solution (2.3.4.1) when t = 0 and

(I) two solitons as initial condition with amplitudes 0.5 and 1 respectively where; $\alpha_1 = 1$, $\alpha_2 = \sqrt{2}$, $d_1 = 0$, $d_2 = 2\alpha_2$, $\varepsilon = 6$, $\mu = 1$ (9.4)

(II) two solitons as initial condition with amplitudes 0.5 and 2.5 respectively where; $\alpha_1 = 1$, $\alpha_2 = \sqrt{5}$, $d_1 = 0$, $d_2 = 10.73$, $\varepsilon = 6$, $\mu = 1$ (9.5)

The boundary conditions are chosen from:

$$u(\mp 20, t) = u_x(\mp 20, t) = u_{xx}(\mp 20, t) = 0 \qquad (9.6)$$

For case (I) we determined the computing time which is required to maintain an accuracy (L_∞) of less than 0.002 throughout the computations. The L_2-error norm, and the relative errors in the second and third conservative quantities are also given in Table 9.4. Assuming that all the computations have been executed on the same computer (VAX 8650) then from Table 9.4 we find that the computing times required to achieve an accuracy (L_∞) of less than 0.002 for the numerical methods Galerkin cubic Hermite, Galerkin cubic spline, Galerkin quadratic spline, collocation quintic spline, and the TA scheme are 4.96 secs, 5.38 secs, 1.51 secs, 1.50 secs, and 6.33 secs respectively:

Table 9.4

Comparison of the computing time which is required with an accuracy (L_∞) < 0.002 for the numerical methods in solving the KdV

Method	Mesh size	Time	$L_2 x10^3$	$L_\infty x10^3$	υ_1	υ_2	CPU secs
Hermite cubic VAX 8650	h =0.7 Δt = 0.025	0.10	1.36	0.97	$-3.64x10^{-4}$	$-1.15x10^{-2}$	4.96
		0.5	1.13	0.64	$-1.60x10^{-4}$	$-3.93x10^{-3}$	
		1.00	1.34	0.64	$-2.84x10^{-4}$	$-6.75x10^{-3}$	
		2.00	1.62	0.88	$-2.60x10^{-4}$	$-3.59x10^{-3}$	
		3.00	1.41	0.88	$-2.62x10^{-4}$	$-3.61x10^{-3}$	
cubic spline VAX 8650	h =0.5 Δt = 0.025	0.10	0.69	0.48	$-1.38x10^{-5}$	$6.73x10^{-4}$	5.38
		0.50	1.07	0.56	$-2.60x10^{-5}$	$5.55x10^{-4}$	
		1.00	1.22	0.52	$-4.08x10^{-5}$	$3.85x10^{-4}$	
		2.00	1.48	0.80	$-6.44x10^{-5}$	$3.01x10^{-5}$	
		3.00	1.36	0.49	$-6.25x10^{-5}$	$7.51x10^{-5}$	
quadratic spline VAX 8650	h =0.5 Δt=.05	0.10	0.88	0.71	$-1.60x10^{-4}$	$-1.71x10^{-2}$	1.51
		0.50	1.18	0.50	$-2.75x10^{-4}$	$-1.57x10^{-2}$	
		1.00	1.43	0.78	$-3.61x10^{-4}$	$-1.34x10^{-2}$	
		2.00	1.90	1.02	$-3.83x10^{-4}$	$-8.13x10^{-3}$	
		3.00	1.89	1.03	$-4.10x10^{-4}$	$-8.02x10^{-3}$	
quintic spline VAX 8650	h =0.4 Δt = 0.025	0.10	0.57	0.43	$-3.46x10^{-6}$	$1.07x10^{-6}$	1.50
		0.50	0.75	0.44	$-1.61x10^{-5}$	$-2.10x10^{-5}$	
		1.00	0.98	0.58	$-2.70x10^{-5}$	$-4.62x10^{-5}$	
		2.00	1.29	0.58	$-3.57x10^{-5}$	$-6.56x10^{-5}$	
		3.00	0.87	0.58	$-3.67x10^{-5}$	$-6.64x10^{-5}$	
TA scheme IBM 4341	Δx=.10 Δt = 0.14	0.10		0.80	$-1.00x10^{-5}$	$1.19x10^{-3}$	19
		0.5		1.13	$-4.00x10^{-5}$	$1.04x10^{-3}$	
		1.00		1.35	$-1.80x10^{-4}$	$6.20x10^{-4}$	
		2.00		1.38	$-4.30x10^{-4}$	$-1.80x10^{-4}$	
		3.00		1.48	$-4.40x10^{-4}$	$-2.30x10^{-4}$	

Again we find that all the methods prsented here are more accurate and efficient than the TA scheme and that the collocation method with quintic splines as shape functions is once more the best of all.

For case (II) we report in Table 9.5 the CPU time (on VAX 8650) required to attain an accuracy of less than 0.02 when the time reaches t = 2.4:

Table 9.5

A comparison between TA scheme and our numerical methods based on the accuracy and efficiency

Method	TA [58]	Galerkin Cubic Hermite	Galerkin Cubic Spline	Galerkin Quadrat Spline	Collocati Quintic Spline	Amplitude
Mesh size	$\Delta x=.075$ $\Delta t=.055$	$h = 0.45$ $\Delta t=.005$	$h = 0.35$ $\Delta t=.005$	$h = 0.3$ $\Delta t=.005$	$h = 0.3$ $\Delta t=.005$	0.5, 2.5
$L_\infty x10^3$	15.02	14.06	8.85	10.02	5.86	
CPU secs	23.67	31.60	31.16	19.30	6.60	

For this problem Galerkin cubic Hermite and Galerkin cubic spline methods are slightly slower than the TA scheme but Galerkin quadratic spline and collocation quintic spline are faster.

The following numerical methods are compared for the mKdV equation:

$$u_t + 6u^2u_x + u_{xxx} = 0 \tag{9.7}$$

(i) the collocation method (finite element method) with quintic splines as shape functions, and

(ii) the TA scheme (finite difference scheme) suggested by Taha and Ablowitz [52].

We compare the computing time required to maintain a certain accuracy throughout the run for various choices of parameters. In this comparison we will use two initial conditions:

(a) The exact solution for a single soliton of the mKdV equation (9.7) is given by:

$$u(x,t) = A \text{ sech } (Ax - Bt + D) \tag{9.8}$$

For the initial condition, put $t = 0$, $A = 1$, $B = A^3 = 1$, and $D = 0$ in equation (9.8). The boundary conditions are imposed:

$$u(\mp 20,t) = u_x(\mp 20,t) = 0 \tag{9.9}$$

The L_2- and L_∞-error norms, the relative errors in the second

and the third conserved quantities, and the computing time are listed in Table 9.6:

Table 9.6

Comparison of the computing time which is required with an accuracy (L_∞) < 0.005 for the numerical methods in solving the mKdV

Method	Mesh size	Time	$L_2 x10^3$	$L_\infty x10^3$	υ_1	υ_2	CPU secs
quintic spline VAX 8650	h =0.5	0.25	6.94	2.86	$1.37x10^{-5}$	$2.16x10^{-4}$	
	Δt =	0.50	5.37	2.93	$5.10x10^{-5}$	$3.72x10^{-4}$	0.36
	0.025	1.00	6.25	3.30	$9.39x10^{-5}$	$4.47x10^{-4}$	
TA scheme IBM 3081	Δx=.10	0.25		1.87	$9.00x10^{-5}$	$4.86x10^{-3}$	
	Δt=.25	0.5		2.79	$1.70x10^{-4}$	$5.08x10^{-3}$	6
		1.00		4.48	$3.30x10^{-4}$	$5.56x10^{-3}$	

Where:

I_{30} = the exact value of $\int (u^4 - u_x^2)\, dx$,

I_2 , I_3 are the second and the third conservative quantities defined by equations (2.4.13), (2.4.14) respectively.

It has been found that the speed of the VAX 8650 is 6 mps, and that of the IBM 3081 is 12 mps. Suppose that all the computations had been run on the same computer (VAX 8650) then from Table 9.6 we see that the computing time required to attain an accuracy (L_∞) of less than 0.005 are 0.36 secs, and 12 secs for numerical methods (i) and (ii) respectively. Then the method (i) is thirty three times faster than the method (ii) (the fastest scheme amongst all the finite difference schemes considered in reference [52]).

(b) The exact solution for the collision of two solitons of the equation (9.7) is given by the equation (2.3.4.9). For the initial condition take t = 0, and

$$\alpha_1 = 0.5,\ \alpha_2 = 2,\ d_1 = 0.625,\ d_2 = 8.75,\ \varepsilon = 6,\ \mu = 1 \quad (9.10)$$

The boundary conditions are chosen from:

$$u(\mp 20, t) = u_x(\mp 20, t) = u_{xx}(\mp 20, t) = 0 \qquad (9.11)$$

The L_2- and L_∞-error norms, the relative errors in the second and the third conserved quantities, and the computing time are recorded in Table 9.7. If all the computations had been made on the same computer (VAX 8650) then from Table 9.7 we find that the computing time required to attain an accuracy (L_∞) of less than 0.02 are 10.50 secs, and 436 secs for numerical methods (i) and (ii) respectively:

Table 9.7

Comparison of the computing time which is required with an accuracy (L_∞) < 0.02 for the numerical methods in solving the mKdV

Method	Mesh size	Time	$L_2 \times 10^3$	$L_\infty \times 10^3$	υ_1	υ_2	CPU secs
quintic spline VAX 8650	h=0.25 Δt = 0.005	0.50	5.61	3.73	-2.48×10^{-4}	-4.71×10^{-4}	10.50
		1.00	4.50	3.73	-2.95×10^{-4}	-4.55×10^{-4}	
		1.50	4.94	3.73	-3.27×10^{-4}	-6.64×10^{-4}	
		2.00	5.09	4.05	-5.42×10^{-4}	-1.62×10^{-4}	
		2.50	10.77	4.05	-9.75×10^{-4}	-3.35×10^{-3}	
		3.00	14.97	9.35	-1.63×10^{-3}	-5.88×10^{-3}	
TA scheme IBM 3081	Δx = 0.0565 Δt = 0.0565	0.50		4.57	-5.30×10^{-4}	-1.27×10^{-3}	218
		1.00		7.08	-9.90×10^{-4}	-4.92×10^{-3}	
		1.50		9.95	-7.30×10^{-4}	-2.28×10^{-3}	
		2.00		13.32	-3.10×10^{-4}	1.89×10^{-3}	
		2.50		15.41	1.60×10^{-4}	5.61×10^{-3}	
		3.00		19.93	5.60×10^{-4}	8.01×10^{-3}	

The collocation method (i) with quintic splines as shape functions is roughly fourty two times faster than The TA scheme (ii). Hence the collocation method with quintic splines as shape functions is the most accurate and efficient method tested for solving the mKdV equation.

We have shown that all four methods described in detail in this thesis are well able to provide efficient and accurate numerical solutions to the **KdV** equation. From above discussion we further conclude that the collocation method with quintic splines as shape functions is the most efficient and accurate numerical method discussed here for solving the **KdV**, **mKdV**, and generalised **KdV** equations. We therefore recommended its use.

Note Added

The single soliton simulations, using the method of collocation with quintic spline shape functions, were repeated with double precision arithematic and no significant effect on the results was obtained. Thus the conclusions already made concerning the efficiency are independent of the computer word length used in the computations.

Appendix A1

Algorithm for the solution of tridiagonal system of equations.

Assume the tridiagonal systems of equations has the general form:

$$- a_i \delta_{i-1} + b_i \delta_i - c_i \delta_{i+1} = d_i \qquad 0 \leq i \leq N$$

with:

$$a_0 = c_N = 0$$

$$\alpha_0 = b_0 \ , \quad \beta_0 = d_0$$

Then compute the following parameters:

$$\alpha_i = b_i - a_i c_{i-1} / \alpha_{i-1}$$

$$\beta_i = d_i + a_i \beta_{i-1} / \alpha_{i-1}$$

$$\text{for } i = 1 , 2 , \ldots , N$$

Then the solution is given by:

$$\delta_N = \beta_N / \alpha_N$$

$$\delta_i = (\beta_i + c_i \delta_{i+1}) / \alpha_i$$

$$\text{for } i = N-1 , N-2 , \ldots , 0$$

Appendix A2

Algorithm for the direct solution of a penta-diagonal system of equations.

Suppose the penta-diagonal systems of equations has the form:

$$a_i \delta_{i-2} + b_i \delta_{i-1} + c_i \delta_i + d_i \delta_{i+1} + e_i \delta_{i+2} = f_i$$

where $0 \le i \le N$ and $a_0 = b_0 = a_1 = e_{N-1} = d_N = e_N = 0$.

Firstly , let:

$$\beta_0 = b_0 \quad , \quad \mu_0 = c_0$$

$$\alpha_0 = d_0/\mu_0 \quad , \quad \lambda_0 = e_0/\mu_0 \quad , \quad \gamma_0 = f_0/\mu_0$$

and

$$\beta_1 = b_1 \quad , \quad \mu_1 = c_1 - \beta_1 \alpha_0$$

$$\alpha_1 = (d_1 - \beta_1 \lambda_0)/\mu_1 \quad , \quad \lambda_1 = e_1/\mu_1 \quad , \quad \gamma_1 = (f_1 - \beta_1 \gamma_0)/\mu_1$$

Then compute the following parameters:

$$\beta_i = b_i - a_i \alpha_{i-2}$$

$$\mu_i = c_i - \beta_i \alpha_{i-1} - a_i \lambda_{i-2}$$

$$\alpha_i = (d_i - \beta_i \lambda_{i-1})/\mu_i$$

$$\lambda_i = e_i/\mu_i$$

$$\gamma_i = (f_i - \beta_i \gamma_{i-1} - a_i \gamma_{i-2})/\mu_i$$

for $i = 2 , 3 , \ldots , N$.

The solution is then given by:

$$\delta_N = \gamma_N$$

$$\delta_{N-1} = \gamma_{N-1} - \alpha_{N-1} \delta_N$$

and

$$\delta_i = \gamma_i - \delta_{i+2} \lambda_i - \delta_{i+1} \alpha_i$$

for $i = N-2 , N-3 , \ldots , 0$.

Appendix A3

Algorithm for the direct solution of a septa-diagonal system of linear equations.

Suppose the septa-diagonal systems of equations has the general form:

$$a_i \delta_{i-3} + b_i \delta_{i-2} + c_i \delta_{i-1} + d_i \delta_i + e_i \delta_{i+1} + f_i \delta_{i+2} + g_i \delta_{i+3} = h_i$$

where $0 \leq i \leq N$ and

$$a_0 = b_0 = c_1 = a_1 = b_1 = a_2 = 0 .$$

$$g_N = f_N = e_N = g_{N-1} = f_{N-1} = g_{N-2} = 0 .$$

Firstly , let:

$$\alpha_0 = b_0 \quad , \quad \beta_0 = c_0 \quad , \quad \mu_0 = d_0$$

$$\zeta_0 = e_0/\mu_0 \quad , \quad \lambda_0 = f_0/\mu_0 \quad , \quad \eta_0 = g_0/\mu_0 \quad , \quad \gamma_0 = h_0/\mu_0$$

$$\alpha_1 = b_1 \quad , \quad \beta_1 = c_1 \quad , \quad \mu_1 = d_1 - \beta_1 \zeta_0$$

$$\zeta_1 = (e_1 - \beta_1 \lambda_0)/\mu_1 \quad , \quad \lambda_1 = (f_1 - \beta_1 \gamma_0)/\mu_1$$

$$\eta_1 = g_1/\mu_1 \quad , \quad \gamma_1 = (h_1 - \beta_1 \gamma_0)/\mu_1$$

and

$$\alpha_2 = b_2 \quad , \quad \beta_2 = c_2 - \alpha_2 \zeta_0 \quad , \quad \mu_2 = d_2 - \lambda_0 \alpha_2 - \beta_2 \zeta_1$$

$$\zeta_2 = (e_2 - \eta_0 \alpha_2 - \lambda_1 \beta_2)/\mu_2$$

$$\lambda_2 = (f_2 - \beta_2 \eta_1)/\mu_2 \quad , \quad \eta_2 = g_2/\mu_2 \quad , \quad \gamma_2 = (h_2 - \alpha_2 \gamma_0 - \beta_2 \gamma_1)/\mu_2$$

Then compute the following parameters:

$$\alpha_i = b_i - a_i \zeta_{i-3}$$

$$\beta_i = c_i - a_i \lambda_{i-3} - \alpha_i \zeta_{i-2}$$

$$\mu_i = d_i - a_i \eta_{i-3} - \lambda_{i-2} \alpha_i - \beta_i \zeta_{i-1}$$

$$\zeta_i = (e_i - \eta_{i-2} \alpha_i - \beta_i \lambda_{i-1})/\mu_i$$

$$\lambda_i = (f_i - \beta_i \eta_{i-1})/\mu_i$$

$$\eta_i = g_i/\mu_i$$

$$\gamma_i = (h_i - \beta_i \gamma_{i-1} - \alpha_i \gamma_{i-2} - a_i \gamma_{i-3})/\mu_i$$

for $i = 3 , 4 , \ldots , N$.

The solution is then given by:

$$\delta_N = \gamma_N$$

$$\delta_{N-1} = \gamma_{N-1} - \zeta_{N-1}\,\delta_N$$

$$\delta_{N-2} = \gamma_{N-2} - \lambda_{N-2}\,\delta_N - \zeta_{N-2}\,\delta_{N-1}$$

and

$$\delta_j = \gamma_j - \zeta_j\,\delta_{j+1} - \delta_{j+2}\,\lambda_j - \delta_{j+3}\,\eta_j$$

for $i = N-3, N-4, \ldots, 0$.

REFERENCES

1. Rusell, J.S., "Report on Waves, Rep. 14th Meeting of the British Assoc. for the Advancement of Science" , John Murray , London , pp.311-390+11 plates , (1844).

2. Bullough, R.K., "Solitons" , Phys. Bull., 29, 78-82, (1978).

3. Stokes, G.G., "On the Theory of Oscillatory Waves" , Camb. Trans. , 8, 441-473 , (1847).

4. Boussinesq, J., "Theorie des Ondes it des Remous Quise Propagent le long d'un Canal Rectangulaire Horizontal, en Communiquant au Liquide Continu Dans ce Canal des Vitesses Sensiblement Parlilles de la Surface au fond" , J.Math.Pures. Appl. , Ser 2, 17, 55-108, (1872).

5. Korteweg, D.J., and de Vries, G., "On the Change of Form of Long Waves Advancing in a Rectangular Canal, and on a New Type of Long Stationary Waves" , Philos. Mag., 39, 422-443, (1895).

6. Zabusky, N.J., "A Synergetic Approach to Problem of Nonlinear Dispersive Wave Propagation and Interaction" , Proc. Symp. Nonlinear partial diff. equations, ed. W Ames Academic Press , 223-258, (1967).

7. Scott, A.C., Chu, F.Y.F.and McLaughlin, D.W., "The Soliton: A New Concept in Applied Science" ,Proc. IEEE , 61, 1443-1483, (1973).

8. Kakutani, T., and Ono, II.,J. Phys. Soc. Japan ,vol.34, 1073-1082, (1973).

9. Gardner, C.S., and Morikawa, G.K., "Similarity in the Asymptotic Behaviour of Collision Free Hydromagnetic Waves and Water Waves" , New York, Courant Inst. Math. Sci. , Res. Rep. NYO-9082, (1960).

10. Zabusky, N.J., "Review Article: Computational Synergetics and Mathematical Innovation" , J. of Comp. Phys. , 43, 195-249, (1981).

11. Su, C.H., and Gardner, C.S., "Korteweg-de Vries Equation and Generalisations III. Derivation of the KdV Equation and Burgers Equation" , J. Math. Phys. , 10, 536-539, (1970).

12. Kruskal, M.D., "A Symptotology in Numerical Computation: Progress and Plans on the Fermi Pasta Ulam Problem" , Proc. IBM Scientific Computing Symposium on Large-Scale Problems in Physics, IBM Data Processing Div., White Plains, N.Y., 43-62, (1965).

13. Zabusky, N.R., "Nonlinear Lattice Dynamics and Energy Sharing" , J. Phys. Soc. Japan , 26, 196-202, (1969).

14. Zabusky, N.J., "Phenomena Associated with the Oscillations of a Nonlinear Model String. The Problem of Fermi, Pasta, and Ulam" , Proc. Conf. on Mathematical Models in the Physical Sciences, S. Drobot, Ed. Prentice-Hall, Englewood Cliffs, N.J., 99-133, (1963).

15. Berezin, Y.H., and Karpman, V.I., "Nonliear Evolution of Disturbances in Plasmas and Other Dispersive Media" , Soviet Phys. JETP , vol. 24, No. 5, 1049-1056, (1967).

16. Washimi, H., and Taniuti, T., "Propagation of Ion Acoustic Solitary Waves of Small Amplitude" , Phys. REv. Lett. , 17, 996-998, (1966)., (1966).

17. Van Wijngaarden, L., "On the Equations of Motion for Mixtures of Liquid and Gas Bubles" , J.Fluid Mech. , 33, 465-474, (1968).

18. Nariboli, G.A., "Nonlinear Longitudinal Dispersive Waves in Elastic Rods" , Iowa State Univ. Engineering Res. Inst. Preprint 442, (1969).

19. Shen, M.C., "Asymptotic Theory of Unsteady Three Dimensional Waves in a Channel of Arbitrary Cross Section" , SIAM J. Appl. Math., 17, 260-271, (1969).

20. Leibovich , S. , "Weakly Nonlinear Waves in Rotating Fluids" " , J. Fluid Mech. , 42 , 803-822 , 1970.

21. Miura,R.M., "The Korteweg-de Vries Equation: A Survey of Results" , SIAM Review , vol.18, No.3, 412-459,z(1976).

22. Schoombie, S.W., "Spline Petrov-Galerkin Methods for the Numerical Solution of the Korteweg-de-Vries Equation" , IMA. J. Num. Anal. , 2, 95-109, (1982).

23. Lax, P.D., "Integrals of Nonlinear Equations of Evolution and Solitary Waves" , Comm. Pure Appl. Math. , vol. 21, 467-490, (1967).

24. Sjöberg, A., "On The Korteweg-de Vries equation: Existence and Uniqueness" , J. Math. Anal. Appl. , vol. 29, 569-579, (1970).

25. Gardner, C.S., Greene, J.M., Kruskal, M.D., and Miura, R.M., "Method for Solving the Korteweg-de Vries Equation" , Phys. Rev. Lett., vol. 19, 1095-1097 , (1967).

26. Greig, I.S., and Morris, J.Ll., "A Hopscotch Method for the Korteweg-de Vries Equations" , J. Comp. Phys. 20, 64-80, (1976).

27. Zabusky, N.J., and Kruskal, M.D., "Interaction of Solitons In a Collisionless Plasma and The Recurrence of Initial States" , Phys. Rev. Lett. , 15, 240-243, (1965).

28. Schamel, H., "Role of Trapped Particles and Waves in Plasma Solitons Theory and Applications" , Physica Scripta, 20, 306-316, (1979).

29. Abe, K., and Inoue, O., "Fourier expansion solution of the Korteweg-de-Vries Equation" , J. Comp. Phys. 31, 202-210, (1980).

30. Gazdag, J., J.Comput.Phys. , 13, 100, (1973).

31. Canosa, J., and Gazdag, j., "The Korteweg-de Vries-Burgers Equation" , J. Comp. Phys. , 23, 393-403, (1977).

32. Fornberg, B., and Whitham, G.B., "A Numerical and Theoretical Study of certain nonlinear wave phenomena" , Phil. Trans. Roy. Soc. , 289, 373-404, (1978).

33. Mitchell, A.R. and Schoombie, S.W., "Finite Element Studies of Solitons" , Numerical Method In Coupled Systems, Edited by R.W. Lewis , P. Bettess , and E. Hintea , John Wiley & Sons Ltd , (1984).

34. Wahlbin, L.B., "A Dissipative Galerkin Method for the Numerical Solution of First Order Hyperbolic Equation. In Mathematical Aspects of Finite Elements in Partial Differential Equations (C.de Boor, Ed.)" , New York, Academic Press , 147-169, (1974).

35. Alexander, M.E., and Morris, J.Ll., "Galerkin Methods for some Model Equations for Nonlinear Dispersive Waves" , J. Comp. Phys. , 30, 428-451, (1979).

36. Sanz-Serna, J.M., and Christie, I., "Petrov Galerkin Methods for Nonlinear Dispersive Waves" , J. Comp. Phys. , 39, 94-102, (1981).

37. Osborne, A.R., and Provenzale, A., "Numerical Methods for Evaluating of the Spectral Transform of Localized Wave Fields Described by the Korteweg-de Vries Equation", to be published in the Proceedings of the Recontre Interdisiplinaire Problems Inverses L'Universite des Sciences et Techniques du Languedoc. Montpellier, France, December (1981).

38. Hearn, A.C., "REDUCE Users Manual Version 3.2", Northwest Computer Algorithms , (1986).

39. Karpman, V.I., "Nonlinear Waves in Dispersive Media", Pergaman Press Ltd , (1975).

40. Dodd, R.K., Eilbeck, J.C., Gibbon, J.D., and Morris, H.C., "Solitons and Nonlinear Waves Equations" , Academic Press , (1984).

41. Nicholson, D.R., "Introduction To Plasma Theory", John Wiley & Sons Inc. , (1983).

42. Lamb, G.L., "Elements of Solitons Theory", John Wiley & Sons , (1980).

43. Pines, D., "The Many Body Problem", New York: Benjamin , (1968).

44. Vliegenthart, A.C., "On Finite Difference methods for the Korteweg-de-Vries Equation", J. Engng. Math., Vol.5, No.2, 137-155, (1971).

45. Miles, J.W., "The Korteweg-de Vries Equation: A Historical essay", J. Fluid Mech., vol. 106, 131-147, (1981).

46. Miura, R.M., "Korteweg-de Vries Equation and Generalisations I. A Remarkable Explicit Nonlinear Transformations", J. of Math. Phys., Vol.9, No.8, 1202-1204, (1968).

47. Drazin, P.G., "The Solitons", Cambridge University Press, (1986).

48. Chen, F., "A Galerkin Method For Strongly Nonlinear **KdV** equations and Schrödinger equations", proc Bail II, (1981).

49. Whitham, G.B., "Linear and Nonlinear Waves", John Wiley & Sons, New York, (1974).

50. Wadati,M., and Toda, M., "The exact N-Soliton solution of the Korteweg-de Vries Equation", J. Phys. Soc. Japan, Vol.32, No.5, 1403-1411, (1972).

51. Gardner, C.S., Greene, J.M. , Kruskal, M.D. and Miura, R.M., "Korteweg-de Vries Equation and Generalisations. VI. Method for the Exact Solution", Commun. Pure, Appl. Maths., 27, 97-133, (1974).

52. Taha, T.R., and Ablowitz, M.J., "Analytic and Numerical Aspects of Certain Nonlinear Evolution Equations, IV.Numeical Modified Korteweg-de Vries Equation", J. Comp. Phys. 77, 540-548, (1988).

53. Eilbeck, J.C., and McGuire, G.R., "Numerical Study of the Regularized Long-Wave Equation, II: Interaction of Solitary Waves", J. Comp. Phys. 23, 63-73, (1977).

54. Miura, R.M., Gardner, C.S., and Kruskal, M.D., "Korteweg-de Vries Equation and Generalisation. II. Existence of Conservation Laws and Constants of Motion", J. of Math. Phys., vol.9, No. 8, 1204-1209, (1968).

55. Zabusky, N.J., "Computational Synergetics", Phys. Tody , 36-46, (1984).

56. Mitchell, A.R., and Griffiths, D. F., "The Finite Difference Method in Partial Differential Equations", John Wiley & Sons, (1980).

57. Smith, G.D., "Numerical Solution of Partial Differential Equation : Finite Difference Methods", Clarendon Press , Oxford, (1978).

58. TAha, T.R., and Ablowitz, M.J., "Analytical and Numerical Aspects of Certain Nonlinear Evolution Equation: III. Numerical Korteweg-de Vries Equation", J. Comp. Phys., 55, 231-253, (1984).

59. Goda, K., "On Stability of some Finite difference scheme for the Korteweg-de-Vries Equation", J. Phys. Soc. Japan 39, 229-236, (1975).

60. Peregrine, D.H., "Calculations of the Development of an Undular Bore", J.Fluid Mech. , vol.25, part 2, 321-330, (1966).

61. Kruskal, M.D., Private communication, (1981).

62. Tappert, F., Lect. Appl. Math. Am. Math. Soc. , 15, 215-216, (1974).

63. Kreiss, H.O., and Oliger,J., Tellus 24 , 199-215, (1972).

64. Fornberg, B., SIAM J. Numer. Anal. 12, 504-528, (1975).

65. Cooley, J.W., Lewis, P.A.W., and Welch, P.D., IEEE Trans. Educ. E-12 No.1, 27-34, (1969).

66. Cooley, J.W., Lewis, P.A.W., and Welch,P.D., J.Sound Vibr.12 , 315-337, (1970).

67. Cooley, J.W., and Tukey, J.W., Math. Comp. 19, 297-301, (1965).

68. Zienkiewicz, O.C., "The Finite Element Method", 3rd ed.,McGraw-Hill, (1982).

69. Zienkiewicz, O.C., and Morgan, K., "Finite Element and Approximation", John Wiley & Sons, (1983).

70. Wait, R., and Mitchell, A.R., "Finite Element Analysis and Applications", John Wiley & Sons, (1985).

71. Davies, A.J., "The Finite Element Method: A First Approach", Clarendon Press Oxford ,(1980).

72. Burden, R.L., and Faires, J.D., "Numerical Analysis", 3rd ed., Prindle,Weber & Schmidt ,(1985).

73. Clough, R.W., "The Finite Element Method in Plane Stress Analysis", J. Struct. Div., ASCE, Proc. 2nd Conf. Electronic Computation, 345-378, (1960).

74. REddy, J.N., "An Introduction to the Finite Element Method", McGraw-Hill, Inc , 1985.

75. Dhatt, G., and Touzot, G., "The Finite Elemente Method Displayed", John Wiely & Sons 1984.

76. Akin, J.E., "Apllication and Implementation of Finite Element Metods", Academic Press, (1982).

77. Schoombie, S.W., "Finite Element Method for the Korteweg-de Vries Equation: I. Galerkin Method with Hermite Cubics", Univ. of Dundee, Dept. of Mathematics, Report NA/43, (1980a)

78. Schoombie, S.W., "Finite Element Methods for the Korteweg-de Vries Equation: II. Petrov-Galerkin Methods with Splines", Report NA/44, Dept. of Mathematics, University of Dundee, (1980b).

79. Gardner, L.R.T., and Ali, A.H.A., "A Numerical Solution of the Korteweg-de Vries Equation Using Galerkin's Method with Hermite Polynomial Shape Functions", Proceedings Intern. AMSE Confer. "Modelling & Simulation", Istanbul (Turkey), June 29-July 1, 1988, AMSE Press, Vol. 1C, p.81-91.

80. Jeffrey, A., and Kakutani, T., "Weak Nonlinear Dispersive Waves: A Discussion Centred Around The Korteweg-de Vries Equation", SIAM Review, vol.14,No.4, 582-643, (1972).

81. Karpman, V.I., "An Asymptotic Solution of the Korteweg-de Vries Equation", Soviet Phys. JETP, vol. 25a, No.10, 708-709, (1967).

82. Prenter, P.M., "Spline and Variational Methods", John Wiley & Sons, (975)

83. Gardner, G., Ali, A.H.A., and Gardner L.R.T., "A Finite Element Solution for the Korteweg-de Vries Equation Using Cubic Spline Shape Functions", Proceedings Intern. AMSE Confer. "Modelling & Simulation", Istanbul (Turkey), June 29-July 1,1988, AMSE Press, Vol. 1C, p.93-104.

84. Khalifa, A.K.A., "Theory and Apllication of the Collocation Method with Splines for Ordinary and Partial Differential Equations", PhD Thesis, Heriot-Watt University, (1979).

85. Khalifa, A.K.A., and Eilbeck, J.C., "Collocation with Quadratic and Cubic Splines", IMA J.Num. Anal.2, 111-121 , (1982).

86. Gardner, L.R.T., and Ali, A.H.A., "A Numerical Solution for the Korteweg-de Vries Equation using Quintic Splines", Proceedings Inter. Confer. "Modelling & Simulation", Istanbul (Turkey), June 29-July 1, 1988, AMSE Press, Vol. 1C, p. 69-80.

87. Behforooz, G.H., and Papamichael, N., "End Conditions for Interpolatory Quintic Splines", IAM J. of Num. Anal., No.1, 81-93, (1981).

88. Maritz, M.F., and Schoombie, S.W., "Parasitic Waves and Solitons In The Numerical Solution of the Korteweg-de Vries and Modified Korteweg-de Vries Equation", J. Comp. Phys. 73, 244-266, (1987).

89. Perelman, T.L., Fridman, A.Kh., and El'Yashevich, M.M., "On The Relationship Between The N-Soliton of The Modified Korteweg-de Vries Equation and KdV Equation Solution", Phys. Lett., Vol. 47a, N. 4, 321-323, (1974).

90. Bona, J., Dougalis, V., and Karakashian, O., "Fully Discrete Methods for the Korteweg-de Vries Equation", Comp.& Maths. with Appls. Vol. 12A, No. 7, 859-884, (1986).

91. Sanz-Serna, J.M., "An Explicit Finite Difference Scheme with Exact Conservation Properties", J. Comput. Phys. 47, 199-210, (1982).