CO23: Soliton

A. Kulanthaivelu
University College

1 Abstract

The aim of this experiment was to numerically determine the time evolution of the soliton solution of the non-linear Korteweg-de Vries equation. We take the localised analytic solution wave-form, i.e a soliton, to the Korteweg-de Vries equation at an initial time and use an explicit finite difference method (the Zabusky-Kruskal algorithm) to plot the time evolution of the wave. We then observe the stability of our numerical solution when the time step is decreased and then increased, finding the solution to be stable for a longer period of time for the former case and go unstable within 0.1 seconds in the latter case. Finally we take the dispersive term in the Korteweg-de Vries equation and set it equal to zero and plot the numerical solution, finding it to steepen and break, but not going unstable before then as it did when the time step was first increased demonstrating that the instability arose from the dispersive term.

2 Introduction

A soliton is a solitary wave, i.e a spatially localised wave with spectacular stability properties. The name, which sounds like a particle was chosen on purpose. A soliton is a wave, but it is also a local maximum in energy density, which preserves its shape and velocity when it moves, exactly as a particle does. It corresponds to the solution of a classical field equation which simultaneously exhibits wave and quasi-particle properties. These are features that one would expect from a quantum system and not from a classical one. The quantum analogy goes so far that soliton tunnelling has been found. Mathematically speaking a soliton is a solitary wave that asymptotically preserves its shape and velocity upon nonlinear interaction with other solitary waves or, more generally, with another arbitrary localised disturbance. As two such solitary waves get closer, they gradually deform and finally merge into a single wave packet; this wave packet, however, soon splits into two solitary waves with the same shape and velocity before "collision" as shown in the figure below.

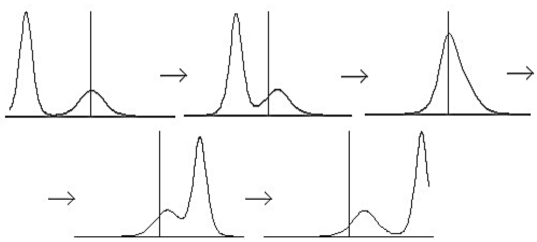


Figure 1: A diagram showing the nonlinear interaction between two solitons

The stability of solitons stems from the delicate balance of "nonlinearity" and dispersion" in the model equations. Nonlinearity drives a solitary wave to concentrate further; dispersion is the effect to spread such a localised wave. If one of these two competing effects is lost, solitons become unstable and eventually cease to exist. In this respect solitons are completely different from "linear waves" like sinusoidal waves.
However these soliton solutions can be used to model various physical situations from the long-time asymptotic behaviour of small, but finite amplitude collisionless-plasma magnetohydrodynamic waves, long waves in anharmonic crystals and shallow water waves. It is the latter of these cases that we will be considering in our analysis.

The basic equation for a disturbance of height $\eta(r,t)$ in a channel with coordinate r at time t is given by

$$\frac{\partial\eta}{\partial t}+c\frac{\partial\eta}{\partial r}+\alpha\eta\frac{\partial\eta}{\partial r}+\delta^2\frac{\partial^3\eta}{\partial r^3}=0 \tag{2.1}$$

where the motion of the disturbance is assumed to be one-dimensional.
If we drop the last two terms we have the wave solution $\eta(r,t)=f(r-ct)$, however the presence of the non-linearity and dispersion terms alters this basic solution.
It's convenient to transform into a frame S moving with a speed c and reexpress the defining differential equation. Therefore we define

$$\begin{gathered} x=r-ct \\ u=\alpha\eta \end{gathered} \tag{2.2}$$

This gives the Korteweg-de Vries (KdV) equation

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+\delta^2\frac{\partial^3 u}{\partial x^3}=0 \tag{2.3}$$

We search for travelling wave solutions to this equation of the form $u(\xi)$ where $\xi=x-ct$.
Substituting this into the KdV equation we get.

$$(u-c)u'+\delta^2u'''=0 \tag{2.4}$$

where the prime refers to differentiation with respect to ξ. We can integrate this solution once with respect to ξ to get.

$$\frac{1}{2}u^2-cu+\delta^2u''+c_1=0 \tag{2.5}$$

where c_1 is an arbitrary integration constant. Multiplying through by u' and integrating with respect to ξ once more we get

$$\frac{1}{6}u^3-\frac{1}{2}cu^2+\frac{1}{2}\delta^2(u')^2+c_1u+c_2=0 \tag{2.6}$$

where c_2 represents another integration constant.
We seek a solitary wave solution so we can add the boundary conditions u, u', $u'' \to 0$ as $\xi\to\infty$ which gives $c_1, c_2=0$. And so we get the differential equation

$$(u')^2=\frac{1}{3\delta^2}u^2(3c-u) \tag{2.7}$$

We find using a substitution that the trial solution $u=3c\,\mathrm{sech}^2\,\theta$ is a solution to this equation provided the following condition is satisfied

$$\theta=\frac{\sqrt{c}}{2\delta}(\xi-\xi_0) \tag{2.8}$$

where ξ_0 is an arbitrary constant.
Therefore we get an exact solution the KdV equation

$$u = A\,\mathrm{sech}^2\left[\frac{(\xi - \xi_0)}{a}\right] \tag{2.9}$$

where $a = \delta\sqrt{\frac{12}{A}}$, $c = \frac{A}{3}$ is the speed of the wave, $\xi = x - ct$ and ξ_0 is an arbitrary constant.
It is a solitary wave or soliton because u is only non-zero in the vicinity of $\xi_0 + Vt$ which is a point moving in the frame S with velocity c. It is characteristic of such solitons that, unlike solutions of the ordinary linear wave equation, the speed and direction of motion depend on amplitude.
The conditions that we imposed whilst solving the equation that $c_1, c_2 = 0$ are not strictly necessary for solving the equation. In the case where we keep them we get periodic cnoidal solutions which involve Jacobi elliptic functions, a generalisation of the ordinary trigonometric function. These solutions reduce to the soliton solution for the correct choice of parameters and so these two solutions are consistent with the each other.
In this experiment we take the analytic soliton solution to the KdV equation as an initial condition and use an explicit finite difference approximation developed by Zabusky and Kruskal to plot its evolution in time. We then investigate the effect on using smaller and larger increments in our approximation on the stability of the wave and the effect of eliminating the dispersion term (leading to the insviscid Burgers' equation $u_t + uu_x = 0$ which characteristically allows "shock-wave" solutions)

3 Solution

We solved the differential equation using a finite difference method, established by Zabusky and Kruskal, to plot the evolution in time of an initial waveform corresponding to the analytic solution of the KdV equation.
We work in step lengths in time and space of Δt and Δx, subdividing the $x - t$ plane into a rectangular mesh as shown in figure 2, so we calculate u at the grid points $x = n\Delta x$ and $t = v\Delta t$ and write

$$u(n\Delta x, v\Delta t) = u_n^v \tag{3.1}$$

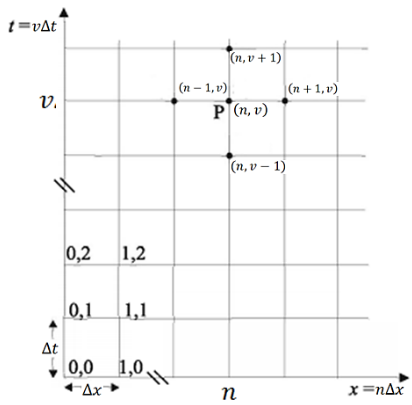


Figure 2: Diagram showing how the x-t plane is subdivided using a numerical mesh

We used the following finite difference approximations

$$\frac{\partial u}{\partial t} \approx \frac{u_n^{v+1} - u_n^{v-1}}{2\Delta t} \tag{3.2}$$

$$\frac{\partial u}{\partial x} \approx \frac{u_{n+1}^{v} - u_{n-1}^{v}}{2\Delta x} \tag{3.3}$$

$$u \approx \frac{u_{n+1}^{v} + u_{n}^{v} + u_{n-1}^{v}}{3} \tag{3.4}$$

$$\frac{\partial^3 u}{\partial x^3} \approx \frac{u_{n+2}^{v} - 2u_{n+1}^{v} + 2u_{n-1}^{v} - u_{n-1}^{v}}{2(\Delta x)^3} \tag{3.5}$$

These approximations are consistent with the Taylor expansion of $u(x, t)$ but they are not unique. It is chosen to make the process as stable as possible over many iterations and are derived in appendix B. These substitutions can be combined to make the recurrence relation:

$$\begin{aligned} u_{n}^{v+1} = u_{n}^{v-1} - \frac{2\Delta t}{2\Delta x}\frac{1}{3}(u_{n+1}^{v} + u_{n}^{v} + u_{n-1}^{v})(u_{n+1}^{v} - u_{n-1}^{v}) - \delta^2 \frac{2\Delta t}{2(\Delta x)^3}(u_{n+2}^{v} \\ - 2u_{n+1}^{v} + 2u_{n-1}^{v} - u_{n-1}^{v}) \end{aligned} \tag{3.6}$$

To avoid problems with boundary conditions we work on a periodic domain, so if we have grid points going from 1 to N $u_{n+N}^{v} = u_{n}^{v}$.
This method computes u using its value at the last by one time step, so for the first step we modify the recurrence relation using the following forward finite difference approximation

$$\frac{\partial u}{\partial t} \approx \frac{u_{n}^{v+1} - u_{n}^{v}}{\Delta t} \tag{3.7}$$

And so for the first step the recurrence relation reads

$$\begin{aligned} u_{n}^{1} = u_{n}^{0} - \frac{\Delta t}{2\Delta x}\frac{1}{3}(u_{n+1}^{0} + u_{n}^{0} + u_{n-1}^{0})(u_{n+1}^{0} - u_{n-1}^{0}) - \delta^2 \frac{\Delta t}{2(\Delta x)^3}(u_{n+2}^{0} \\ - 2u_{n+1}^{0} + 2u_{n-1}^{0} - u_{n-1}^{0}) \end{aligned} \tag{3.8}$$

The momentum

$$\sum_{v=0}^{N-1} u_{n}^{v} \tag{3.9}$$

is conserved since if one sums both sides of the equation with respect to v, then the quantities multiplied by delta x vanish.
The energy

$$\sum_{v=0}^{N-1} \frac{1}{2}(u_{n}^{v})^2 \tag{3.10}$$

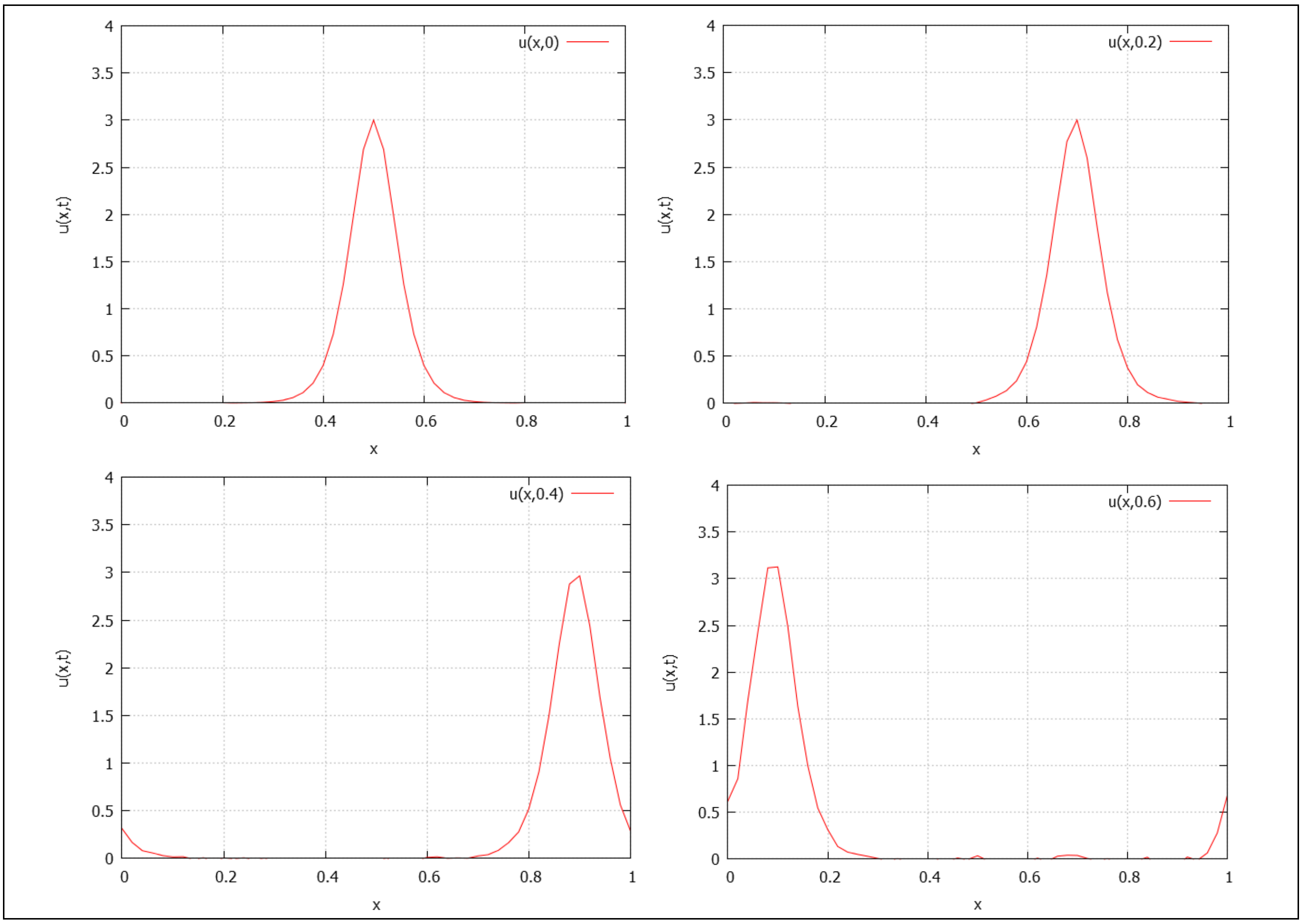
is almost conserved, that is, the above quantity is invariant if we neglect terms

$$(\frac{\Delta x^2}{6}) \sum_{v=0}^{N-1} u_n (u_n)_{ttt} + \mathcal{O}(\delta x^4) \tag{3.11}$$

This is evident if we replace $\frac{u_{n+1}^{v} - u_{n-1}^{v}}{2\Delta x}$ by $\frac{\partial u_n}{\partial t}$, multiply through by u_n, and sum. In practice, those runs which are numerically stable are those which conserve the energy to five significant figures.

4 Results

In our investigation we took the analytic equation (2.9) as the initial condition and plotted its evolution in time using a periodic domain with $N = 50$. We chose increments $\Delta x = 0.02$, $\Delta t = 0.002$, $\delta = 0.03$ and we took the amplitude of the wave to be 3, giving us a theoretical speed of 1.
The numerical solution plots at times $t = 0, t = 0.2, t = 0.4, t = 0.6, t = 0.8$ and $t = 1.0$ have been displayed. From these graphs we can clearly see that the velocity is approximately 1, since it travels a distance of one, i.e one period, in one second. It should be noted that whilst the wave does match the theoretical soliton solution in terms of speed, the numerical solution does become slightly unstable after 0.8 seconds for this value of Δt. The stability can be improved by using a smaller increment.
We demonstrate this by showing the solution for $\Delta t = 0.0002$, a factor of ten smaller. At this level the solution does not go very unstable after one second but it becomes apparent that the speed of the travelling wave is just less than 1. It in fact travels a distance of 0.968 in a time of 1 second assuming of course that the peak of the wave after 1 second corresponds to the peak of the wave at the beginning. This is possibly due to the fact that the method we use does not entirely conserve energy.



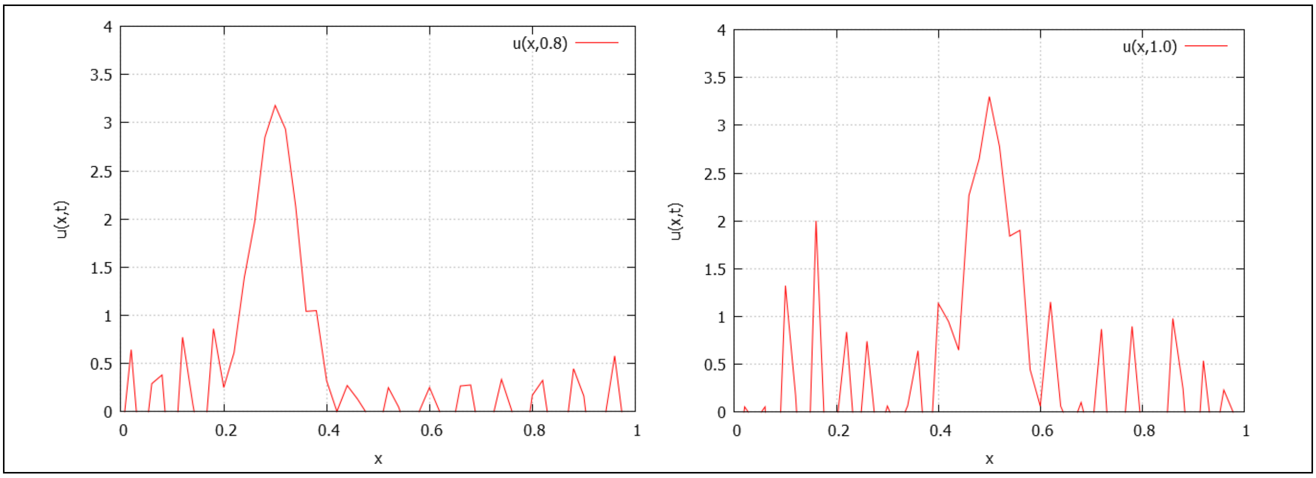


Figure 3: Two dimensional plot showing the time evolution of a soliton from t=0 to 1 with $\delta = 0.03$, $\Delta x = 0.02$ and $\Delta t = 0.002$

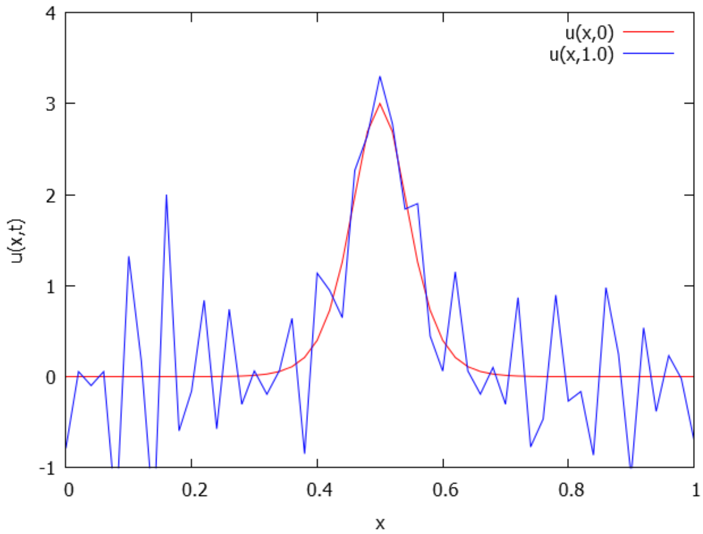


Figure 4: Plot of the soliton at t=0 and t=1 with $\delta = 0.03$, $\Delta x = 0.02$ and $\Delta t = 0.002$

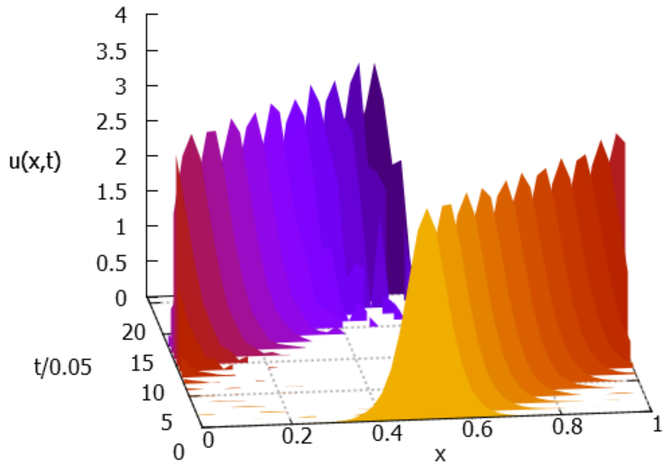


Figure 5: Fence plot of the soliton from t=0 to 1 with $\delta = 0.03$, $\Delta x = 0.02$ and $\Delta t = 0.002$

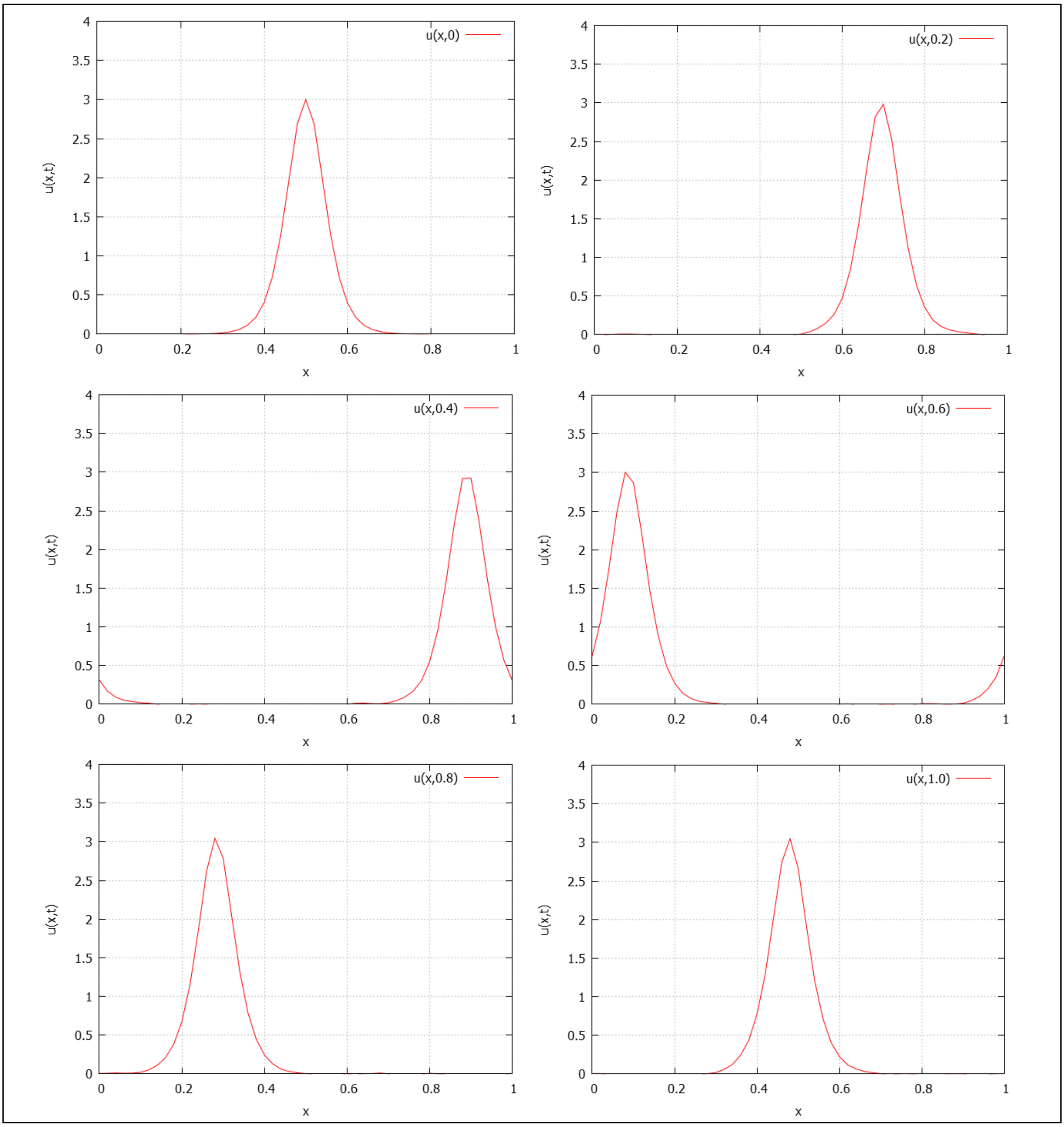


Figure 6: Two dimensional plot showing the time evolution of a soliton from t=0 to 1 with $\delta = 0.03$, $\Delta x = 0.02$ and $\Delta t = 0.0002$

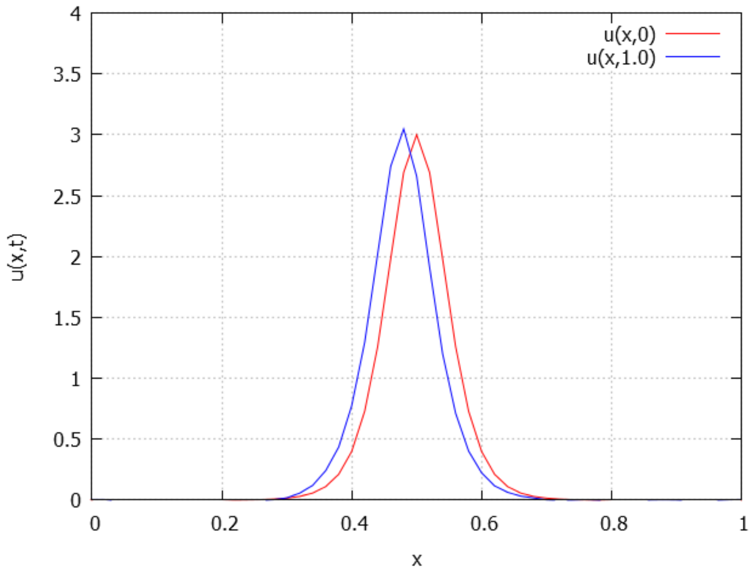


Figure 7: Plot of the soliton at t=0 and t=1 with $\delta = 0.03$, $\Delta x = 0.02$ and $\Delta t = 0.0002$

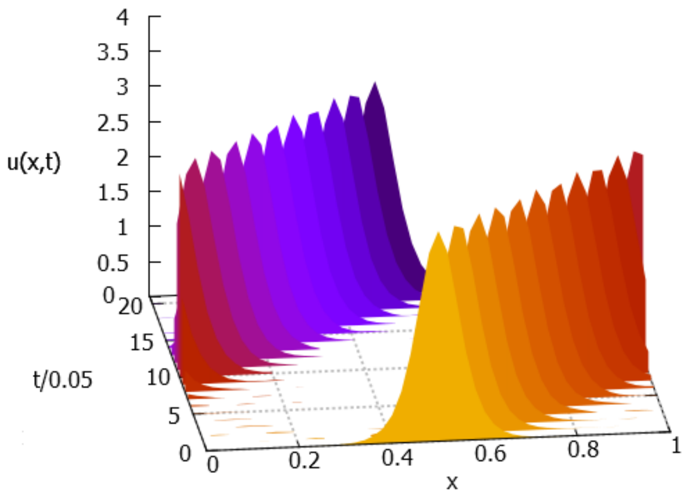


Figure 8: Fence plot of the soliton from t=0 to 1 with $\delta = 0.03$, $\Delta x = 0.02$ and $\Delta t = 0.0002$

We then examine the consequence of increasing the time step in our method and plot graph of the wave over a short period of time. The time evolution of the wave until instability has been plotted both in a two dimensional viewpoint and through a fence plot. The solution is extremely unstable, losing its form in just a few time steps. The stability criterion that Δx and Δt must satisfy can be derived using Von Neumann stability analysis and is done so in the appendix C. Here we fix the spatial separation so only the temporal separation can be varied.

Finally we eliminate the dispersive term by setting $\delta = 0$. We find that whilst the wave does not go unstable as quickly as it does in the previous case, demonstrating that the instability arose from the dispersive term. However it does follow a different time evolution pattern, steepening and then breaking. This is because the KdV equation with $\delta = 0$ becomes the inviscid Burgers' equation whose solutions can develop discontinuities (shock waves) as shown in the plot. Again a two dimensional plots and a fence plot has been provided for clarity.

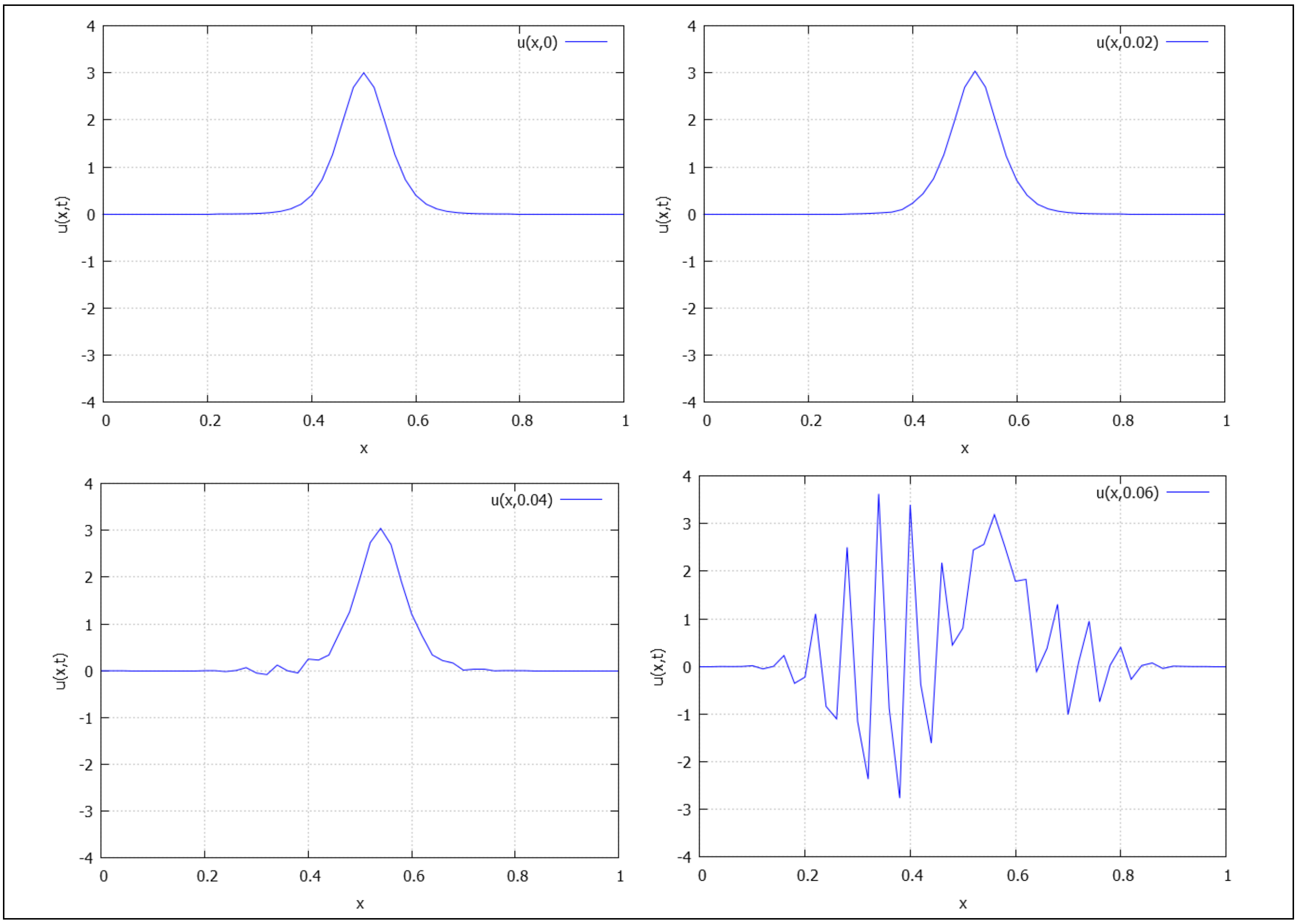


Figure 9: Two dimensional plot showing the time evolution of a soliton from t=0 to 0.06 with $\delta = 0.03$, $\Delta x = 0.02$ and $\Delta t = 0.005$

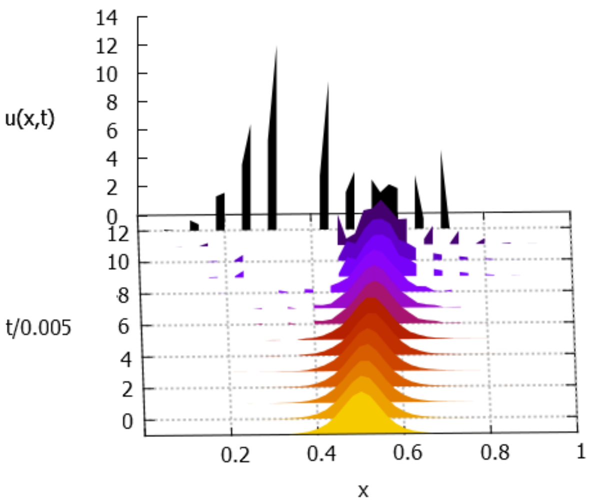


Figure 10: Fence plot of the soliton from t=0 to 1 with $\delta = 0.03$, $\Delta x = 0.02$ and $\Delta t = 0.005$

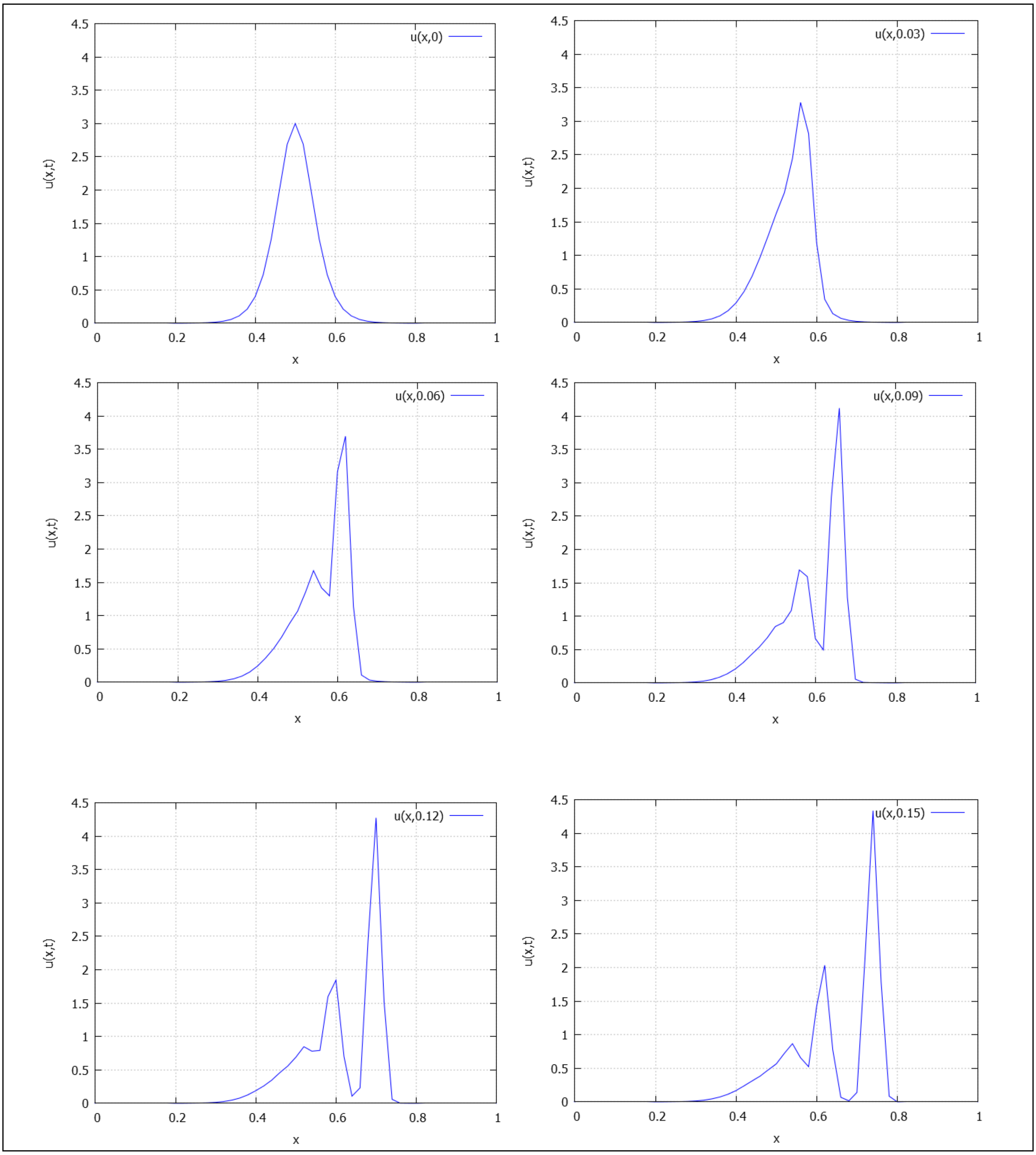


Figure 11: Two dimensional plot showing the time evolution of a soliton from t=0 to 0.06 with $\delta = 0$, $\Delta x = 0.02$ and $\Delta t = 0.005$

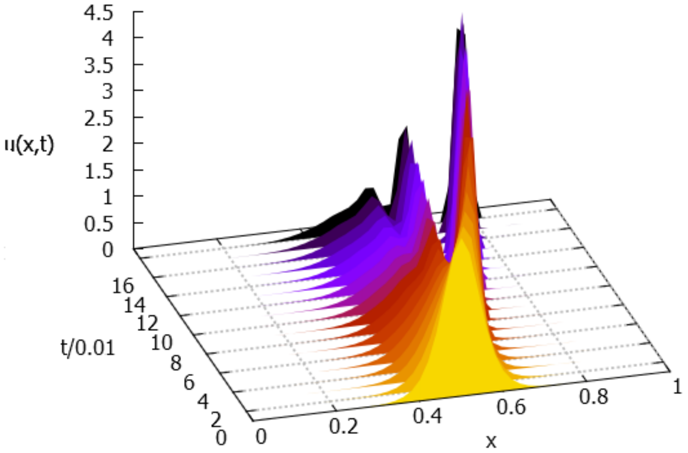


Figure 10: Fence plot of the soliton from t=0 to 0.16 with $\delta = 0$, $\Delta x = 0.02$ and $\Delta t = 0.005$

5 Conclusion

Using our computer model which implemented the Zabusky-Kruskal algorithm we computed the numerical solution to the KdV equation for any arbitrary time when we used the analytic soliton solution to the equation as the initial wave-form, generating solitary wave solutions. For the parameters $\Delta x = 0.02$, $\Delta t = 0.002$, $\delta = 0.03$ we found that the method was successful but became unstable after just 1 second. Using the same parameters but with, $\Delta t = 0.0002$, we found that the solution did not become unstable after 1 second as it did with the previous set of parameters but it also became apparent that the generated wave did not move with a speed that corresponded to the theoretical speed of the wave, instead moving slightly slower. Following this increased the parameter Δt to 0.005 and found that the numerical solution went unstable extremely quickly, within 0.2 seconds, producing a plot that oscillated in an unphysical manner. We then found that setting the dispersion δ equal to zero results in a solution that doesn't go unstable within the same time period, demonstrating that the instability arose from the dispersive term, but instead resulted in a wave that steepened and then broke as time evolved, corresponding to the shockwave solutions of the inviscid Burgers' equation.

The method with which we computed the numerical solution to the KdV equation has several advantages:

- The method takes an almost intractable non-linear partial differential equation and through a process of linearization makes it numerically soluble without a large amount of computation. The KdV equation itself can be considered to be "universal" in the sense that it always arises when the governing equation has weak quadratic nonlinearity and weak dispersion so this method can be quite useful in other physical contexts.
- The method is very accurate since it uses so many centred difference approximations, having a truncation error of $\mathcal{O}((\Delta x)^2 + (\Delta t)^2)$, and it has been shown to converge to a stable solution for a suitable choice of parameters for a large time period.
- The method conserves momentum/mass and almost conserves energy, having an error of $\mathcal{O}(\Delta x^2)$, which is very important in producing physical solutions.
- The model can tolerate a reasonable level of variation in the input parameters such as δ or the increment parameters so long as the still satisfy the stability condition. This allows for studying the solutions to other equations such as the inviscid Burgers' equation or the effects of varying the dispersion on the numerical solutions.
- The method used can, in principle, be modified so that the nonlinear interaction between different solitons can be studied.

However the method still has several disadvantages:

- The method produces unstable solutions if the correct choice of parameters aren't chosen. If you don't know the stability criterion for the method then this could cause some difficulties.
- The method can only be applied to partial differential equations which have the same form as the KdV equation. For other differential equations a different finite difference scheme must be implemented.
- The numerical solutions produced are never totally stable and will go unstable after a sufficiently long time. This could possibly be remedied by using higher order finite differences but the benefit of a more accurate solution would come at the expense of computing time.
- The method is sensitive to the initial conditions and so you can only model a pure soliton wave (or in our case periodic soliton wave pulses) if you have the correct initial condition which you can only know if you know the analytic solution to the KdV equation. Otherwise for given initial conditions the wave-form would decay into a series of soliton pulses. Furthermore it would be difficult to generate the generalised cnoidal wave solutions using this method.
- The method is not the most accurate of the methods generated to solve the KdV equation and is perhaps the least accurate of all the methods that have been created as of yet.

Nevertheless, in terms of our problem, the Zabusky-Kruskal method was an efficient and effective algorithm for integrating the Korteweg-de Vries equation, producing reasonably physical results within the space and time intervals we chose, which were shown to correspond with the solitary wave solutions of the equation. As for the problems we had with instability and accuracy, the could be compensated for by a suitable choice of input parameters in return for longer computation.

References

[1] *CO23 Practical Script,* Oxford Physics, 2009

[2] A. O'Hare, *Numerical Methods for Physicists,* Oxford Physics, 2005

[3] R. H. Enns, *It's a Nonlinear World*, 1st edition, SUMAT, 2010

[4] P. G. Drazin and R. S. Johnson, *Solitons: An Introduction*, 2nd edition, CUP, 1989

[5] N. J Zabusky and M. D. Kruskal, Physical Review Letter Series **15**, 240-243 (1965)

[6] M. J. Albowitx, *Nonlinear Dispersive Waves: Asymptotic Analysis and Solitons*, CUP, 2011

[7] T. Dauxois and M. Peyrard, *Physics Of Solitons,* CUP, 2006

Appendix A Source Code

The source code used to compute the time evolution of a wave-form satisfying the KdV equation using the Zabusky-Kruskal method up to a specified input time:

```
/************************************************************************
*  CO23 Soliton
*  A program designed to compute the time evolution of a wave pulse solution
*  given an initial waveform corresponding to the analytic solution to the
*  KdV equation
*************************************************************************/

#include <math.h>
#include <stdio.h>
#include <stdlib.h>

int main()
{   int i, j=1;
    double U[3][56];
    double t, dt=0.002, dx=0.02, d=0.03, x;

/*The user inputs an end time. The program will run until it reaches this end time
  and the final values printed*/
    printf("Enter a value for the time limit t\n");
    scanf("%lf", &t);

/*We set the initial values to be those of the analytic soliton solution*/
    for (i=3; i<53; i++)
    { U[0][i]=3*(1/(cosh((((double)(i-3)*dx)-0.5)/(0.06))))*(1/(cosh((((double)(i-3)*dx)-
0.5)/(0.06))));}

/*Here the array is made periodic, i.e the edge values are set so that the
  resulting data looks periodic*/
    for (i=0; i<3; i++)
    { U[0][i]=U[0][i+50];}

    for (i=53; i<56; i++)
    { U[0][i]=U[0][i-50];}

     FILE *fout;

    if ((fout=fopen("S0.txt","w"))==NULL)
    { printf("Cannot open %s\n", "S0.txt");
    exit(EXIT_FAILURE);}

    for (i=0; i<56; i++)
    {x=(i-3)*dx;
     fprintf(fout,"%f\t %f\n", x, U[0][i]);}

 /*The initial step in the Zabusky-Kruskal method is implemented; the edge
   values of the array are made to conform with the periodic boundary conditions
   and with the resulting values exchanged from the third column of the array
   to the first one*/
    for (i=3; i<53; i++)
    {U[2][i]=U[0][i]-(dt/(6*dx))*(U[0][i+1]+U[0][i]+U[0][i-1])*(U[0][i+1]-U[0][i-1])
             -((d*d*dt)/(2*dx*dx*dx))*(U[0][i+2]-(2*U[0][i+1])+(2*U[0][i-1])-U[0][i-2]);}

    for (i=0; i<3; i++)
    { U[2][i]=U[2][i+50];}

    for (i=53; i<56; i++)
    { U[2][i]=U[2][i-50];}
```

```c
    for (i=0; i<56; i++)
    {U[1][i]=U[0][i];
     U[0][i]=U[2][i];}

     if ((fout=fopen("S1.txt","w"))==NULL)
    { printf("Cannot open %s\n", "S1.txt");
    exit(EXIT_FAILURE);}

    for (i=0; i<56; i++)
    {x=(i-3)*dx;
     fprintf(fout,"%f\t %f\n", x, U[0][i]);}

 /*The main iterative step of the Kruskal-Zubusky iteration method is implemented
   and is run until the we reach the input time, at which point the array should
   consist of the values of the wave at a position between x=0 and 1 time t*/
    while(j<(t/dt))
    {
        for (i=3; i<53; i++)
        { U[2][i]=U[1][i]-(dt/(3*dx))*(U[0][i+1]+U[0][i]+U[0][i-1])*(U[0][i+1]-U[0][i-1])
                  -((d*d*dt)/(dx*dx*dx))*(U[0][i+2]-(2*U[0][i+1])+(2*U[0][i-1])-U[0][i-
2]);}

        for (i=0; i<3; i++)
        { U[2][i]=U[0][i+50];}

        for (i=53; i<56; i++)
        { U[2][i]=U[0][i-50];}

        for (i=0; i<56; i++)
        { U[1][i]=U[0][i];
          U[0][i]=U[2][i];}

        j++;
    }

    if ((fout=fopen("S2.txt","w"))==NULL)
    { printf("Cannot open %s\n", "S2.txt");
    exit(EXIT_FAILURE);}

    for (i=0; i<56; i++)
    {x=(i-3)*dx;
     fprintf(fout,"%f\t %f\n", x, U[0][i]);}

     fclose(fout);

     exit(0);
     }
```

Appendix B Finite Difference Approximations

When a function $U(x,t)$ and its derivatives are single-valued, finite and continuous functions then we can write by Taylor's theorem

$$\begin{aligned} U(x+\delta x,t) = U(x,t) + \delta x U_x(x,t) + \frac{\delta x^2}{2} U_{xx}(x,t) + \frac{\delta x^3}{6} U_{xxx}(x,t) \\ + \frac{\delta x^4}{24} U_{xxxx}(x,t) + \frac{\delta x^5}{120} U_{xxxxx}(x,t) + \cdots \end{aligned} \tag{B.1}$$

$$U(x-\delta x,t)=U(x,t)-\delta x U_x(x,t)+\frac{\delta x^2}{2}U_{xx}(x,t)-\frac{\delta x^3}{6}U_{xxx}(x,t)$$
$$+\frac{\delta x^4}{24}U_{xxxx}(x,t)-\frac{\delta x^5}{120}U_{xxxxx}(x,t)+\cdots$$

We can add these to get

$$U(x+\delta x,t)+U(x-\delta x,t)=2U(x,t)+\delta x^2 U_{xx}(x,t)+\mathcal{O}(\delta x^4) \tag{B.2}$$

and subtract them to get

$$U(x+\delta x,t)-U(x-\delta x,t)=2\delta x U_x(x,t)+\mathcal{O}(\delta x^3) \tag{B.3}$$

We can rearrange (B.2) to get

$$U(x,t)\approx\frac{1}{3}\{U(x+\delta x,t)+U(x,t)+U(x-\delta x,t)\} \tag{B.4}$$

with a leading error of order δx^2 on the right hand side.

And we can rearrange (B.3) to get

$$U_x(x,t)\approx\frac{1}{2\delta x}\{U(x+\delta x,t)-U(x-\delta x,t)\} \tag{B.5}$$

which also has an error of δx^2.

By a similar procedure we can discretise the partial derivative with respect to time, getting

$$U_t(x,t)\approx\frac{1}{2\delta t}\{U(x,t+\delta t)-U(x,t-\delta t)\} \tag{B.6}$$

with a leading error of order δt^2.

As for third order derivatives we can perform two further Taylor expansions

$$\begin{aligned}U(x+2\delta x,t)&=U(x,t)+2\delta x U_x(x,t)+2\delta x^2 U_{xx}(x,t)+\frac{4}{3}\delta x^3 U_{xxx}(x,t)\\&\quad+\frac{2}{3}\delta x^4 U_{xxxx}(x,t)+\frac{4}{15}\delta x^5 U_{xxxxx}(x,t)+\cdots\\U(x-2\delta x,t)&=U(x,t)-2\delta x U_x(x,t)+2\delta x^2 U_{xx}(x,t)-\frac{4}{3}\delta x^3 U_{xxx}(x,t)\\&\quad+\frac{2}{3}\delta x^4 U_{xxxx}(x,t)-\frac{4}{15}\delta x^5 U_{xxxxx}(x,t)+\cdots\end{aligned} \tag{B.7}$$

and combining these with the previous Taylor expansions we get

$$\begin{aligned}U(x+2\delta x,t)-2U(x+\delta x,t)+2U(x-\delta x,t)-U(x-2\delta x,t)\\=2\delta x^3 U_{xxx}(x,t)+\mathcal{O}(\delta x^5)\end{aligned} \tag{B.8}$$

We can rearrange this to get the approximation

$$U_{xxx}(x,t)\approx\frac{1}{3}\{U(x+2\delta x,t)-2U(x+\delta x,t)+2U(x-\delta x,t)-U(x-2\delta x,t)\} \tag{B.9}$$

which has a truncation error of $O(\delta x^2)$

Appendix C Stability of the Zabusky-Kruskal Algorithm

The issue of computational stability in numerically solving nonlinear PDEs with explicit schemes is a very important one. Von Neumann stability analysis is used to determine the stability of the solution. Suppose that u^0 is the exact solution of the proposed numerical scheme, while $u = u^0 + w$ is the actual solution where w is the error. For example, w can arise because of using a finite number of digits in the computer solution. Substituting u into the proposed algorithm, one retains only linear terms in w. Then w can be written as a fourier series and the behaviour representative of the term $w_n^v = e^{in\Delta x\theta}e^{iv\Delta y\lambda}$ can be examined as time increases. Setting $\lambda = \alpha + i\beta$ with $\alpha, \beta \in \mathbb{R}$, then $w_n^v \sim e^{-i\beta v\Delta y}$. If $\beta < 0$ then the proposed numerical scheme is unstable since u_n^v will diverge as $v\Delta y$ increases so for stability one must have $\beta > 0$.

Setting $r = \frac{\Delta t}{\Delta x^3}$ the Zabusky-Kruskal algorithm is

$$u_n^{v+1} = u_n^{v-1} - \Delta x^2 r\left(\frac{u_{n+1}^v + u_n^v + u_{n-1}^v}{3}\right)(u_{n+1}^v - u_{n-1}^v) - \delta^2 r(u_{n+2}^v - 2u_{n+1}^v + 2u_{n-1}^v - u_{n-1}^v) \tag{C.1}$$

Substituting $u_n^v = (u_n^v)^0 + w_n^v$, retaining linear terms in w and neglecting the $\Delta x^2 r$ term compared to the r term we get

$$w_n^{v+1} = w_n^{v-1} - \delta^2 r(w_{n+2}^v - 2w_{n+1}^v + 2w_{n-1}^v - w_{n-1}^v) \tag{C.2}$$

Setting $u_n^v = e^{in\Delta x\theta}e^{iv\Delta y\lambda}$ and simplifying we obtain

$$\begin{aligned} e^{i\lambda} &= e^{-i\lambda} - \delta^2 r\left(e^{2i\theta} - 2e^{i\theta} + 2e^{-i\theta} - e^{-2i\theta}\right) \\ &= e^{-i\lambda} + 2i\delta^2 r[2\sin(\theta) - 2\sin(2\theta)] \end{aligned} \tag{C.3}$$

Multiplying by $e^{i\lambda}$ and solving the resulting quadratic yields

$$\begin{aligned} e^{i\lambda} &= iR \pm \sqrt{1 - R^2} \\ R &\equiv \delta^2 r[2\sin(\theta) - 2\sin(2\theta)] = \delta^2 r f(\theta) \end{aligned} \tag{C.4}$$

Setting $\lambda = \alpha + i\beta$ with $\alpha, \beta \in \mathbb{R}$, and taking the absolute value of (C.4) we have

$$\left|e^{i\lambda}\right| = e^{-\beta} = \left|iR \pm \sqrt{1 - R^2}\right| \tag{C.5}$$

For stability we require that $\beta > 0$, i.e, that $R = \delta^4 r^2 f^2(\theta) < 1$
The maximum value of $f(\theta)$, namely, $f_{max} = \frac{3\sqrt{3}}{2}$ occurs when $\theta = \frac{2\pi}{3}$ radians.
So the numerical scheme will be stable for

$$r = \frac{\Delta t}{\Delta x^3} < \frac{1}{\delta^2 f_{max}} = \frac{2}{3\sqrt{3}\delta^2} \tag{C.6}$$

For $\Delta x = 0.02, \delta = 0.03$ we must have $\Delta t < 0.0034$ for stability which agrees with our numerical findings. If this inequality is violated, increasingly large unphysical oscillations can occur in the numerical solution.