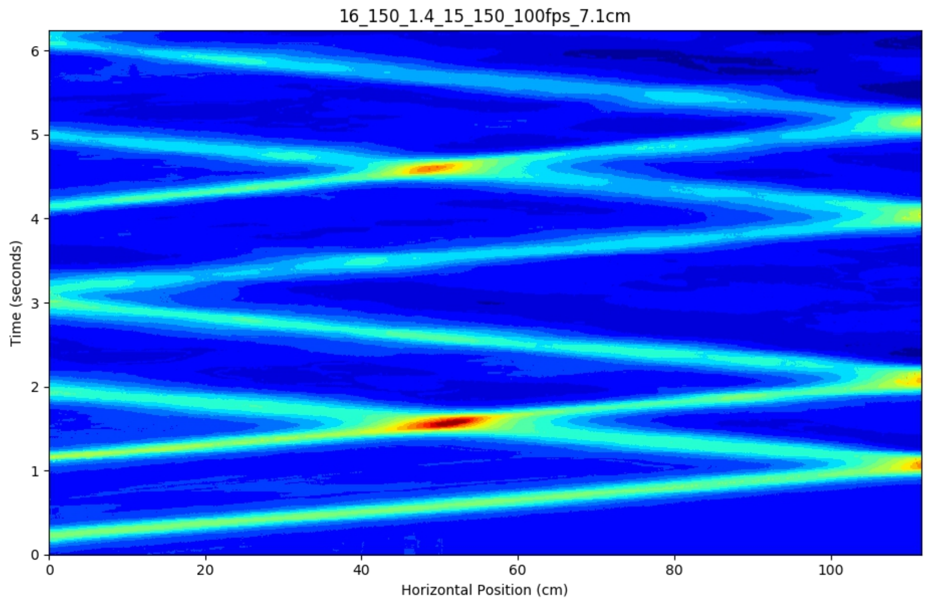
University of Toronto
ADVANCED PHYSICS LABORATORY

SOL

Solitons



Revisions
September 5, 2019: Stephen Morris <smorris@physics.utoronto.ca>

Please send any corrections, comments, or suggestions to the professor currently supervising this experiment, the author of the most recent revision above, or the Advanced Physics Lab Coordinator.

This pdf file has live links that look like this.

1 Introduction

A *soliton* is a very general and ubiquitous kind of nonlinear wave. Unlike a normal linear wavetrain, a soliton is tightly localized and has a single maximum, hence the name “solitary wave”, or soliton. It retains its form while propagating and even survives collisions with other solitons. These properties are made possible by a balance between amplitude nonlinearity (which makes larger amplitude waves move faster) and dispersion (which makes wavetrains of different frequency move at different speeds). Solitons occur as surface waves on fluids (as in this experiment) [1, 2], in nonlinear optical pulses in fibers [3], in theories of signal propagation in nerve fibers [4], in Bose-Einstein condensates [5], and in many other physical and mathematical contexts.

The 1834 discovery of solitons by John Scott Russel is a classic moment in the history of nonlinear science. Russel chased a soliton on horseback as it propagated along the Union canal in Scotland, and later reproduced and studied the effect in a water channel similar to the one used in the present experiment. Solitons crucially involve nonlinearity, but, unlike most nonlinear problems, are amenable to analytic theory. The literature on the mathematics and physics of solitons is vast. An important, exactly solvable, nonlinear partial differential equation, the Korteweg-de Vries (KdV) equation, which has soliton solutions, can be derived from the fluid equations for shallow water in a channel [1, 2].

In this experiment, we will generate and collide solitons and study their properties. We will also explore the application and limitations of the KdV soliton as a model for localized nonlinear surface waves.

2 Theory

A brief outline of the theory is presented in Appendix A.

3 Experiment

3.1 overview

The experiment consists of a long channel-shaped tank with an automated wavemaker at one end. A portion of the tank has an LED back light which may be imaged by a 120 fps camera. Waves are visualized by dying the water with red food colouring and detecting their edges with python codes.

The wavemaker consists of a motorized paddle that extends to the bottom of the channel. The paddle moves along a rail between starting and stopping positions which are defined by limit switches. The motor controller may be programmed to make various impulsive paddle motions between these two limits, launching waves down the tank. Waves may reflect off the left end of the tank.

In addition to observing nonlinear solitary waves and their collisions, the student may allow waves to interact with ramps or other topography, or track the motion of floating objects moved by waves.

3.2 important safety notices

- **This experiment uses water.** Take care when filling or draining the tank to avoid spills. Water on the floor should be mopped up promptly. Only fill the tank to less than half full. Try gentle motions first to prevent splashing at the end of the tank or at the wavemaker.
- **Stay clear of moving mechanical parts.** Avoid putting your fingers near the moving wavemaker. Do not interfere with the limit switches. The wavemaker paddle should not touch the sides of the tank; if this happens, it needs adjustment and should not be operated.
- **Do not alter the basic motor control software.** The basic functions in the python code `servo_class.py` must not be changed. Motor or wavemaker damage may result.
- **Red dye may stain clothes**. The red dye is food safe, but may be hard to wash out. Avoid getting it on nice clothes or use a lab coat.

3.3 general remarks on procedure

- Fill the tank slowly by closing the valve below the outlet of the tap and cracking open the main valve on the tap. Watch the level and do not overfill.
- Always drain the tank when you are done using it. Water left in the tank quickly becomes slimy, especially if food dye is present. Water left in the fill line should be drained out by opening the valve below the outlet of the tap.
- Wipe the empty tank with a damp soft sponge, taking care not to scratch it. The tank may be gently cleaned with water and dish soap to remove any "bathtub ring".
- Red food dye should be mixed into the water to look uniform. The dye colour needs to be solidly red, strong enough to be clearly detected by the image analysis. Something like this example.
- Position the camera on the tripod so that the full length of the illuminated part of the tank is just framed. Make sure that the image is properly horizontal and focussed.
- The camera and tripod should be moved aside when you are done using them, but not before a length calibration image is taken.
- Clear all your images and code off the computer when you are completely done with them. Otherwise, put them in a directory with your name on it. Remember that the experiment may be used by someone else during the week.

3.4 suggested procedure

3.4.1 controlling the wavemaker

The python code `servo_class.py` contains various low-level operating functions for the motor controller. You should familiarize yourself with them, but **they should not be altered**, or equipment damage may result. The code `wavemaker.py` contains higher level commands that you can use or modify to control the wavemaker from a python command line. Read the comments in these codes to see how they work.

To launch a wave from the python command line, start by typing

```
from wavemaker import * .
```

This import will issue a `Servo.config()` command that initializes the motor controller. This initialization **must always** be done before any other commands, because it activates the limit switches (via the DL1 command). Once the controller is configured, the wavemaker paddle can be sent to its home (leftmost) position with the command

```
s.home() .
```

To have consistent initial conditions, it's a good idea to return the wavemaker to the home position and wait for the water to come to rest before launching new waves. Once the wavemaker is home, a single wave can be launched with the command

```
wave(10, 100) .
```

This example creates a single wave with a paddle displacement of 10 cm and a paddle acceleration and deceleration of 100 cm/s^2. Similarly, the command

```
two_wave(10, 100, 2, 10, 100)
```

launches two of the same waves in succession, separated by a delay of 2 s. You can customize the motion to have different acceleration and deceleration using the optional `af` argument in `wave()`. The function `two_wave_custom()` allows two customized waves in succession.

More wavemaker protocols are possible: consult the `wavemaker.py` code and the general command reference for the ST5-S motor controller.

After each shot, use the `s.home()` command to return the wavemaker to its home position.

3.4.2 setting up the camera and lighting

The DC voltage applied to the LED back light should be fixed at about 60V. The camera is controlled by the `FlyCapture2` program. Adjust the aperture and focus of the camera with water in the tank by zooming up the image on the screen to see individual pixels. Leave the focus and aperture settings fixed for the experiment. A wide aperture will cause the automatic camera to use a fast shutter speed, which is a good choice.

To calibrate lengths in the images, take a reference image of the coloured meter stick inside the tank with the back lighting off. Do not allow the camera to be moved between calibration images. To be careful, a calibration image may be taken before and after each series of experiments.

You can use all the default settings of the `FlyCapture2` program except the frame rate. In the `camera control` dialog window, uncheck the `auto` box on the frame rate. The frame rate can be set to a precise number by typing in the box. The maximum is 120 fps.

When you press record, you will get the `record settings` window. Here you should set the directory and file name for the run. It is a good idea to use a name that contains information not already recorded by the camera, such as the water depth, the wavemaker parameters, and the frame rate. Something like `H7cm_d10cm_a120cmsec2_120fps`. The program will add a date and time stamp and a frame number to the name. Use a new directory to hold the images for each run.

`Saving options` should be set to capture a few hundred frames; a few seconds worth is usually enough. `Image format` should be set to TIFF, with `compression method` set to LZW.

Click the `Start Recording` button *just before* you actuate the wavemaker.

3.5 Data analysis

3.5.1 data analysis tools

In the experiment the data are mostly in the form of *images*. An image is a 2D position measurement. **Every pixel is a measured data point.** The camera captures 8 bit RGB colour images. Each colour pixel is a triplet (R, G, B) of red, green and blue values between 0 - 255. We will be converting the colour images into 8 bit binarized images — images for which every pixel is either black (greyscale level 0) or white (greyscale level 255). The shape of the wave is extracted from the binarized images by an edge detection operation.

All images must be in a format that preserves pixel-level information with lossless compression. Tagged Image Files (.tif) format with LZW compression achieves this. The camera should be set to save images in this format. Do not use .jpg images except to be used qualitatively as frames in movies.

To analyze images, we will use a combination of python codes and an open source application called **Image J**. It is easy to take an overwhelming number of images in this experiment. Storage capacity may become a limitation. Be judicious about choosing frame rates, deleting useless images, decimating the data *etc.* to avoid having more images than can be reasonably analyzed. It's a good idea to do some preliminary runs and then try the analysis on a few images first before taking any detailed data. Image analysis may require long computation times.

Read the comments in the python codes for details about how they work. Only a rough outline is given here. Of course you can modify any of the analysis codes as you see fit. The following are merely suggestions.

Report any bugs or suggested improvements to these codes to Stephen Morris <smorris@physics.utoronto.ca>.

3.5.2 data analysis workflow

- The raw images must first be cropped to exclude everything except a long narrow image of the red water and the white background. This may be done with the code called `crop_SOL.py`. If you would like to make a movie of cropped frames, you can use `crop_movie_SOL.py` (which calls `ffmpeg` to construct the movie).
- The cropped images may then be processed by `edge_SOL.py`, which uses functions from `image_analysis_SOL.py` to detect the edge of the wave. Isolating the edge involves subtracting the different (RGB) color values from one another and comparing the result to a threshold. The thresholds required may be guessed by scrutinizing the raw images with **Image J**'s pixel inspector. Then a specialized edge detecting function is applied to make a logical binary image. The x, y coordinates of the edge pixels are extracted and the edge position data is stored as a text file. Various binary images are generated to check the results.
- The detected edge shapes may be fit to a sech^2 soliton shape using the code `fit_wave_shape_SOL.py` which can be used to extract the peak position, width and amplitude of single solitons.

- Finally, the detected edges may be contour plotted in spacetime using `plot_spacetime_SOL.py`. One such contour plot is shown at the top of this document.

Further data analysis, say of wave speeds *etc.* is left to the student. A python code called `kdv_diss_SOL.py` to simulate the KdV equation, with or without dissipation, is also provided. See Appendix A for a discussion of dissipation.

4 questions

The following are some rough ideas for things to study.

4.1 single waves

- Create and study a range of waves with different amplitudes, using various water depths H. Do their speeds, widths and amplitudes scale as predicted by the KdV equation?
- Over what range of wavemaker parameters are KdV-like soliton waves generated? When does KdV-like behavior break down and how do the resulting (breaking) waves propagate?
- How does the presence of dissipation affect the waves? Can the effect of dissipation be adequately modeled by adding a phenomenological dissipative term to the KdV equation?
- Do the waves survive reflection off the end of the tank? What are the characteristics of reflected waves? Are they still KdV-like?
- Does the wave suffer a small delay when it reflects from the end of the tank? This effect is related to the nonlinear phase shift caused by a head-on collision between a soliton and its mirror image twin.

4.2 wave collisions

- Collide two waves head-on by hitting a direct wave with another that has been reflected from the end of the tank. Do the respective waves survive the collision?
- Can you observe and quantify the nonlinear phase shift created when two oppositely propagating waves pass through each other?
- Try to collide two waves by having a faster, larger amplitude wave overtake a slower, smaller amplitude one. (This corresponds to a "classic" KdV soliton collision in a moving reference frame). Try to observe and quantify the nonlinear phase shift, and compare it to the KdV prediction.

The following questions might require the development of additional apparatus components and analysis methodologies.

4.3 further ideas

- Consider the effect of varying depth on the propagation of the waves. For example, a "beach" at one end of the tank would allow a study of shoaling waves whose amplitude will grow to breaking.
- A sudden localized depth change below the surface could be created with a plastic block in the bottom of the tank. What effect does this have on a nonlinear wave passing over it?
- Consider the trajectory of a very light and small floating object that can be tracked by the camera as a wave passes. What do you observe for small amplitude linear waves *vs.* large amplitude soliton-like nonlinear waves? Breaking waves? What does theory predict?

References

(Electronic versions, if available, are accessible by clicking on the title.)

[1] T. Dauxois and M. Peyrard, *Physics of Solitons*, Cambridge University Press (2006).

[2] R. S. Johnson, *A Modern Introduction to the Mathematical Theory of Water Waves*, Cambridge University Press (2010).

[3] A. Hasegawa, *Optical Solitons in Fibers*, *Springer Tracts in Modern Physics*, **16**, Springer, (1989).

[4] A. C. Scott, *The Electrophysics of a Nerve Fiber*, *Rev. Mod. Phys.*, 47, 487–533 (1975).

[5] L. M. Aycock, H. M. Hurst, D. K. Efimkin, D. Genkina, H-I. Lu, V. M. Galitski, and I. B. Spielman, *Brownian motion of solitons in a Bose–Einstein condensate*, *PNAS*, **114**, 2503 (2017).

[6] D. J. Acheson, *Elementary Fluid Dynamics*, Oxford University Press (1990).

[7] W. Craig, P. Guyenne, J. Hammack, D. Henderson, and C. Sulem, *Solitary water wave interactions*, *Phys. Fluids*, **18**, 057106 (2006).

[8] C. S. Gardner, J. M. Greene, M. D. Kruskal, R. M. Miura, *Method for Solving the Korteweg-deVries Equation*, *Phys. Rev. Lett.*, **19**, 1095 (1967).

[9] H. Borluka and H. Kalischb, *Particle dynamics in the KdV approximation*, Wave Motion, **49**, 691 (2012).

A Nonlinear wave theory

Water waves can be extremely complex, as a trip to beach will convince you. In order to account for wave phenomena, one must solve the highly nonlinear field equations for the flow velocity $\vec{\mathbf{u}}$ and pressure p in the interior of the fluid, subject to the boundary conditions at the rigid bottom surface (if there is one) and at the *moving* upper surface. The equations of motion for the position of this surface are the objective of the calculation, and must be found in a self-consistent way while solving for the interior flow. We will consider only two dimensional flows and make several simplifying approximations. Fig. 1 shows the geometry of a localized wave. A detailed derivation of the nonlinear wave equation can be found in Appendix A of Ref. [1]. We only outline the theory here.

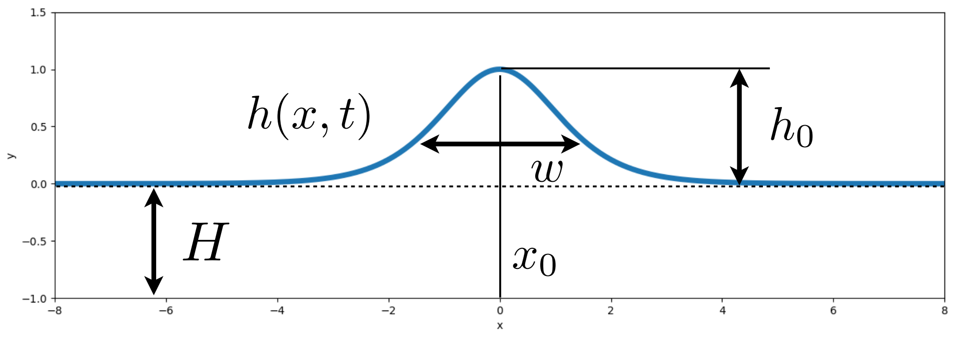


Figure 1: The soliton wave geometry.

Let $h(x,t)$ be the disturbance of the surface relative to a layer of constant depth H. The fluid has constant density ρ and is acted upon by gravity with acceleration g. We assume the fluid has zero viscosity and zero surface tension. These approximations turn out to be pretty good for water on the scale of our tank. We further assume that the fluid is not turbulent, so that $\nabla \times \vec{\mathbf{u}} = 0$. In this case, $\vec{\mathbf{u}}$ can be found from the gradient of a velocity potential ϕ, so that $\vec{\mathbf{u}} = \nabla\phi$. Then ϕ obeys a Laplace equation $\nabla^2\phi = 0$. All of the difficulty of the problem is related to the boundary conditions on ϕ at the moving boundary $h(x,t)$. The book by Acheson [6] has a good treatment of this problem for the linearized case.

An equation for the surface position can be found in the limit when two dimensionless parameters are small; they are

$$\epsilon = A/L \qquad \text{and} \qquad \delta = H/L \ , \tag{1}$$

where A is the length scale of the amplitude of the wave (the scale of the vertical surface deformation h) and L is the scale of the horizontal variation of the wave. L will be the wavelength λ for periodic wavetrains and the width w of the peak for localized disturbances. $\epsilon \ll 1$ is the familiar small amplitude limit, while $\delta \ll 1$ corresponds to the *shallow water approximation.* These two small parameters are physically different and the theory requires a double expansion in both. The nonlinear equation with soliton solutions emerges when $\epsilon \sim \delta^2 \sim 1$. In the lab frame, in physical units, the result of a careful derivation (see Appendix A of Ref. [1]) is

$$\left(\frac{1}{c_0}\right)h_t + h_x + \epsilon\left(\frac{3}{2H}\right)hh_x + \delta^2\left(\frac{H^2}{6}\right)h_{xxx} = 0 \ , \tag{2}$$

where the subscripts denote partial derivatives, $h_x = \partial h/\partial x$, *etc.* The speed c_0 is given by

$$c_0 = \sqrt{gH} \ . \tag{3}$$

This equation was first found by Boussinesq in 1877 (he reported it only in a footnote!) and rediscovered by Korteweg and de Vries in 1895 (over 60 years after Russel's discovery). The ϵ term is the crucial nonlinearity, while the δ term provides weak dispersion. Section 1.1.3 of Ref. [1] has a nice discussion of the physical effects of the various terms.

A.1 The linear, shallow water limit

Taking the limits $\epsilon \to 0$ and $\delta \to 0$ in Eqn. 2, we get the linear, shallow water wave equation

$$\left(\frac{1}{c_0}\right) h_t + h_x = 0 \ , \tag{4}$$

which has simple solutions $h(x,t) = f(x - c_0 t)$ for *any* function f. This corresponds to an arbitrary wave with shape $h(x,0) = f(x)$ at $t = 0$ that thereafter moves toward $+x$ with speed c_0, without changing its shape.

Tsunami waves travel at speed c_0, because they have very small amplitudes and their lateral extent is much greater than the depth of the ocean; they become nonlinear only near shore.

Linear surface waves have interesting dispersion when the shallow limit $\delta \to 0$ is relaxed, especially when surface tension is included [6].

A.2 The KdV equation

Setting $\epsilon = \delta^2 = 1$ in Eqn. 2 gives the dimensional version of the KdV equation, in the lab frame. It is traditional to simplify this equation in two ways. First, we move to a reference frame moving toward $+x$ with speed c_0 by changing to variables $(X, T) = (x - c_0 t, t)$. The result is

$$\left(\frac{1}{c_0}\right) h_T + \left(\frac{3}{2H}\right) h h_X + \left(\frac{H^2}{6}\right) h_{XXX} = 0 \ . \tag{5}$$

This just eliminates the h_x term. Then we change to (non-obvious) dimensionless length and time variables

$$\xi = X/X_0, \qquad \eta = h/X_0, \qquad \tau = T/T_0 \ , \quad \text{with} \tag{6}$$
$$X_0 = (2/3)^{1/3} H, \quad \text{and} \quad T_0 = 4(H/c_0) \ . \tag{7}$$

This results in the standard dimensionless form of the celebrated KdV equation

$$\eta_\tau + 6\eta\, \eta_\xi + \eta_{\xi\xi\xi} = 0 \ . \tag{8}$$

This equation is one of the most studied nonlinear PDEs there is. It has many types of exact solutions, a lot of which were not discovered until numerical methods became practical in the 1960s. A particularly elegant and general exact solution method known as the *inverse scattering transform* was discovered in 1967 [8]. Surprisingly, it involves solving the quantum scattering problem for an associated linear Shrödinger equation!

The exact, single soliton solution of Eqn. 8 is

$$\begin{aligned}\eta(\xi,\tau) &= \eta_0 \operatorname{sech}^2\left[\sqrt{\frac{\eta_0}{2}}(\xi - 2\eta_0\tau)\right] &(9)\\ &= \eta_0 \operatorname{sech}^2\left[(\xi-\beta\tau)/\alpha\right] , &(10)\end{aligned}$$

where the only free parameter is η_0, the peak amplitude, and $\operatorname{sech}(x) = 1/\cosh(x)$. This shape is illustrated in Fig. 1. Notice that once the amplitude η_0 is selected, the dimensionless wave speed in this frame, $\beta = 2\eta_0$ is completely determined, as is the dimensionless width $\alpha = \sqrt{2/\eta_0}$. Taller solitons are narrower and move faster. Their shape is preserved as they move. The amplitude η_0 is always positive; there are no negative KdV solitons. It is important to realize that Eqn. 8 is valid only in a moving reference frame with $c_0 > 0$. All solitons of the form of Eqn. 10 move toward $+\xi$ with speed $\beta \geq 0$ in this reference frame, and never move toward $-\xi$. In the limit $\eta_0 \to 0$, they come to a standstill in this reference frame, which corresponds in the lab frame to a low amplitude linear wave moving toward $+x$ at c_0.

Eqn. 8 has an infinite number of solutions containing numerous localized solitons of different amplitudes, moving in the same direction at different speeds. An exact, two soliton solution is given in Ref. [1]. Eqn. 8 also has periodic solutions that look like infinite chains of equally spaced, equal amplitude solitons called *cnoidal waves*. The latter map smoothly onto the usual sinusoidal waves in the linear limit $\eta_0 \to 0$. It can be proven [1] that *any* initial condition eventually breaks up into a spreading flock of localized solitons as $\tau \to \infty$.

If we go back and unwind the nondimensionalization and coordinate transformations of Eqn. 7, we find that a single soliton solution to Eqn. 2, with $\epsilon = \delta^2 = 1$, in the lab frame is given by

$$\begin{aligned}h(x,t) &= h_0 \operatorname{sech}^2\left[\frac{1}{2H}\sqrt{\frac{3h_0}{H}}\left(x - c_0\left[1+\frac{h_0}{2H}\right]t\right)\right] &(11)\\ &= h_0 \operatorname{sech}^2\left[\frac{x-Ut}{w}\right] , &(12)\end{aligned}$$

corresponding to a wave with amplitude h_0,

$$\text{speed} \quad U = c_0\left[1+\frac{h_0}{2H}\right] \qquad \text{and width} \quad w = 2H\sqrt{\frac{H}{3h_0}} . \qquad (13)$$

The waves are always supersonic, in the sense that $U > c_0$.

A.3 Soliton collisions

Solitons survive collisions unscathed, which is why their nickname ends in "-on", denoting particle-like behaviour. Collisions can happen in two distinct ways; within the co-moving reference frame of the KdV equation, a faster (taller) soliton can collide with and overtake a slower (shorter) one. This is the classic case. In the lab frame, two oppositely travelling solitons can collide head on. This case is easier to achieve in our tank. In both cases, neglecting dissipation, the solitons are predicted to emerge from their interaction with essentially unchanged shapes and speeds. But since solitons are nonlinear waves, they cannot simply superpose during the collision. It turns out that while their shapes and speeds survive, their outgoing positions are slightly advanced or delayed by their interaction during the collision — a purely nonlinear phase shift effect.

A.3.1 Overtaking collisions

Section 1.2.2 of Ref. [1] has a discussion of the exact two soliton solution of KdV. See also this web page for nice animations. The exact expression is sufficiently complicated that its just as easy to plot numerically. Fig. 2 shows a spacetime view of an overtaking collision, calculated numerically using the code `kdv_collision_SOL.py` (which also allows for dissipation, see below). You can clearly see that the faster soliton has been advanced, while the slower one is retarded, after they emerge from the collision.

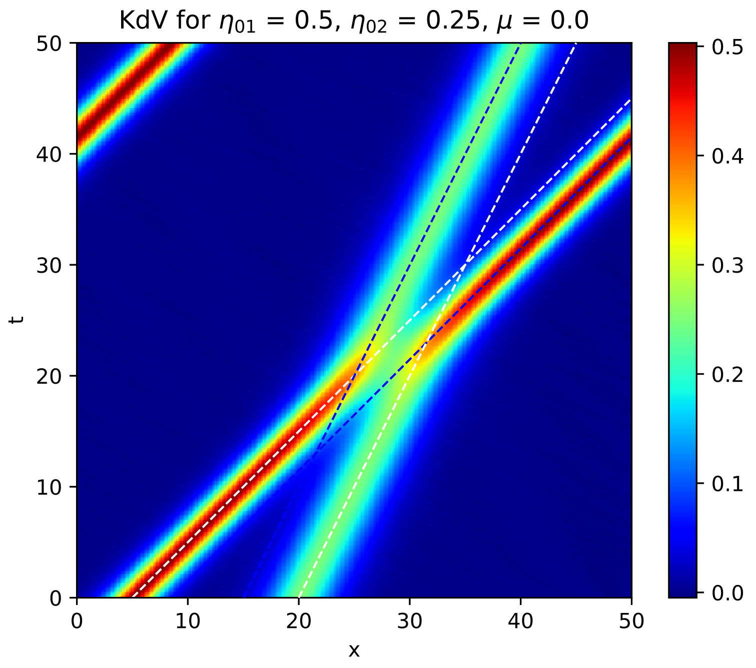


Figure 2: An overtaking collision of KdV solitons with amplitudes $\eta_{01} = 0.5$ (faster) and $\eta_{02} = 0.25$ (slower). The dotted lines trace the peak positions before (white) and after (blue) the collision. The shifts are calculated from Eqn. 14

The nonlinear phase shifts are given exactly [1] by

$$D_1 = 2\Delta\sqrt{\frac{2}{\eta_{01}}} \quad \text{(faster)} \qquad D_2 = -2\Delta\sqrt{\frac{2}{\eta_{02}}} \quad \text{(slower)}\ , \tag{14}$$

with

$$\Delta = \tanh^{-1}\left(\sqrt{\frac{\eta_{02}}{\eta_{01}}}\right)\ . \tag{15}$$

You can play around with the soliton parameters in the code `kdv_collision_SOL.py`.

A.3.2 Head-on collisions

This case is considered by [7].

A.4 The effect of dissipation

When deriving the KdV equation, two assumptions were made: that the fluid had zero viscosity and that the flow was two-dimensional. But the water does have a small viscosity which causes some dissipation. Weakly viscous flow near a rigid wall leads to the formation of thin boundary layers. Within these layers, the flow is dragged to a stop at the wall by viscous forces. In our tank, boundary layers form on the bottom and sides of the tank as the wave passes by. These have two effects: they slow the bulk flow down and also make it less two-dimensional. A complete 3D model of the viscous dissipation is rather complex, but the main effects on the waves may be modelled by adding a phenomenological diffusive term to the KdV equation:

$$\eta_\tau + 6\eta\,\eta_\xi + \eta_{\xi\xi\xi} - \mu\eta_{\xi\xi} = 0\ , \tag{16}$$

where the new dimensionless parameter $\mu > 0$ is small, $\mu \sim 0.01$. This new term spoils all the nice exact solutions of the KdV equation, but of course if μ is sufficiently small, we expect the solutions to be nearly the same as the exact ones for $\mu = 0$. It helps that μ does *not* multiply the highest derivative in the equation, so this new term is *not* a so-called singular perturbation.

Eqn. 16 is straightforward to solve numerically. The provided python code `kdv_diss_SOL.py` can be used to solve the KdV equation, with and, by setting $\mu = 0$, without dissipation. In general, $\mu \neq 0$ causes solitary waves to slowly decay in amplitude, and to become wider and slow down with time. The code allows these effects to be studied numerically for various values of μ.

In the lab frame, the dimensional equation corresponding to Eqn. 16 is

$$\left(\frac{1}{c_0}\right)h_t + h_x + \left(\frac{3}{2H}\right)hh_x + \left(\frac{H^2}{6}\right)h_{xxx} - Mh_{xx} = 0\ , \tag{17}$$

with

$$M = \left(\frac{2}{3}\right)^{1/3} H\,\mu\ = 0.874\,H\mu. \tag{18}$$

The dimensional dissipation coefficient M has the units of length.

The literature on dissipative generalizations of KdV is enormous and many references could be added to this discussion.

A.5 The wavemaker and end reflections

Comments about initial conditions and something about reflections from hard boundaries.

A.6 Changes of bottom topography

Ref. [2] has some information about the effects of topography like steps and beaches.

A.7 The trajectory of neutrally buoyant floating objects

The trajectory of a massless neutrally buoyant floating particle due to the passage of a single KdV soliton is discussed in Ref. [9]. Even for nearly linear waves, such trajectories do not exactly close, leading to what is known as Stokes Drift.