

# Finite Difference Solution of the Korteweg & de Vries (KdV) Equation

PHYS 210 Term Project Proposal

Matthew Choptuik

- **Overview**

- The KdV equation is a nonlinear wave equation in one space variable and time which admits interesting “particle-like” solutions known as **solitons**
- The KdV solitons travel with velocities that are dependent on their amplitudes, and solitons with different amplitudes can undergo interactions (collisions) which, when done, leave each soliton unchanged

- **Project Goals**

- To write an MATLAB (octave) code which solves the KdV equation numerically, using second-order finite difference techniques
- To establish correctness of the implementation of the code through convergence tests and comparison with known solutions
- To investigate a variety of initial conditions for the equation, including those describing single and multi-soliton solutions

- **Mathematical Formulation (Equations of Motion)**

- The KdV equation can be written in the form

$$\frac{\partial u}{\partial t} + 12u \frac{\partial u}{\partial x} + \frac{\partial^3 u}{\partial x^3} = 0 \quad (1)$$

where  $u \equiv u(t, x)$

- The equation will be solved as an initial boundary problem on the domain

$$-x_{\max} \leq x \leq x_{\max} \quad 0 \leq t \leq t_{\max} \quad (2)$$

with initial conditions

$$u(0, x) = u_0(x) \quad (3)$$

where  $u_0(x)$  is some specified function, and boundary conditions

$$u(t, -x_{\max}) = u(t, x_{\max}) = 0 \quad (4)$$

- **Numerical Approach**

- The KdV equation will be discretized using a second-order finite difference technique, wherein the continuum domain  $(t, x)$  will be replaced with a discrete grid (lattice) of points  $(t^n, x_j)$  such that

$$x_j = -x_{\max} + jh, \quad j = 1, 2, \dots, n_x \quad (5)$$

$$t^n = n\lambda h, \quad n = 0, 1, 2, \dots, n_t \quad (6)$$

- I will then approximate (discretize) the KdV equation as follows

$$\frac{u_j^{n+1} - u_j^n}{\lambda h} + \mu_t (D_x u_j^n) + 12\mu_t (u_j^n) \mu_t (D_x u_j^n) + \mu_t (D_{xxx} u_j^n) = 0 \quad (7)$$

where the operators  $\mu_t$ ,  $D_x$  and  $D_{xxx}$  are defined by:

$$\mu_t (u_j^n) = \frac{1}{2} (u_j^{n+1} + u_j^n) \quad (8)$$

$$D_x (u_j^n) = \frac{u_{j+1}^n - u_{j-1}^n}{2h} \quad (9)$$

$$D_{xxx} (u_j^n) = \frac{u_{j+2}^n - 2u_{j+1}^n + 2u_{j-1}^n - u_{j-2}^n}{h^3} \quad (10)$$

- **Numerical Approach (continued)**

- When supplemented with discrete versions of the boundary conditions, and assuming that the values of  $u_j^n, j = 0, 2, \dots, n_x$  are known (for  $n = 0$ , these will be determined from the initial conditions), the approximation to the KdV equation becomes a set of nonlinear algebraic equations in the unknowns  $u_j^{n+1}, j = 0, 2, \dots, n_x$
- I will solve this system of equations using a multi-dimensional Newton-Raphson method
- Each iteration of the Newton method will require the solution of a banded linear system which will be solved using MATLAB (octave) functions that are based on LAPACK (Linear Algebra PACKage) routines
- When the Newton method has converged, I will have advanced the discrete solution in the time, and can then repeat the process for the desired number of time steps

- **Visualization and Plotting Tools**

- I will use **xvs** for interactive analysis and generation of mpeg animations, and MATLAB's plotting facilities for plots to be included in my report

- **Testing & Numerical Experiments**

- **Testing**

- Convergence testing: Fix initial data, compute solutions using discretization scales  $h, h/2, h/4, h/8 \dots$ , and ensure that  $O(h^2)$  convergence behaviour is obtained
- Check the numerical results against the known closed-form solutions for single solitons

- **Numerical Experiments**

- Investigate the interaction of two or more solitons with different amplitudes, and measure the expected “phase-shift” effect which results from the interaction (collision)
- Investigate evolution of generic types of initial data (such as Gaussian profiles) to see if solitons generically emerge

- **Project Timeline**

<b>Dates</b>	<b>Activities</b>
10/13–10/24	Do basic research, derive equations & begin code design
10/25–11/15	Implement code
11/16–11/19	Test code
11/20–11/25	Run numerical experiments, analyze data, begin report
11/26–11/28	Finish report
11/28	Submit project! (absolute deadline is 12/03)

- **References**

- R.K.Dodd et al, *Solitons and Nonlinear Equations*, Academic Press, London, (1982)
- <http://bh0.phas.ubc.ca/~matt/Teaching/05Fall/PHYS410/Projects/kdv.pdf>

**QUESTIONS?**  
**COMMENTS?**  
**SUGGESTIONS?**



**THUNDEROUS APPLAUSE!!**