Numerical Approximations to the Boussinesq Equations

Joshua B Wilson

A thesis submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Master of Science in the Department of Physics and Astronomy.

Chapel Hill

2010

Approved by:

Dr. Alberto Scotti

Dr. Charles Evans

Abstract

Joshua B Wilson

Numerical Approximations to the Boussinesq Equations

(Under the direction of Alberto Scotti)

A study of the Boussinesq equations in one dimension is presented. These equations describe the nonlinear wave propagation of a free surface under inviscid, incompressible, and irrotational constraints. Physically, they describe the motion of long waves (compared to the depth of the domain) which find applications in oceanography and coastal engineering. Dispersive properties are examined and numerical solutions are found ($O(\Delta x^2, \Delta t^2)$) using a hybrid Finite Volume / Finite Differencing Method. Since all of the numerical code is original (written in C++/Python), a detailed explanation of the numerical method is given. To validate the numerical model, convergence rates are computed using analytic solutions found by S. Ding and X. Zhao [4]. Special attention is given to solitary waves (”solitons”) which can arise in systems exhibiting weakly non-linear and dispersive properties.

Acknowledgements

This research was funded by the Office of Naval Research (ONR) under grant N00014-05-1-0361. Without that support, none of this would have been possible. I would like to thank my advisor, Alberto Scotti, for guiding me through the past two and a half years of research and for helping me edit and finalize this document. I would also like to thank Ed Santilli for listening to the numerous problems that I encountered along the way and helping me find solutions to those problems.

Table of Contents

List of Figures

1 Introduction

1.1 Boussinesq Equations

For irrotational ($\nabla \times \vec{u} = 0$), incompressible ($\nabla \cdot \vec{u} = 0$) flows with constant density, one can define a velocity potential which satisfies Laplace's equation. If we let the boundary conditions define a free surface and utilize Bernoulli's equation (1.3), we have the following system (see [2]),

$$\nabla \cdot (\nabla \phi) = 0 \qquad \nabla \phi = \vec{u} \tag{1.1}$$

$$\phi_z\Big|_{z=\eta} = \eta_t + \phi_x\Big|_{z=\eta}\eta_x \qquad \phi_z\Big|_{z=-h} = 0 \tag{1.2}$$

$$\{\phi_t + \frac{1}{2}|\nabla\phi|^2\}_{z=\eta} + g\eta = 0 \tag{1.3}$$

where $\eta(x,t)$ is the free surface displacement from $z = 0$, h is the height, ϕ is the velocity potential, $\vec{u} = (u,w) = (\phi_x, \phi_z)$ is the velocity field, and g is the gravitational constant. The second boundary condition in (1.2) states that the bottom of the domain is flat and rigid. Expanding ϕ in powers of z around $z = -h$ and then applying (1.1) and (1.2),

$$\phi(x,z,t) = \phi_o + z\phi_z\Big|_{z=-h} + \frac{z^2}{2!}\phi_{zz}\Big|_{z=-h} + \frac{z^3}{3!}\phi_{zzz}\Big|_{z=-h} + \frac{z^4}{4!}\phi_{zzzz}\Big|_{z=-h} + O(z^5) \tag{1.4}$$

$$\phi(x,t) = \phi_o - \frac{h^2}{2}\phi_{xx} + O(h^4) \tag{1.5}$$

This eliminates the z dependence and the odd terms from the potential expansion. Since the velocity is derived from the potential, this also eliminates the z dependence from the velocity. Upon applying (1.5) to the free surface boundary condition (see Appendix A)

$$\eta_t + \partial_x \int_{-h}^{\eta} u\,dz = 0 \qquad u(x,t) = \phi_x$$

and Bernoulli's equation (1.3), we arrive at the following system of equations in one dimension.

$$u_t + \frac{1}{2}(u^2)_x + g\eta_x = \frac{h^2}{2}u_{xxt} \tag{1.6}$$

$$\eta_t + [u(h+\eta)]_x = 0 \tag{1.7}$$

This is known as the Boussinesq system. If only the constant term in (1.5) is kept, the right hand side of (1.6) would be zero. Doing this is known as the shallow water approximation. This term will be examined in more detail later. Under a few additional assumptions [2], we can write the above system as one equation. This is the familiar form of the Boussinesq equation.

$$\alpha = \frac{h^2 {c_0}^2}{2} \quad \beta = \frac{{c_0}^2}{2h} \quad {c_0}^2 = gh$$

$$\eta_{tt} - {c_0}^2 \eta_{xx} - \alpha \eta_{xxxx} - \beta(\eta^2)_{xx} = 0 \tag{1.8}$$

It is advantageous to non-dimensionalize (1.6) and (1.7). In order to do this, we will apply the following transformations to primed variables,

$$\lambda x' = x \quad hz' = z \quad (\frac{\lambda}{c_0})t' = t \quad a\eta' = \eta \quad (\frac{ag\lambda}{c_0})\phi' = \phi \quad \mu = \frac{h}{\lambda} \quad \varepsilon = \frac{a}{h}$$

μ parameterizes the relative depth and ε parameterizes the non-linearity of the system. For linear considerations, $\varepsilon << 1$. For long waves, $\mu << 1$. 'a' is the initial amplitude of $\eta(x,t)$ and λ is a characteristic wavelength. Dropping the primes, we have,

$$u_t + \frac{\varepsilon}{2}(u^2)_x + \eta_x = \frac{\mu^2}{2} u_{xxt} \tag{1.9}$$

$$\eta_t + [u(1+\varepsilon\eta)]_x = 0 \tag{1.10}$$

Here the Boussinesq parameters become,

$$\alpha = \frac{\mu^2}{2} \quad \beta = \frac{\varepsilon}{2} \quad {c_0}^2 = 1 \tag{1.11}$$

1.2 Dispersive Properties

The steps taken to derive (1.9) and (1.10) allowed non-linearity but required additional assumptions on the form of ϕ, namely (1.4). A solution for ϕ can be found without imposing additional assumptions on the form, but at the expense of only working for the linearized equations ($\varepsilon << 1$). Although all of the analysis presented in this section assumes the linearized form of (1.9) and (1.10), the discussion carries over to the fully non-linear solutions. Start by non-dimensionalizing (1.1), (1.2), and (1.3),

$$\mu^2 \phi_{xx} + \phi_{xx} = 0 \tag{1.12}$$

$$\phi_z\Big|_{z=\varepsilon\eta} = \mu^2(\eta_t + \varepsilon\phi_x\eta_x) \qquad \phi_z\Big|_{z=-1} = 0 \tag{1.13}$$

$$\{\phi_t + \frac{\varepsilon}{2}(\phi_x^2 + \frac{1}{\mu^2}\phi_z^2)\}_{z=\varepsilon\eta} + \eta = 0 \tag{1.14}$$

then, using separation of variables with $\eta(x,t) = \eta_o e^{i(kx-\omega t)}$, the velocity potential is

$$\phi(x,z,t) = \frac{-\mu i\omega}{k}\eta(x,t)\Big[\sinh(\mu kz) + \coth(\mu k)\cosh(\mu kz)\Big]$$

Utilizing Bernoulli's equation (1.14) yields a dispersion relationship for linear water waves.

$$\omega_{\text{linear}}^2 = \frac{k}{\mu}\tanh(\mu k) \tag{1.15}$$

This dispersion relationship has two limits which depend on the size of μ: shallow water and deep water.

$$\omega_{\text{shallow}}^2 = k^2 \qquad \omega_{\text{deep}}^2 = \frac{k}{\mu} \tag{1.16}$$

Moreover, a dispersion relationship can be found for (1.9) and (1.10). If ∂_x is applied to (1.9) and ∂_t is applied to (1.10), then the linear system can be cast into an equation only in η.

$$\eta_{\text{tt}} - \eta_{\text{xx}} - \frac{\mu^2}{2}\eta_{\text{xxtt}} = 0 \tag{1.17}$$

Assuming the linear wave form for η as before, the dispersion relationship is

$$\omega_{\text{boussinesq}}^2 = \frac{k^2}{1 + \frac{\mu^2}{2}k^2} \tag{1.18}$$

Accordingly, waves with higher wave numbers (shorter wavelengths, $k = \frac{2\pi}{\lambda}$) travel slower than waves with lower wave numbers (longer wavelengths). Here are the phase and group velocities for boussinesq waves.

$$\frac{\omega}{k} = c_o = \frac{1}{\sqrt{1 + \frac{\mu^2}{2}k^2}} \qquad \frac{d\omega}{dk} = c_g = c_o - \frac{\mu^2 k^2}{2\left(1 + \frac{\mu^2}{2}k^2\right)^{3/2}}$$

For $\mu = 0$, $\omega_{\text{boussinesq}}$ reduces to ω_{shallow}. In fact, if (1.18) is expanded in powers of μ around $\mu = 0$ we have,

$$\omega_{\text{boussinesq}} = k\left(1 - \frac{k^2\mu^2}{4} + \frac{3k^4\mu^4}{32} - O(\mu^5)\right)$$

Here one can easily see the effect that μ has on the nature of the solution. As μ is increased from zero, the solution will depart from the ω_{shallow} relationship and start to exhibit wave dispersion. Wave dispersion is one of the two key components of the Boussinesq system (the other being non-linearity). Looking back to (1.9), one can see that μ controls the only dispersive term present in the equations: u_{xxt}. This term is also present in a similar weakly non-linear, dispersive equation described by Benjamin-Bona-Mahony which exhibits similarities to the KdV equation, the classic weakly non-linear, dispersive equation [6].

It is important to examine for what values of μ and k the boussinesq approximation (1.5), is valid. That is to say, the dispersive properties of numerical solutions to (1.9) and (1.10) should approximate $\omega_{\text{boussinesq}}$ but that does not necessarily conclude that these solutions are physical. The dispersive properties of true physical solutions should also approximate ω_{linear}. Fig. 1 has phase velocities (magnitudes squared) as a function of μk. A tolerance for error of 5% is allowed and regions where $c^2_{\text{boussinesq}}$ overlaps with c^2_{linear} are regions where the boussinesq approximation is valid.

It appears that with a tolerance for error of 5%, boussinesq waves are valid for $\mu k < 0.5$ and shallow water waves are valid for $\mu k < 0.3$. Fig. 2 illustrates the available values of k for a given μ. The result is that as the system becomes increasingly more 'shallow', higher wave numbers can be physically represented using the boussinesq equations. For example, if $\mu = 0.05$ then the boussinesq equations can only accurately resolve waves up to $k = 10$. However, if $\mu = 0.01$ then the boussinesq equations can accurately resolve waves up to $k = 50$.

2 Numerical Method

2.1 Projection Method

In order to solve this system of equations, we will start by integrating (1.9) and (1.10) over one time step

$$u^{n+1} - u^n + \int_{t_n}^{t_{n+1}} \left(\frac{\varepsilon}{2}u^2 + \eta\right)_x dt = \frac{\mu^2}{2}\left(u_{xx}^{n+1} - u_{xx}^n\right)$$

$$\eta^{n+1} - \eta^n + \int_{t_n}^{t_{n+1}} (\varepsilon u\eta + u)_x dt = 0$$

which can then be split into hyperbolic and non-hyperbolic parts. This is sometimes referred to as the projection method. Solutions can be found by defining the following two steps:

$$q^* - q^n + \int_{t_n}^{t_{n+1}} f(q)_x \, dt = 0 \tag{2.1}$$

and

$$Tq^{n+1} = q^* - Pq^n \tag{2.2}$$

where

$$q = \begin{pmatrix} u \\ \eta \end{pmatrix} \qquad f(q) = \begin{pmatrix} \frac{\varepsilon}{2}u^2 + \eta \\ \varepsilon u\eta + u \end{pmatrix} \qquad T = \begin{pmatrix} \Theta & 0 \\ 0 & I \end{pmatrix} \qquad P = \begin{pmatrix} \frac{\mu^2}{2}D_+D_- & 0 \\ 0 & 0 \end{pmatrix}$$

$$u = \begin{pmatrix} u_0 \\ u_1 \\ ... \\ u_{N-1} \\ u_N \end{pmatrix} \quad \eta = \begin{pmatrix} \eta_0 \\ \eta_1 \\ ... \\ \eta_{N-1} \\ \eta_N \end{pmatrix} \quad D_+D_-u_i = \frac{S_+ + S_- - 2I}{\Delta x^2}u_i$$

$$S_+u_i = u_{\mathrm{i+1}} \qquad S_-u_i = u_{\mathrm{i\text{-}1}} \qquad Iu_i = u_i$$

$$\Theta = I - \frac{\mu^2}{2}D_+D_- \qquad t_n = n\Delta\mathrm{t} \qquad L = N\Delta\mathrm{x} \qquad x = (j + \frac{1}{2})\Delta x$$

Cell-averaged data is used. Periodic boundary conditions are rolled into the FD operators. The solution to (2.1) is discussed in the next section. Once q^* is found, a relaxation technique can be used to invert T in order to solve for q^{n+1}

2.2 Finite Volume Method

(2.1) can be written in hyperbolic conservative form,

$$q_t + f(q)_x = 0$$

This motivates the use of a Finite Volume Method (FVM), which is founded in dividing the problem's domain into small, locally linear, cell averages. For these cells, the conservation law takes on the linear form

$$q_t + Aq_x = 0 \tag{2.3}$$

where A is the Jacobian matrix. With that said, the solution to q^* can be found by first defining the following averages over space and time

$$q_j^n = \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} q(x,t_n)\, dx \tag{2.4}$$

$$F_{j\pm\frac{1}{2}}^n = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} f(q(x_{j\pm\frac{1}{2}},t))\, dt \tag{2.5}$$

Applying these averages to (2.1) converts it from a continuous conservation form to a discrete one.

$$q_j^* - q_j^n + \frac{\Delta t}{\Delta x}\left(F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n\right) = 0 \tag{2.6}$$

All that is left is to specify the numerical flux functions, $F_{j\pm\frac{1}{2}}^n$. There are a multitude of algorithms that have been developed to handle this problem; probably the simplest of which will now be considered.

2.2.1 Godunov's Method

A detailed account of the REA (recontruct - evolve - average) algorithm introduced by Godunov can be found in [5]. The basic idea starts with reconstructing a piecewise constant function out of cell averages,

$$q(x,t_n) = q_j^n \quad for\ all\ j$$

Doing this permits an exact solution to the integral in (2.5) since $q(x,t_n)$ is obviously constant for $t_n < t < t_{n+1}$. The numerical flux function is thereby equal to the continuous flux function

$$F^n_{j\pm\frac{1}{2}} = f(q(x_{j\pm\frac{1}{2}},t_n))$$

Moreover, all of this analysis pertains to locally linear, cell averages. Therefore the non-linear, continuous flux function must be replaced by the linear form identified in (2.3).

$$F^n_{j\pm\frac{1}{2}} = f(q(x_{j\pm\frac{1}{2}},t_n)) = Aq(x_{j\pm\frac{1}{2}},t_n)$$

Since hyperbolic problems transmit information with finite speed, the flux through a cell edge (at $x_{j\pm\frac{1}{2}}$) can be defined as a function of the two adjacent cells (one cell to the left and one to the right) [5].

$$F^n_{j-\frac{1}{2}} = F(q^n_{j-1},q^n_j) = F_-(q^n_L,q^n_R)$$

Then (2.6) becomes,

$$q^*_j - q^n_j + \frac{\Delta t}{\Delta x}\left[F_+(q^n_L,q^n_R) - F_-(q^n_L,q^n_R)\right] = 0 \tag{2.7}$$

Of course, the value of q that is used in the flux function is located at the cell edge, which is not q_L or q_R. A method must be developed, which will use q_L and q_R, that will return the value of q at the cell edge. Let $\tilde{q}$ be the value of q at the cell edge. Then,

$$\tilde{q}^n = \tilde{q}(q^n_L,q^n_R)$$

and

$$q^*_j - q^n_j + \frac{\Delta t}{\Delta x}\left[F_+(\tilde{q}^n) - F_-(\tilde{q}^n)\right] = 0 \tag{2.8}$$

Simply averaging q_L and q_R is not satisfactory. Since (2.7) is a completely general result, consider what happens to the advection equation if the averaging method is used,

$$q_t + aq_x = 0 \quad f(q) = aq$$

$$q_j^* - q_j^n + \frac{a\Delta t}{\Delta x}\left((q_{j+1}^n - q_j^n)/2 - (q_j^n - q_{j-1}^n)/2\right) = 0$$

$$q_j^* - q_j^n + \frac{a\Delta t}{2\Delta x}\left(q_{j+1}^n - q_{j-1}^n\right) = 0 \tag{2.9}$$

This is analogous to an explicit Euler forward (in time) / centered (in space) FD scheme, which is unstable [5]. Therefore, a different method for finding $\tilde{q}^n$ must be explored. This is presented in the next section.

After $\tilde{q}^n$ is found, the next stage of the algorithm is to evolve the hyperbolic equation. This is done by applying (2.8) in order to obtain q_j^*. The final stage of the method, average, takes place after the rest of the projection method is completed and $q(x,t_{n+1})$ is obtained. Use this result to re-calculate the cell averages for the next iteration.

2.2.2 Riemann Solutions

The first step of the REA algorithm involved reconstructing a piecewise constant function out of cell averages. This places a discontinuity at each cell interface. Determining what happens to this discontinuity as the underlying equations evolve is known as a Riemann problem. The theory of Riemann problems is covered, in detail, in many published works (see [5]). To re-create the discussion here would be detracting, cumbersome, and most likely incomplete. Instead, this section will focus on the algebraic steps necessary to generate solutions to Riemann problems, all with respect to the Boussinesq system. Start with the Jacobian matrix, A, as defined in (2.3)

$$A = \begin{pmatrix} \varepsilon\bar{u} & 1 \\ \varepsilon\bar{\eta} + 1 & \varepsilon\bar{u} \end{pmatrix} \tag{2.10}$$

where the bar over the value represents the average of q_L^n and q_R^n. Now if A is diagonalizable and non-defective then it can be decomposed into a product of e'vector/e'value matrices.

$$A = R\Lambda R^{-1}$$

$$R = \begin{pmatrix} \frac{-1}{\sqrt{1+\varepsilon\bar{\eta}}} & \frac{1}{\sqrt{1+\varepsilon\bar{\eta}}} \\ 1 & 1 \end{pmatrix} \qquad \Lambda = \begin{pmatrix} \varepsilon\bar{u} - \sqrt{1+\varepsilon\bar{\eta}} & 0 \\ 0 & \varepsilon\bar{u} + \sqrt{1+\varepsilon\bar{\eta}} \end{pmatrix}$$

Define $z = R^{-1}q$, then (2.3) becomes

$$z_t + \Lambda z_x = 0$$

which is simple advection for each row. Further transformation of interest include,

$$z_L^n = R^{-1} q_L^n \qquad z_R^n = R^{-1} q_R^n \qquad \tilde{q}^n = R\tilde{z}^n$$

The components of $\tilde{z}^n$ depend on the sign of the e'values in Λ. If the first e'value is negative, the solution is traveling to the left and the first component of $\tilde{z}^n$ is equal to the first component of z_R^n. If the second e'value is positive, the solution is traveling to the right and the second component of $\tilde{z}^n$ is equal to the second component of z_L^n. Once $\tilde{z}^n$ is formed, transform back to q-space.

2.2.3 Lax - Wendroff Correction

Due to the use of the aforementioned piecewise constant function, Godunov's method is, theoretically , O(Δx,Δt) accurate. In order to improve the theoretical accuracy to O(Δx^2,Δt^2), the Lax-Wendroff method will be utilized. However, this presents another problem: Near a discontinuity, Lax-Wendroff produces spurious oscillations (dispersion) as opposed to Godunov's method which smooths out discontinuities (diffusion). In order to correct this problem, a flux - limiting technique is applied when near a discontinuity.

The Lax - Wendroff Method involves modifying the numerical flux function in order to cancel out the lowest order of truncation error (in time) with another term (in space). This will bump the method to O(Δx, Δt^2). Start by implementing Euler forward (in time) for (2.3),

$$\frac{q^{n+1}-q^n}{\Delta t} + Aq_x \quad = \quad \frac{q(t+\Delta t)-q(t)}{\Delta t} + Aq_x \quad = \quad 0$$

$$\frac{q(t)+\Delta t q_t + \frac{(\Delta t)^2}{2} q_{tt} + \frac{(\Delta t)^3}{6} q_{ttt} + O(\Delta t^3) - q(t)}{\Delta t} + Aq_x = 0$$

$$\underbrace{q_t + Aq_x + \frac{\Delta t}{2} q_{tt} + \frac{(\Delta t)^2}{6} q_{ttt} + O(\Delta t^3)}_{numerical} - \underbrace{q_t + Aq_x}_{exact} = \underbrace{\frac{\Delta t}{2} q_{tt} + \frac{(\Delta t)^2}{6} q_{ttt} + O(\Delta t^3)}_{truncation\ error}$$

where 'numerical' refers to the equation one is actually solving if Euler forward (in time) is implemented and 'exact' refers to the equation before discretization. Therefore, the lowest order error term resulting from the time discretization of (2.3) is O(Δt). If this term was subtracted from the original equation, it would cancel out this error. Start by rewriting q_{tt} in terms of spatial variables using (2.3),

$$q_{tt} = -Aq_{xt} \quad , \quad q_{tx} = -Aq_{xx} \quad \longrightarrow \quad q_{tt} = A \cdot Aq_{xx}$$

then subtract this term from (2.3) to get the modified equation.

$$q_t + Aq_x - \frac{\Delta t}{2} A \cdot Aq_{xx} = 0 \tag{2.11}$$

Now all that is left is to cast this equation into the numerical flux form, (2.7), in order to extract the new numerical flux functions. According to the analysis resulting in (2.9), there is a close relationship between FD and FVM methods. If (2.11) is to be solved using stable O(Δx^2) FD techniques, the spatial discretization would be

$$q_{jt} + A\left(\frac{q_{j+1} - q_{j-1}}{2\Delta x}\right) - \frac{\Delta t}{2} A \cdot A\left(\frac{q_{j+1} + q_{j-1} - 2q_j}{(\Delta x)^2}\right) = 0 \tag{2.12}$$

After grouping appropriate terms, we have

$$F^n_{j+\frac{1}{2}} = \frac{A}{2}(q_j + q_{j+1}) - \frac{\Delta t}{2\Delta x} A \cdot A(q_{j+1} - q_j)$$

$$F^n_{j-\frac{1}{2}} = \frac{A}{2}(q_{j-1} + q_j) - \frac{\Delta t}{2\Delta x} A \cdot A(q_j - q_{j-1})$$

or in (L,R) notation,

$$F^{LW}_{\pm}(q^n_L, q^n_R) = \frac{A}{2}(q^n_L + q^n_R) - \frac{\Delta t}{2\Delta x} A \cdot A(q^n_R - q^n_L)$$

As mentioned above, solely using F^{LW} will produce a O(Δx^2,Δt^2) method, but at the expense of high truncation error (in this case, spurious oscillations) near discontinuities. In order to correct this problem, a flux - limiter, $\Theta(r)$,will be used in order to change the numerical flux function back to O(Δx,Δt) near a discontinuity. Denote the flux function resulting from Godunov's method as F^{low} and F^{LW} as F^{high}. Then,

$$F_{\pm}(q^n_L, q^n_R) = F^{low}_{\pm} + \Theta(r)(F^{high}_{\pm} - F^{low}_{\pm})$$

There are many forms of $\Theta(r)$. These solutions appear to respond well to the monotonized central (MC) limiter.

$$\Theta(r_i) = max[0, min(2r_i, \frac{1 + r_i}{2}, 2)]; \quad \lim_{r_i \to \infty} \Theta(r_i) = 2$$

where r_i is either r_L or r_R. r_L is the ratio of relevant slopes for the left traveling wave and r_R is the ratio of relevant slopes for the right traveling wave (q_{RR} is to the right of q_R and q_{LL} is to the left of q_L)

$$r_L = \frac{q_{RR} - q_R}{q_R - q_L} \qquad r_R = \frac{q_L - q_{LL}}{q_R - q_L}$$

$$\Theta(r) = min[\Theta(r_L), \Theta(r_R)]$$

3 Solutions

3.1 Limiting Cases

A great way to start analyzing these solutions is to consider what happens if μ and ε are varied. As mentioned in the introduction, ε can be tuned down to approximate the linear Boussinesq equations. Snapshots of this for $\varepsilon = 0.001$ and $\mu = 0.01$ can be found in Fig. 3. The left traveling wave has been manually removed so that we can see how the singular waveform evolves over time. Notice the wave dispersion which is described by $\omega_{\text{Boussinesq}}$. Also mentioned in the introduction, μ can be tuned down to approximate the nonlinear shallow water equations. Snapshots of this for $\mu = 0.001$ and $\varepsilon = 1$ can be found in Fig. 4. Here no extra steps were taken to remove the left traveling wave nor do the wave forms travel through the periodic boundaries. Notice, in Fig. 4, how the initial gaussian bump quickly develops shocks due to the non-linear terms. The success of the FVM can be seen in how well these shocks are propagated. As a final example of limiting cases, consider tuning down both μ and ε, as in Fig. 5. Here we recover the linearized shallow water equation which is the linear wave equation.

3.2 Solitons

The full nonlinear, dispersive equations emit solutions which take on a solitary wave form, or soliton. The word "soliton" is an amalgamation of solitary wave and particle-like words such as electron, photon, or phonon. This name suits it well since the soliton has both wave-like and particle-like properties including a localized, permanent form as well as the ability to interact with other solitons and emerge from a collision unchanged. One of the key characteristics about solitons is that their existence is due to a balance in non-linear and dispersive terms [1]. The non-linear terms continually steepen the waveform (as seen in Fig. 4) while the dispersive terms do the opposite (as seen in Fig. 3). Fig. 6 through Fig. 9 are great examples of this balance.

Fig. 6 contains snapshots of the flux-limited solutions for a smooth initial gaussian. Fig. 7 contains snapshots of the same solution with the left-traveling wave (crudely) removed. Here the z-scale was changed and the initial condition is not shown. It was difficult to manually remove the left traveling wave without affecting the right traveling wave because the nonlinear solutions are more coupled than their linear counterparts. Nevertheless, the point is clear: Solitons are high stable waveforms despite the size of the dispersive and nonlinear terms.

Solitary wave formation also occurs in Fig. 8, where an initial square wave is used. Although the square wave initially contains Fourier components that are too high to be considered "shallow and long waves", the hyper-diffusive truncation error (discussed in the next section) quickly eliminates those components. You can also see dispersive truncation error in this solution.

Analytic solutions to (1.8) have been found by S. Ding and X. Zhao [4]. From [4],

$$\eta(x,t) = \frac{\omega^2 - c_o^2 k^2 - 8c\alpha k^4}{2k^2\beta} + \frac{6k^2\alpha c}{\beta}\tanh^2[\sqrt{-c}(kx+\omega t)]$$

then using (1.11), $\tilde{k} = k\sqrt{-c}$, and $\frac{\omega^2}{k^2} = c_v^2 = c_o^2 + 4\alpha\tilde{k}^2$, we have

$$\eta(x,t) = \frac{6\mu^2\tilde{k}^2}{\varepsilon}\left(1 - \tanh^2[\tilde{k}(x+c_v t)]\right) \tag{3.1}$$

Now, if this function was taken at $t = 0$ then we would have half of the initial conditions necessary to implement this solution numerically. For the $u(x,t)$ initial condition, transform coordinates to a reference frame moving with speed c_v. In this frame $\eta_t = 0$ and (1.10) becomes $u(\varepsilon\eta + 1) = Q$, which is constant. If we move back to a stationary frame we have,

$$u(x,t) = \frac{Q}{\varepsilon\eta + 1} + c_v$$

where $Q = -c_v$ if $\eta(x,t)$ and $u(x,t)$ go to zero at infinity, which it does. Evaluating $u(x,t)$ at t=0 provides the initial conditions necessary to implement this solution. Snapshots of this single soliton can be found in Fig. 9. This is a wonderful result because it shows that the numerical method implemented in this paper is capable of reproducing solitons which can be described analytically, if the initial conditions are carefully chosen. This solution takes on the form of the $sech^2$ soliton presented in full detail in [1]. Specifically, the author's argument hinges on $\mu << 1$ and the wave amplitude being small. As you can see from (3.1), those requirements are one in the same. In fact, in order for the FVM to produce this soliton, μ was lowered to 0.005. From [1],

$$\eta \approx \eta_0 sech^2[\sqrt{\frac{3\eta_0}{4K}}(x - (1 + \frac{1}{2}\varepsilon\eta_0)t)]$$

we see that the speed of the wave $(1 + \frac{1}{2}\varepsilon\eta_0)$ increases as η_0 increases. Furthermore the width of the wave is inversely proportional to $\sqrt{\eta_0}$. Both of these phenomena can be observed in Figs. 6 through 8.

4 Accuracy

As mentioned above, the FVM consists of a $O(\Delta x^2, \Delta t^2)$ technique wherever the solution is well behaved and drops to a $O(\Delta x, \Delta t)$ technique near a discontinuity. Using the Lax-Wendroff technique, we were able to cancel out the lowest order error term which was $O(\Delta t)$, bumping the method to $O(\Delta t^2)$. It has not, however, been discussed so far how the spatial discretization that resulted in the Lax-Wendroff flux functions is $O(\Delta x^2)$. Revisiting (2.12),

$$q_{jt} + A\left(\frac{q_{j+1} - q_{j-1}}{2\Delta x}\right) - \frac{\Delta t}{2} A \cdot A\left(\frac{q_{j+1} + q_{j-1} - 2q_j}{(\Delta x)^2}\right) = 0$$

the FD operators and their symbols are

$$D_+ D_- q_i = \frac{S_+ + S_- - 2I}{\Delta x^2} q_i \quad \xrightarrow{FT} \quad D_+ D_- \hat{q}_i = \frac{2\cos(k\Delta x) - 2}{\Delta x^2} \hat{q}_i$$

$$D_0 q_i = \frac{S_+ - S_-}{2\Delta x} q_i \quad \xrightarrow{FT} \quad D_0 \hat{q}_i = \frac{i\sin(k\Delta x)}{\Delta x} \hat{q}_i$$

It is important to note that the above are eigenvalue equations. Expanding the discrete eigenvalues in a taylor series (keeping only the lowest order), one can see how they differ from the continuous eigenvalues.

$$\underbrace{-k^2(1 - \frac{(k\Delta x)^2}{12})}_{discrete} - \underbrace{-k^2}_{continuous} = \underbrace{\frac{\Delta x^2}{12} k^4}_{T.E.} \tag{4.1}$$

$$\underbrace{ik(1 - \frac{(k\Delta x)^2}{6}}_{discrete} - \underbrace{ik}_{continuous} = \underbrace{-\frac{\Delta x^2}{6} ik^3}_{T.E.} \tag{4.2}$$

Therefore all of the operators used in this numerical method are $O(\Delta x^2)$. As long as $k << N$ the discrete spectrum will closely approximate the continuous spectrum. For large k, the RHS of (4.1) and (4.2) will not be zero. This is known as truncation error (T.E.). Taking the inverse Fourier transform, these 'truncation equations' become

$$\underbrace{D_+ D_-}_{discrete} - \underbrace{\partial_{xx}}_{continuous} = \underbrace{\frac{\Delta x^2}{12} \partial_{xxxx}}_{T.E.}$$

$$\underbrace{D_0}_{discrete} - \underbrace{\partial_x}_{continuous} = \underbrace{\frac{\Delta x^2}{6} \partial_{xxx}}_{T.E.}$$

This supports the visual conclusion of dispersive and hyper-diffusive truncation error found in the sharp solution. In order to examine this error, we will revisit the discontinuous (sharp) solution by first turning off the flux limiter and then running the solution at $N = 256$ and $N = 1024$. By turning off the limiter, we are ensuring that truncation error will quickly build up near the discontinuities. The error should be more prevalent in the $N = 256$ solution. Fig. 10 and Fig. 11 contain the $N = 256$ and $N = 1024$ solutions, respectively. As you can see, hyper-diffusive error takes longer to build up in Fig. 11, as expected. Furthermore, Fig. 10 and Fig. 11 contain significantly more wave dispersion than the flux-limited version (Fig. 8) which was also expected.

Using the analytic solutions found by S. Ding and X. Zhao [4] discussed earlier, rate of convergence can be computed. With the limiter off, these solutions should approach O(Δx^2). In order to calculate the rate of convergence, let

$$E(\Delta x) = \|\eta^{analytic}(x,T) - \eta^{\text{numeric}}(x,T)\|_2 = \left(\int_0^L \left| \eta^{analytic}(x,T) - \eta^{\text{numeric}}(x,T) \right|^2 dx \right)^{1/2}$$
$$= \left(\Delta \text{x} \sum_{i=0}^{N} |\eta^{analytic}(x_i,T) - \eta^{\text{numeric}}(x_i,T)|^2 \right)^{1/2}$$

then plot $E(\Delta x)$ versus Δx on a log-scale (base 2). The slope of this plot is the rate of convergence. These results are summarized in the following table.

log(Δx)	log(E(Δx))	r
-4	-10.33	
-5	-12.23	1.90
-6	-14.29	2.06
-7	-16.36	2.06

Using (3.1) for $\eta^{analytic}$, we calculate $r = 2.06$, or O($\Delta x^{2.06}$). This means that when the limiter is off, the solution is behaving close to its theoretical capacity. Of course, this worked out well because the analytic solution does not contain any discontinuities. If discontinuities were present, the limiter would need to be used and the actual rate of convergence would be less than r=2.06.

Appendix A: Free Surface Boundary Condition

Let G = z - η(x,t) = 0. Then $\frac{DG}{Dt} = G_t + (u \cdot \nabla)G = 0.$

$$\frac{DG}{Dt} = -\eta_t - u\eta_x + w = 0$$

Then using incompressibility,

$$u_x + w_z = 0 \quad w = -\int_{-h}^{\eta} u_x \, dz$$

Substituting this into the first equation,

$$\eta_t + u\eta_x + \int_{-h}^{\eta} u_x \, dz = 0$$

Leibniz integral rule states that,

$$\partial_x \int_{-h}^{\eta} u \, dz = \int_{-h}^{\eta} u_x \, dz + u\eta_x$$

The result of which is the free surface boundary condition,

$$\eta_t + \partial_x \int_{-h}^{\eta} u \, dz = 0$$

Appendix B: PDELAB Source Code

Throughout this paper numerical methods are explained in some detail, however the source code used to generate these solutions has not be included. PDELAB is the project that has grown organically out of the need to analysis and solve the boussinesq equations. It is a purely object-oriented, command-line driven program whose goal is to provides a common set of tools/commands for whatever PDE solution is implemented under its framework. It is primarily written in C++, however its most recent development includes Python modules which automate much of the required tasks and analysis. A PDE solution is implemented as a subclass of the *Example* abstract class. This provides a common set of functions/methods which can then be called by a subclass of the *Solver* abstract class (specifically *SimSolver* for these PDEs) which will actually solve the equations, record the results, and add additional instructions (if any). For a full explanation of the PDELAB project, as well as source downloads and quick start instructions, visit `http://code.google.com/p/pdelab`.

Figures

The following ten pages contain all of the Figures referenced in this thesis. It is difficult to truly capture the essence of wave solutions as they travel through time in still images. Some authors prefer to plot time on a second axis (if the system they are analyzing is one dimensional), however there are trade-offs to this approach. If this is done, the solution has to be viewed from an awkward angle (usually at 45° above the x-t plane and at 45° off the t-axis). Doing this usually prevents the reader from observing any part of the solution below the x-t plane and it can also distort the reader's perception of the propagating wave. This is precisely why I have chosen to display all of my solutions from a viewpoint that is on axis and in the plane. The cost if this approach is that time will be difficult to capture. I must appeal to the reader's imagination by asking that they interpolate the sequence of snapshots that are presented for each solution. A standard approach will be implemented in order to aid with this interpolation: All of the solution snapshots will start in the top left corner and evolve from left to right and then down the page.

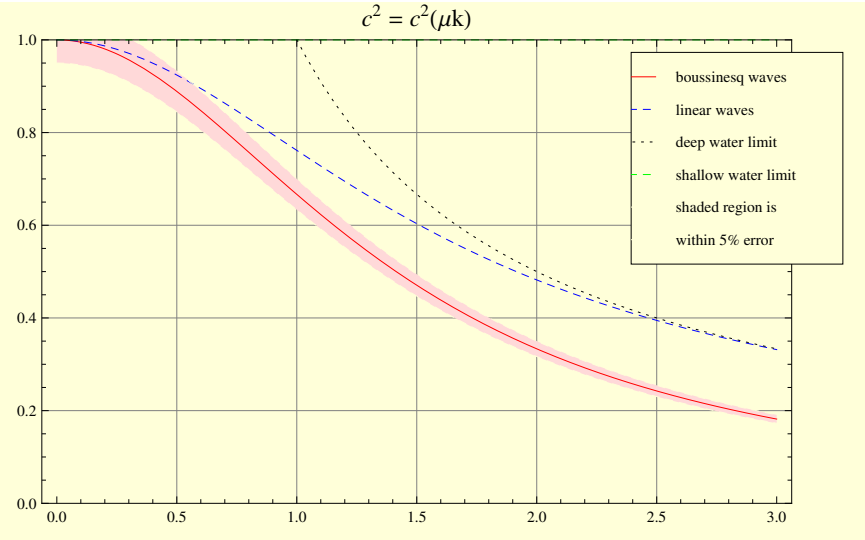

Figure 1: Phase velocities (magnitudes squared) as a function of μk. The shallow water limit runs across the top of the graph ($c_{\text{shallow}} = 1$)

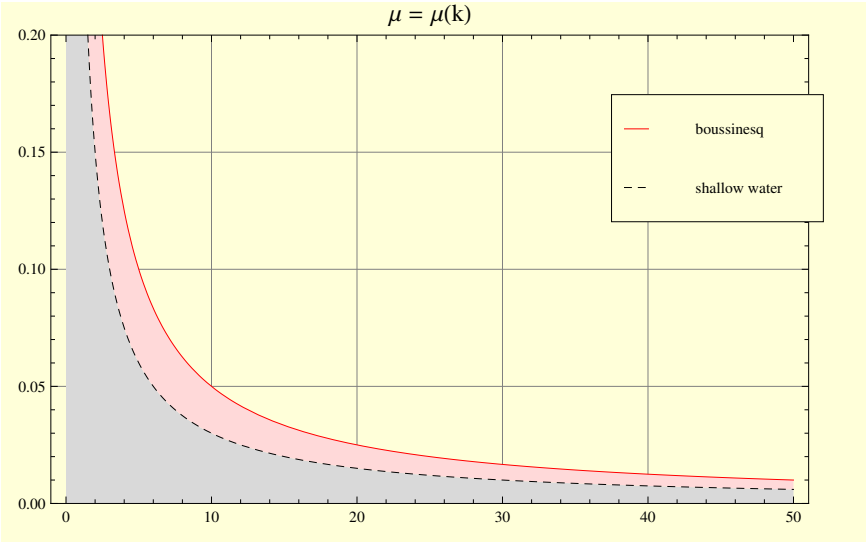

Figure 2: Limiting values for μ and k.

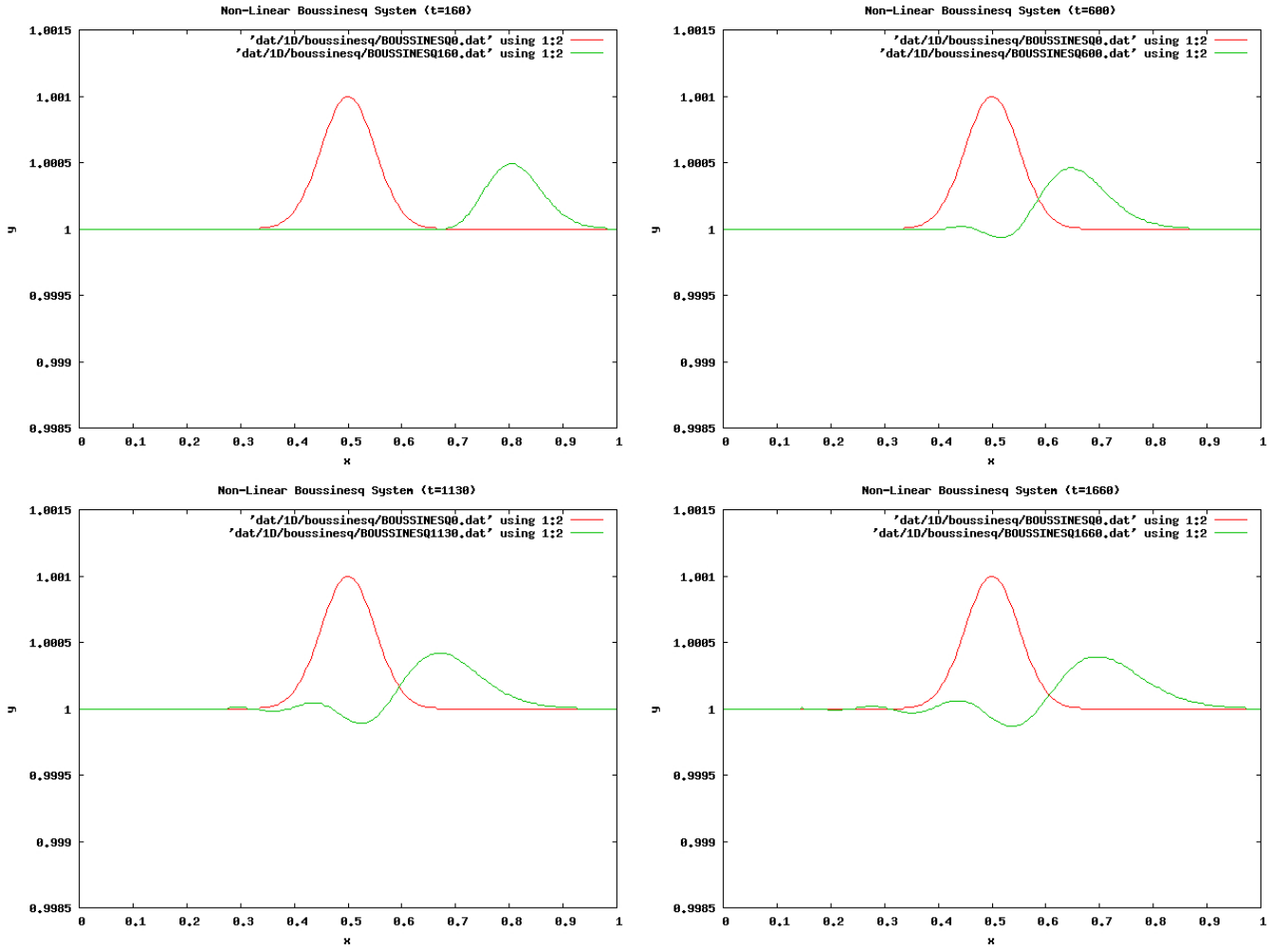

Figure 3: Initial gaussian wave with zero velocity. $\mu = 0.01$, $\varepsilon = 0.001$, L = 1, N = 256. cfl# = 0.1, $\sigma = 0.05$. Linear Boussinesq. In image 1, the left traveling wave has just been manually removed. In image 2, 3, and 4, the right traveling wave has wrapped around the domain once, twice, and three times, respectively.

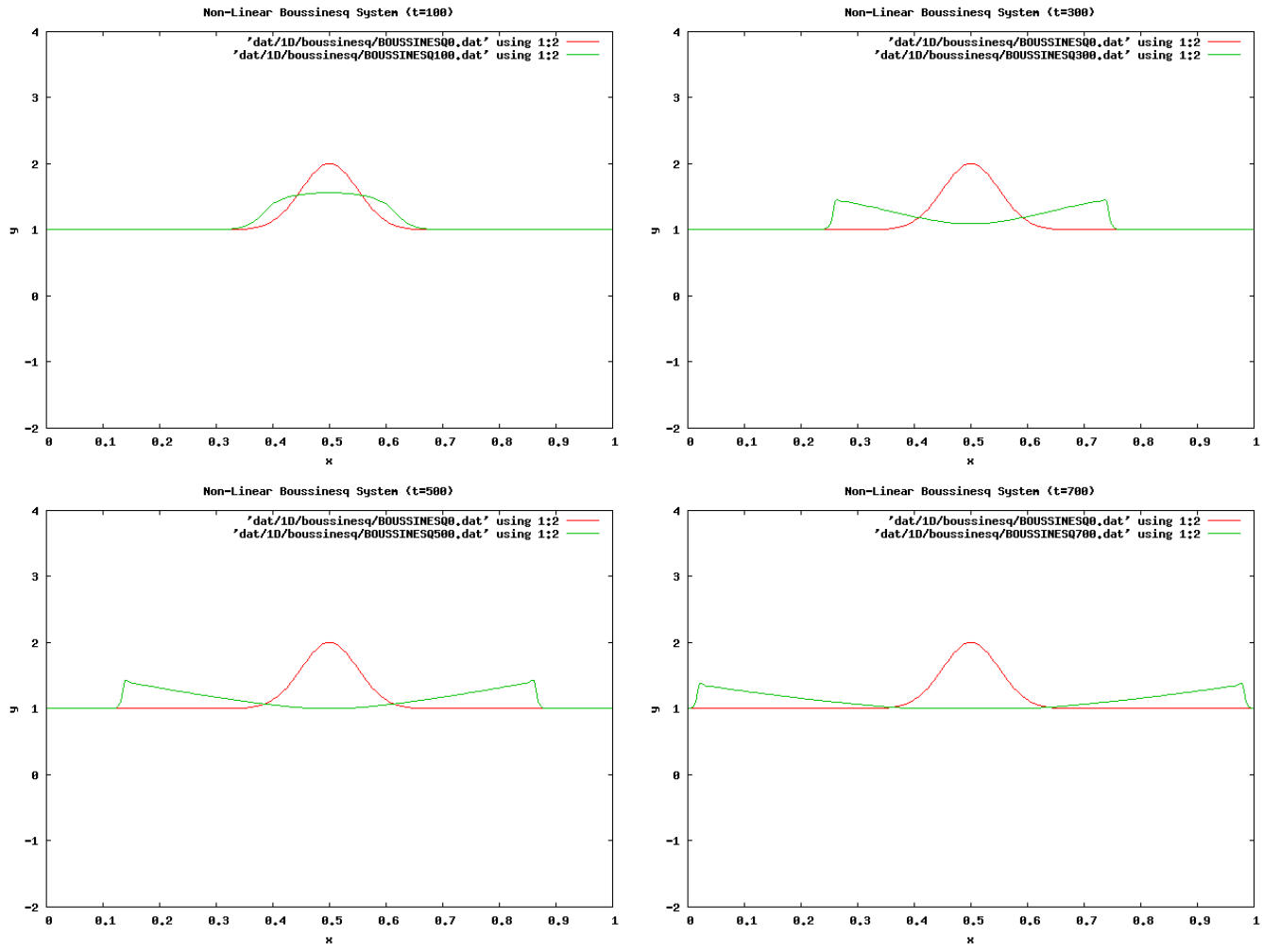

Figure 4: Initial gaussian wave with zero velocity. $\mu = 0.001$, $\varepsilon = 1$, L = 1, N = 256. cfl# = 0.1, $\sigma = 0.05$. Nonlinear shallow water.

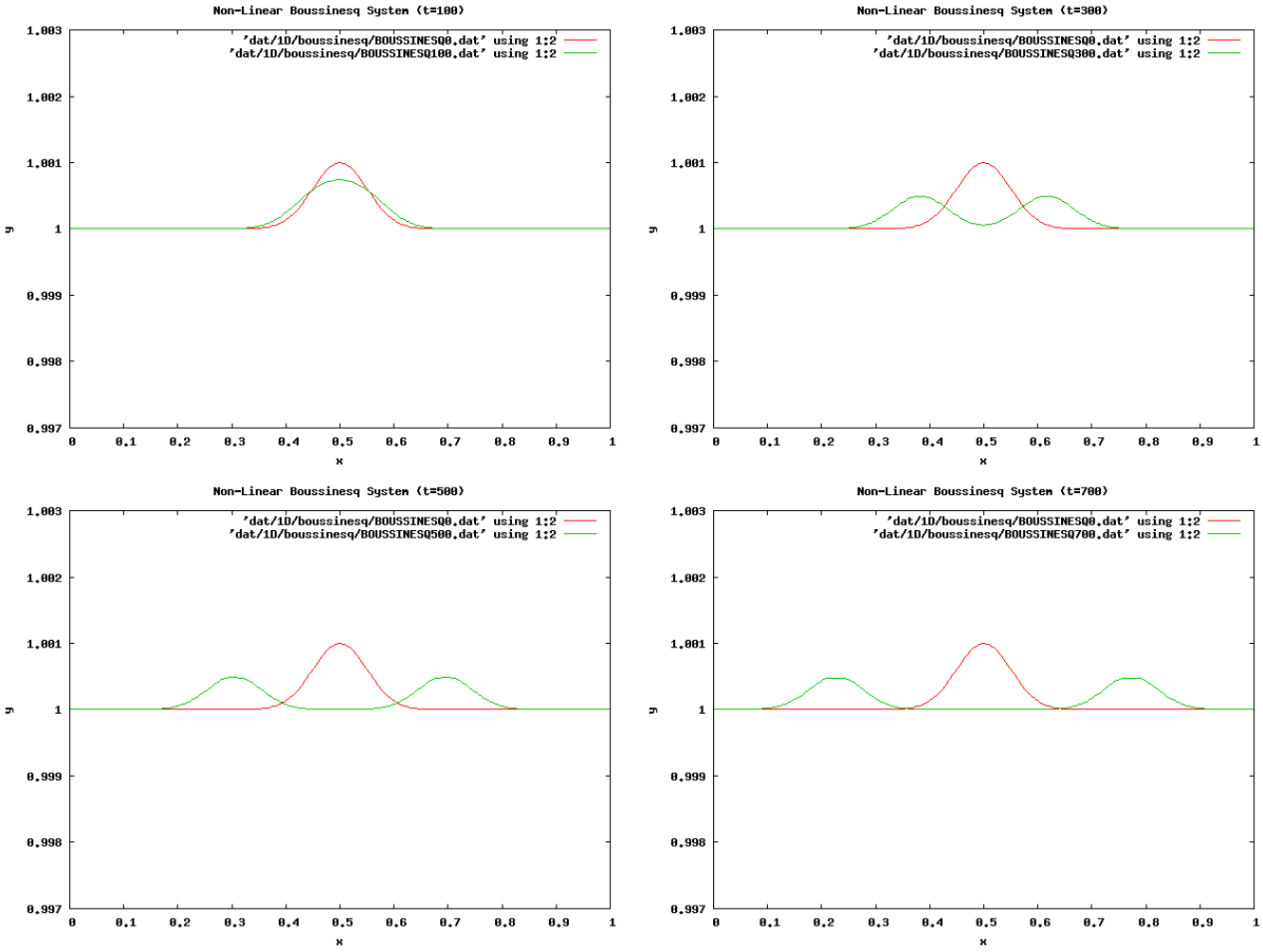

Figure 5: Initial gaussian wave with zero velocity. $\mu = 0.001$, $\varepsilon = 0.001$, L = 1, N = 256. cfl# = 0.1. Linearized shallow water (wave equation).

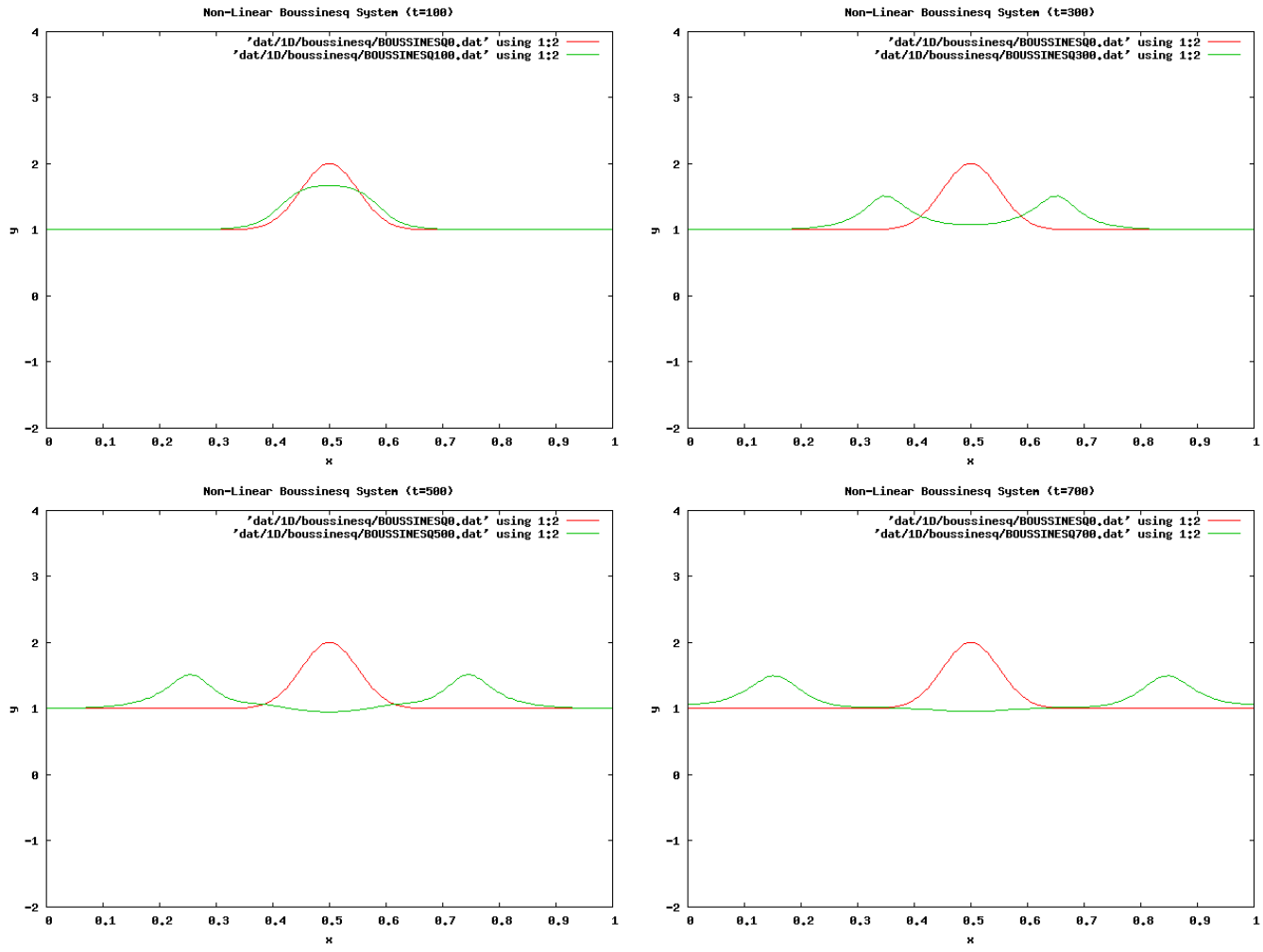

Figure 6: Initial gaussian wave with zero velocity. $\mu = 0.05$, $\varepsilon = 1$, L = 1, N = 256. cfl# = 0.1, $\sigma = 0.05$. Nonlinear Boussinesq

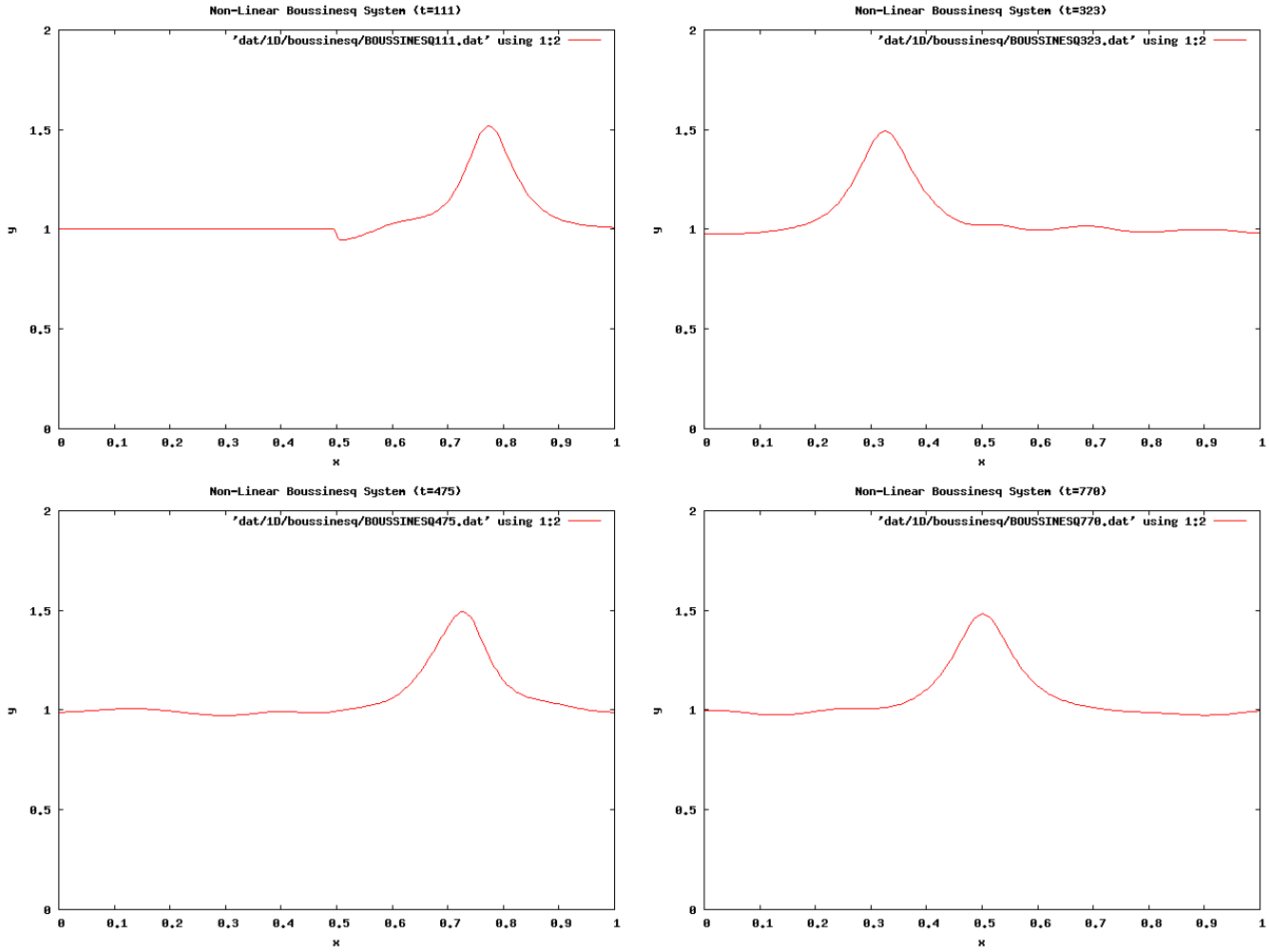

Figure 7: Same initial conditions as in Fig. 6. In image 1, the left traveling wave has just been (crudely) removed, the initial condition is not displayed, and the z-scale has been changed. In image 2 and 3, the wave has passed through the periodic boundary once. In image 4 it has passed through twice.

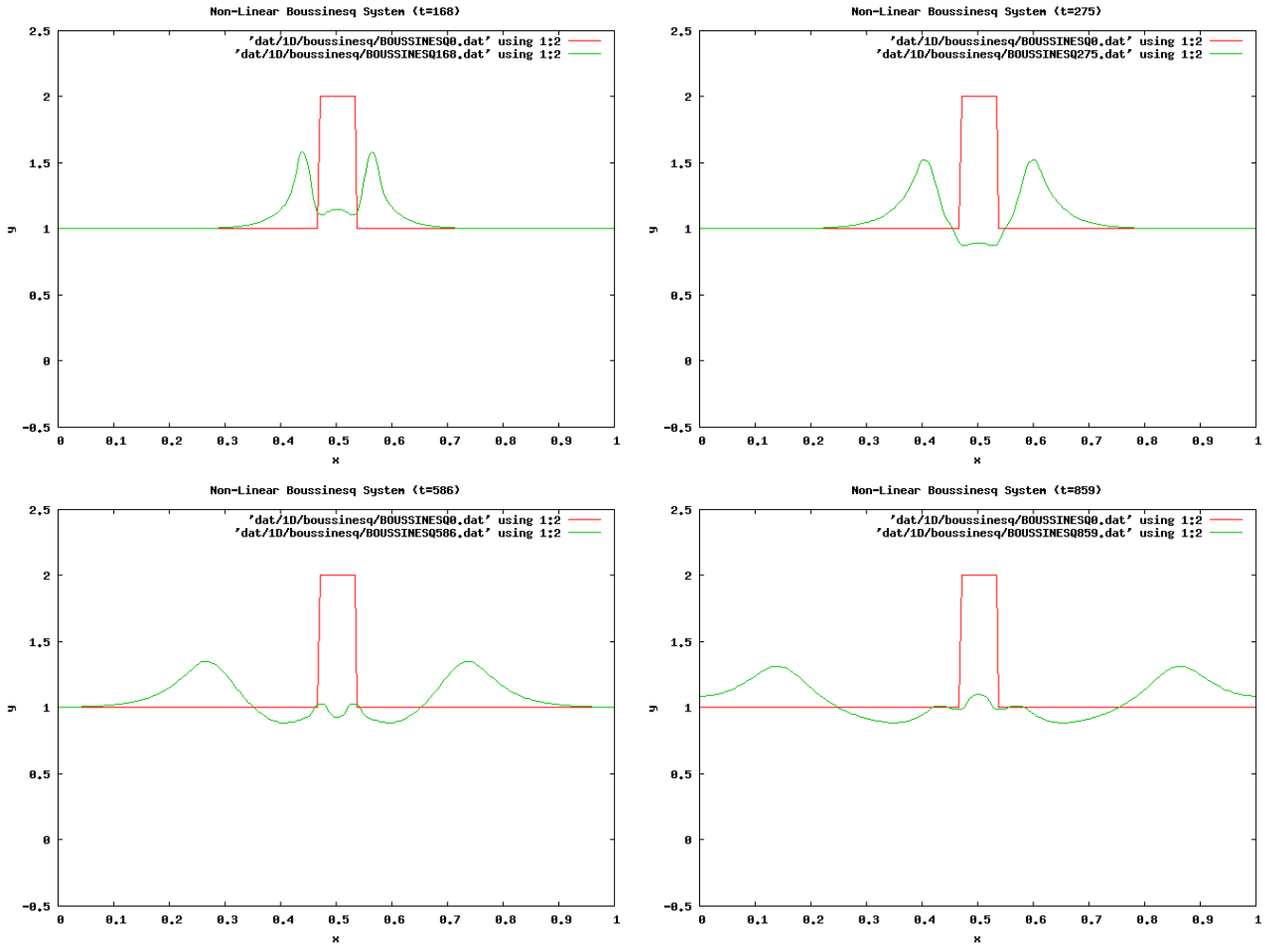

Figure 8: Initial square wave with zero velocity. $\mu = 0.05$, $\varepsilon = 1$, L = 1, N = 256. cfl# = 0.1. Nonlinear Boussinesq

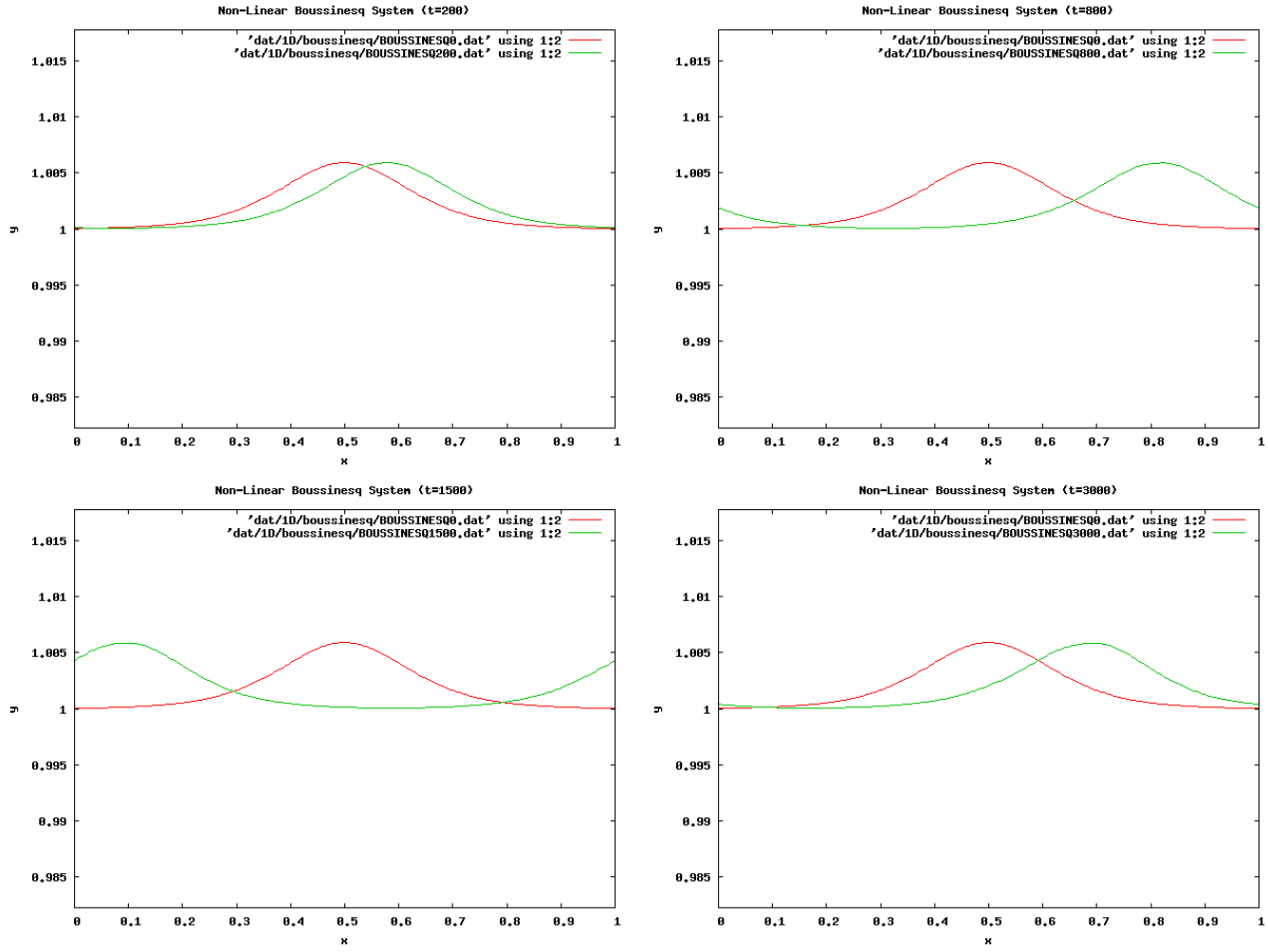

Figure 9: (3.1) initial conditions. $\mu = 0.005$, $\varepsilon = 1$, L = 1, N = 256. cfl# = 0.1. Nonlinear Boussinesq. This is a right traveling wave with periodic boundary conditions. From image 1 to image 4 the wave passes through the right boundary once.

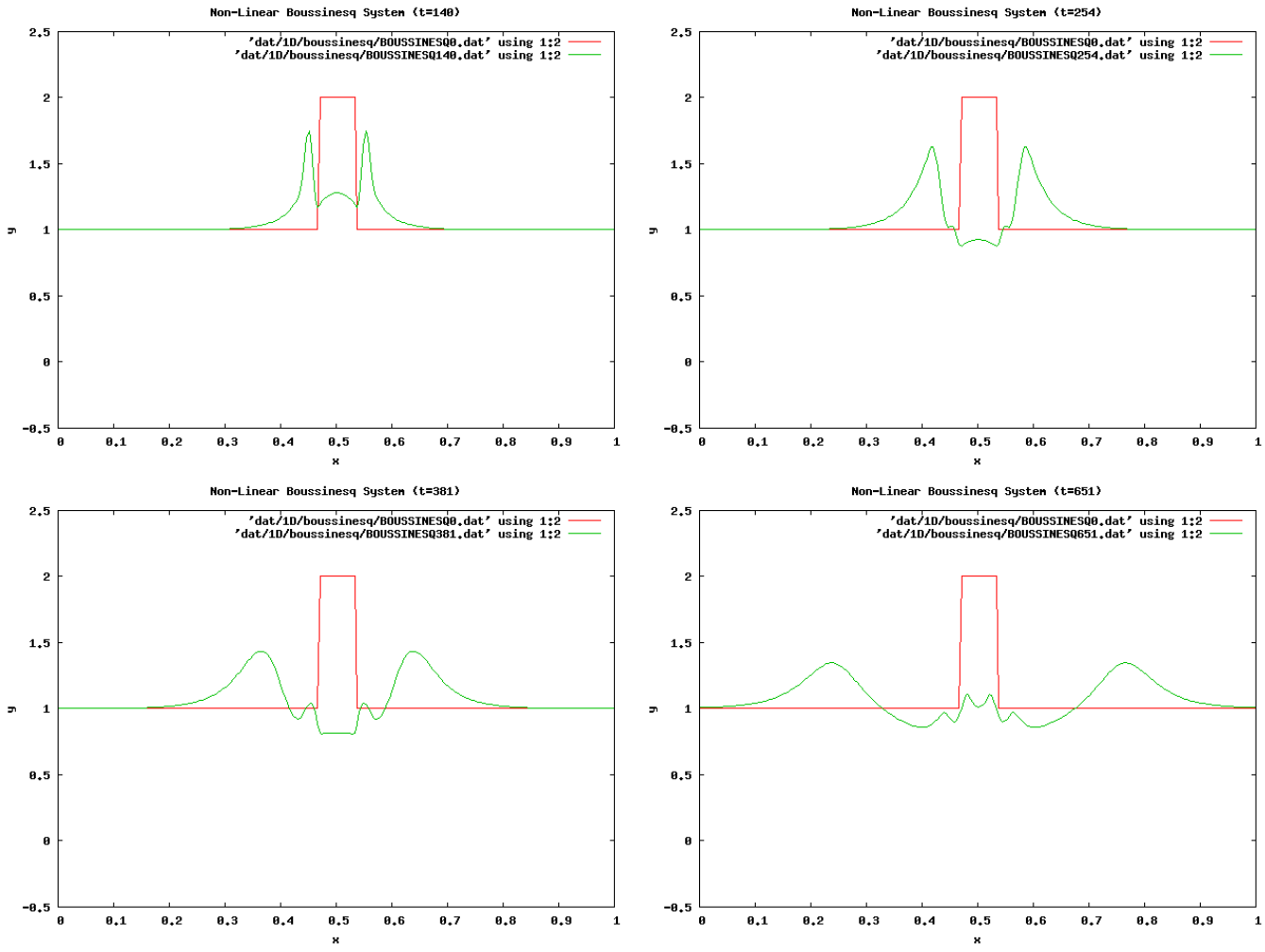

Figure 10: Initial square wave with zero velocity. μ = 0.05, ε = 1, L = 1, N = 256. cfl# = 0.1. Nonlinear Boussinesq, flux-limiter off.

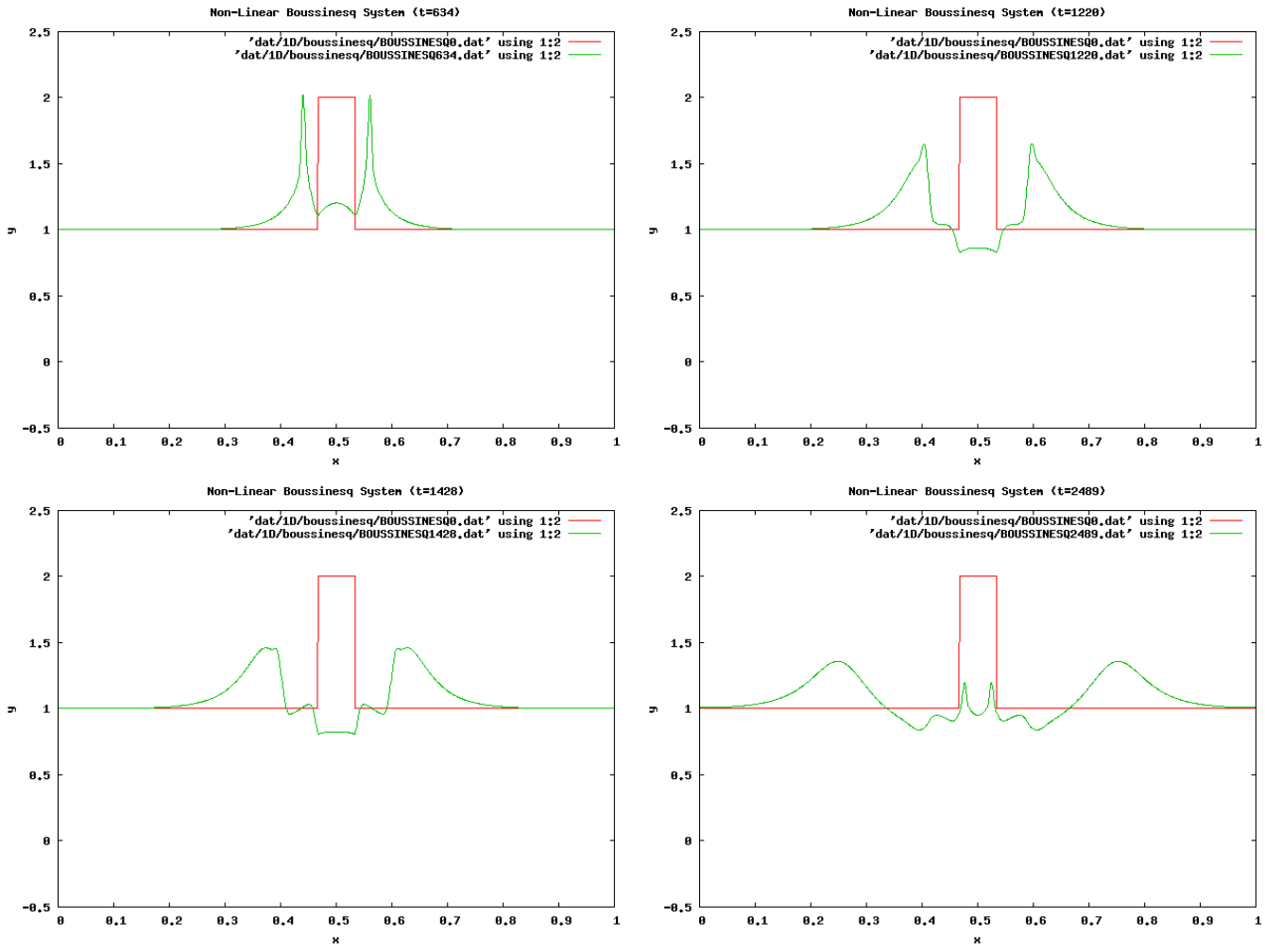

Figure 11: Initial square wave with zero velocity. μ = 0.05, ε = 1, L = 1, N = 1024. cfl# = 0.1. Nonlinear Boussinesq, flux-limiter off.

References

[1] Johnson, R.S. (1997), A Modern Introduction to the Mathematical Theory of Water Waves, Cambridge University Press, Cambridge, UK, p. 165 - 176.

[2] Ostrovsky, Lev A. and Potapov, Alexander S. (1999), Modulated Waves, Theory and Applications, Johns Hopkins University Press, Baltimore, MD, p. 212 - 214: 222.

[3] Durran, Dale R. (1999), Numerical Methods for Wave Equations in Geophysical Fluid Dynamics, Springer-Verlag, New York, NY, p. 68-69.

[4] Ding, S. and Zhao, XQ. (2006), "Exact traveling wave solutions of the Boussinesq equation",Chaos, Solitons, and Fractals, 29 1032 - 1036.

[5] Leveque, Randall J. (2004), Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, Cambridge, UK, p. 67-82:47-62.

[6] Benjamin, T.B., Bona J.L., and Mahony J.J., (1972), "Model Equations for Long Waves in Nonlinear, Dispersive Systems", Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences 272 (1220): 47-48