## Coltin James Lux

## Mathematics Capstone

## 1 December 2016

Tsunami Wave Modeling 101:

Problem:

Tsunamis (seismic sea waves) are characterized by very large, tall, and powerful waves that are created by an underwater disturbance creating a large amount of energy. An underwater disturbance such as an earthquake, landslide, volcanic eruption, or meteorite can release very large amounts of energy underwater. Tsunamis can be known to travel hundreds of miles per hour and get as tall as a hundred feet or more. Typical tsunami waves will travel in all directions from the occurrence of the underwater disturbance. However, when the wave approaches shore, when the ocean floor rises in height, the wave builds in height due to the amount of energy.

All tsunamis can be dangerous. There have been several destructive tsunamis throughout history and the more we can prepare for such natural disasters the better off and safer we will be. That is, the more tools we have for modeling the behavior of these waves, the more we can prepare for the waves crashing into shore. Thus, in this written work we will be modeling tsunami wave behavior.

To model tsunami waves, we will be using a variation of finite difference methods for two commonly used partial differential equations commonly seen in shallow water wave theory for modeling tsunami waves. That is, we will be numerically approximating solutions for these models. We will be determining the correctness of our models based on velocity and the actual velocity solution given by a function or we will be testing using numerical convergence if there is no actual known solution. We will need to find a solution that preserves momentum and energy over time. Thus, our model must preserve the shape of the wave over time.

In this project, we will be utilizing our knowledge of Numerical Analysis, Calculus, Differential Equations, and Linear Algebra, along with utilizing computer software in the programming language, python.

Approach:

Let us first simplify the problem. We will not be modeling 3D waves, but instead 2D waves. In several applications of shallow water wave theory, we see the utilization of the KDV Equation, Camassa-Holm (CH) Equation, and the Two Component Camassa-Holm (2CH) Equation. To approximate these solutions we will be using a variety of methods for discretizing space and time, including finite difference methods and Runge-Kutta methods. We will be using initial conditions such as functions that are known, plus using collected wave data.

KDV Equation:

We will start by testing the KDV Equation. The KDV Equation is given by:

Where “u” (Velocity) is a function of space and time. “u” is a vector because the velocity is different at each given location in space. Given the equation, we will need to discretize this function in both time and space, not only in order to numerically solve the equation, but also to obtain the velocity for the next time step.

First, we will discretize the spatial derivatives. First, let us simplify our equation:

Thus, the equation is rewritten as:

Now that our equation is simplified, we can now use our finite difference method to numerically approximate the spatial solution. Let us replace our partial differentials using a finite difference method (specifically using a central difference):

Thus, the rewritten equation is:

Let us simplify by writing this as:

Now that we have our equation simplified and spatially discretized, we now need to discretize our equation in time. There are two distinct ways to go about discretizing our equation in time. The first method is called the Forward Euler Method. This method is given by:

Thus, given our equation:

Let us now discretize and rewrite the variable as:

Thus, the equation would be:

Rewritten and simplified as:

Now that we have our equation, all we need is our initial conditions, our time and space intervals. Our initial condition for velocity will be given by:

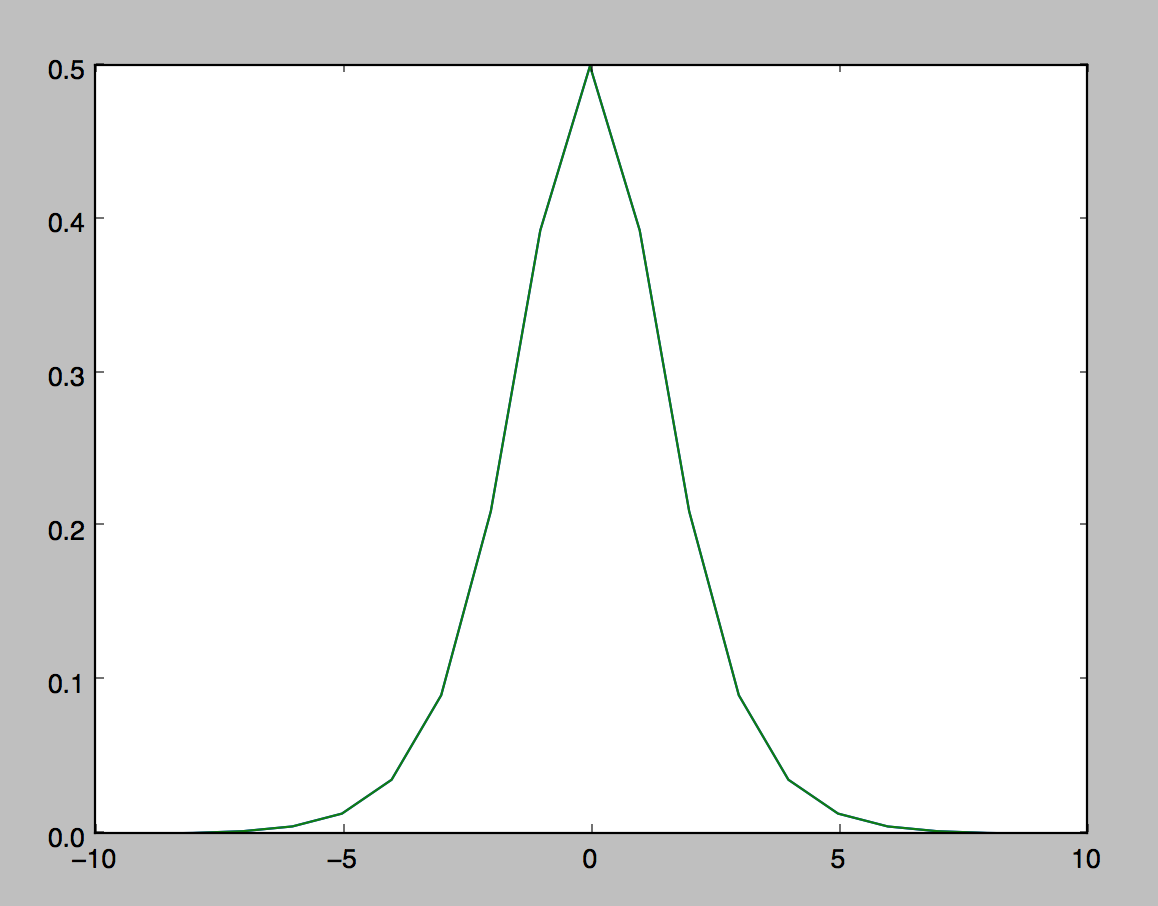
Space Interval: [-10,10], step size: 1

Time Interval: [0,2], step size: .01

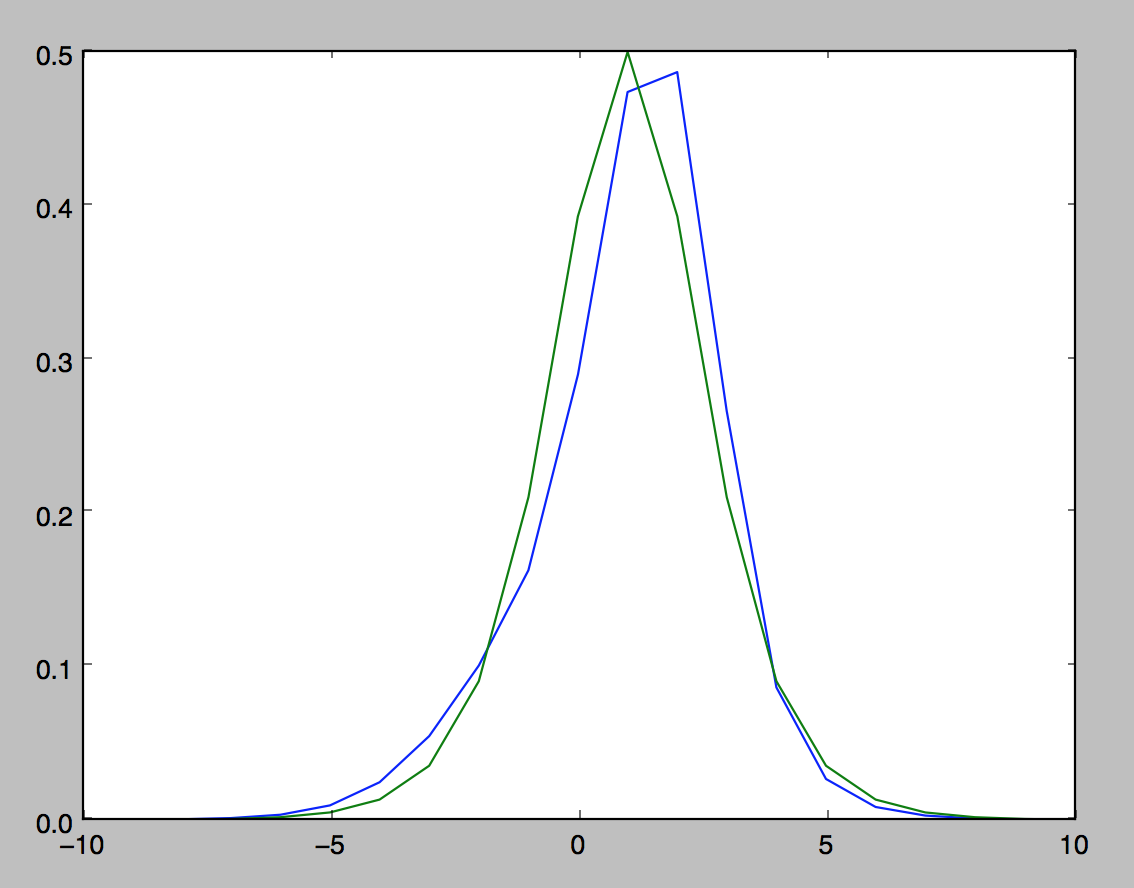
We will test our solution against the actual solution. The actual solution is given by:

KDV Equation Solution:

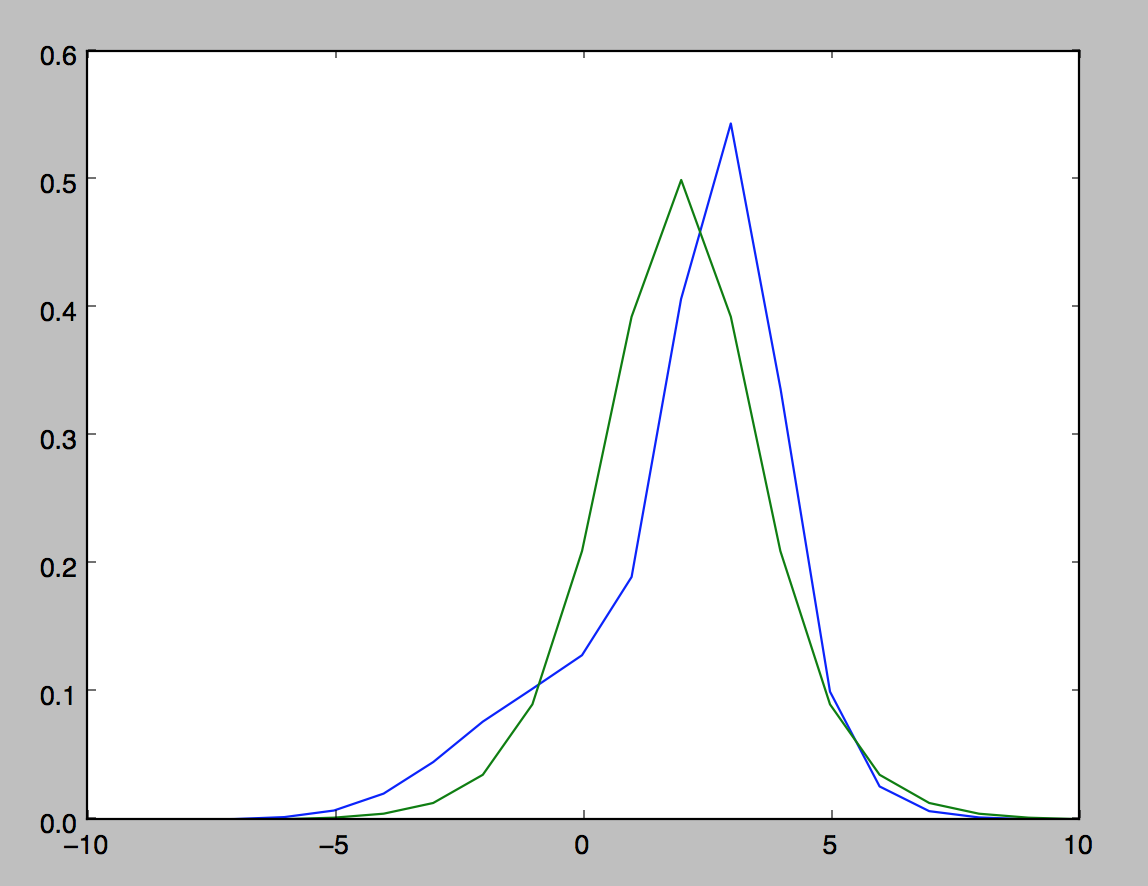
(Initial Condition) Velocity at Time = 0:



Velocity at Time = 1:



Velocity at Time = 2:



Analysis:

In numerical approximation, there will be error. Our goal is to minimize that error as much as possible. Our first model follows some aspects of wave behavior (for example moving forward as time progresses, but it isn’t very accurate to velocity). There are a few different things we can do to minimize that error. Let us begin by changing our method for time discretization.

KDV Equation (2):

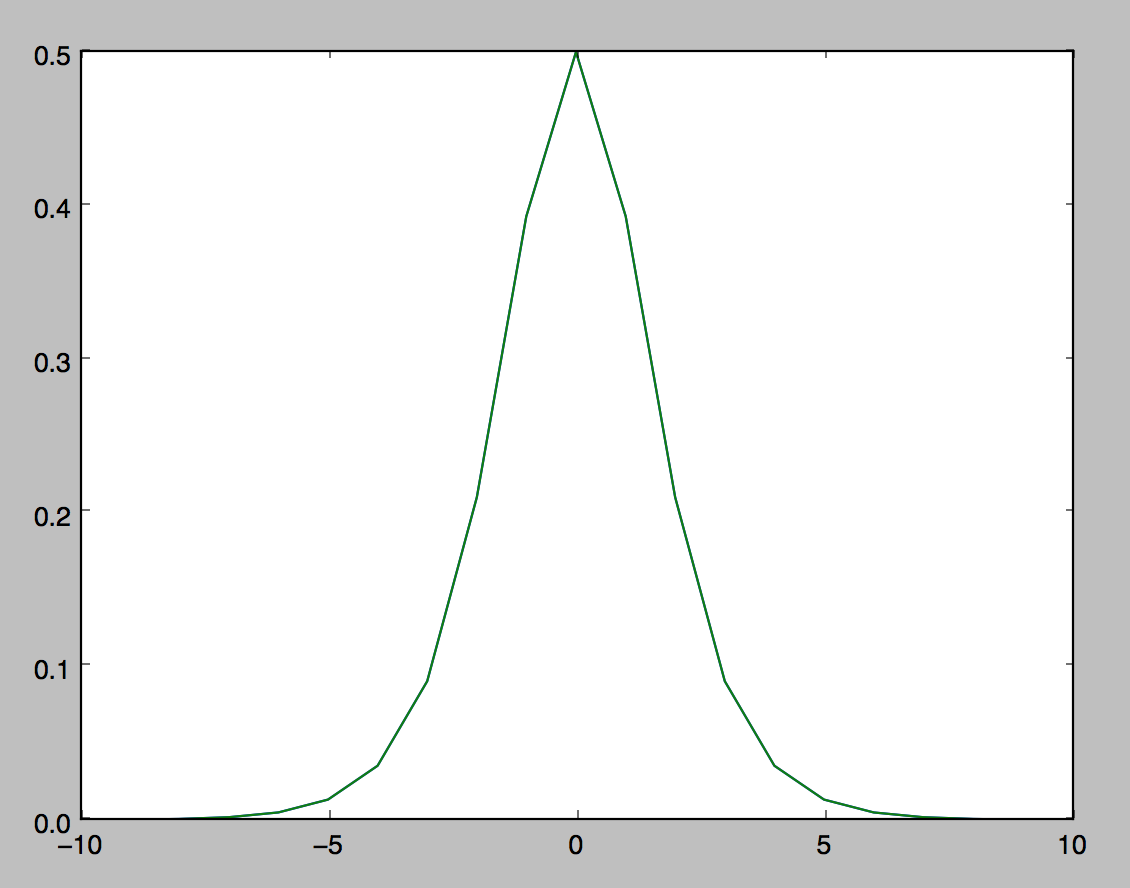
Next we will make a slight modification to the KDV Equation to try and stabilize the solution a bit more. To do this we will use a third order Runge-Kutta Method, which is a strong-stability preserving (SSP) method, to discretize the time derivative in the given partial differential equation.

Thus we will replace the Forward Euler Method with the third order Runge-Kutta Method, which is given by the following:

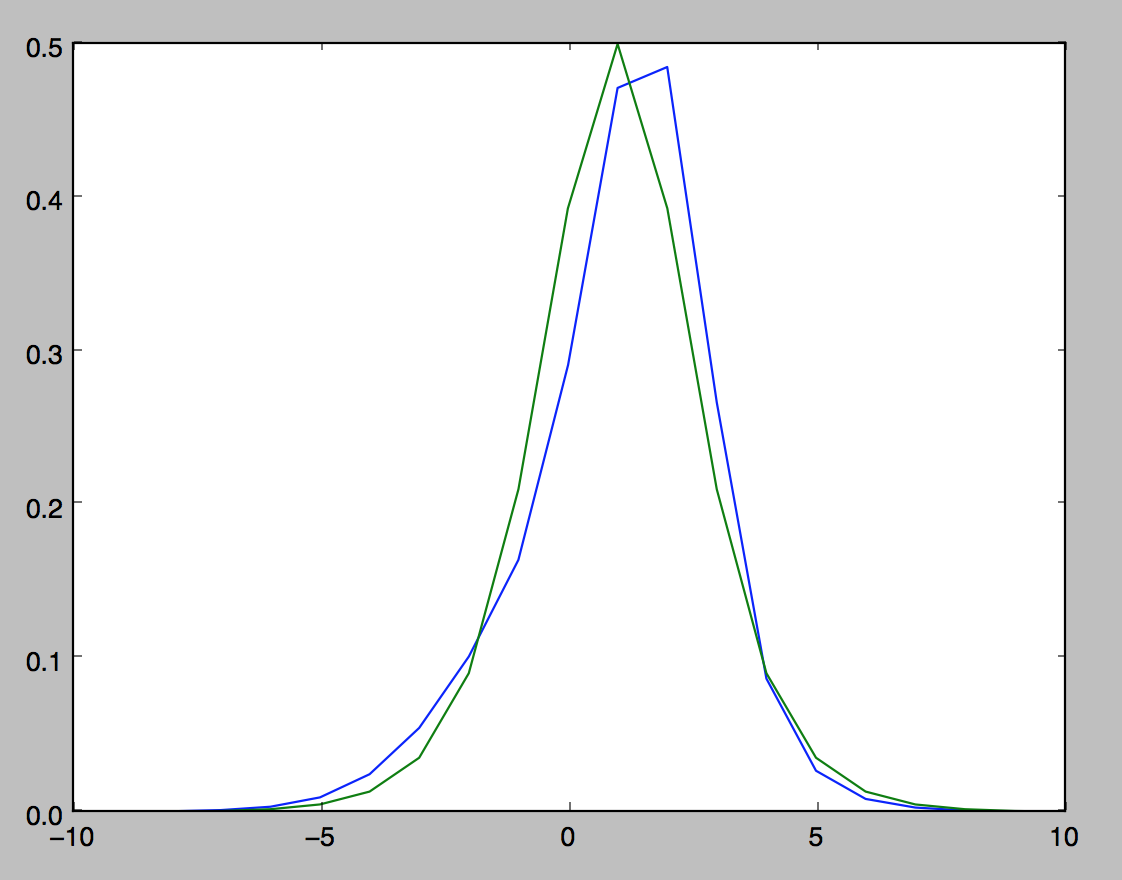
That is, we are keeping everything else the same. We will be using the same discretization for the spatial derivatives, the same initial condition, and the same intervals for both space and time.

KDV(2) Equation Solution:

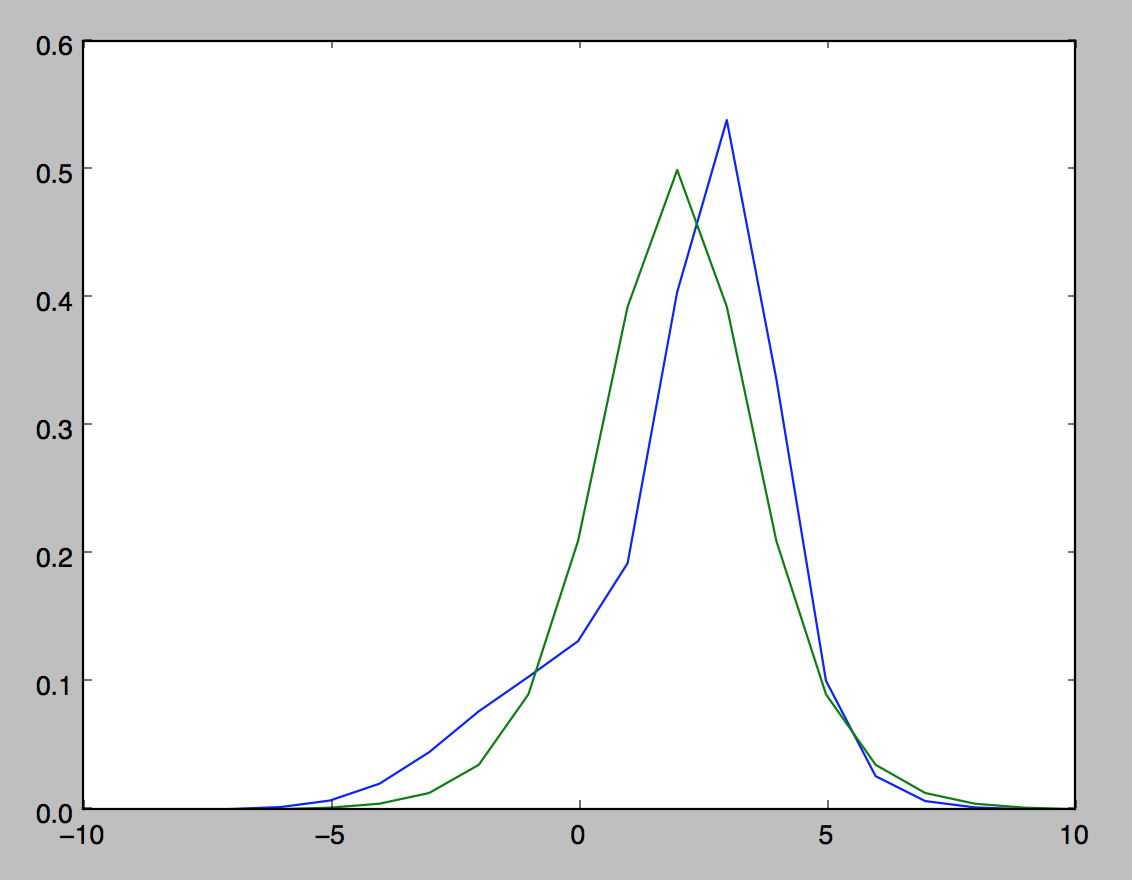
(Initial Condition) Velocity at Time = 0:



Velocity at Time = 1:



Velocity at Time = 2:



Analysis:

However, as we can see the solution still isn’t as stable as we need it to be and the Runge-Kutta method didn’t change our solution. The equation must not be as stable as we need it to be. Let us next replace the KDV Equation with a better equation for approximating tsunami wave behavior. That is, let us try the Camassa-Holm Equation and see if we arrive at better results.

Camassa-Holm (CH) Equation:

Next we will test the Camassa-Holm (CH) Equation using finite difference methods to approximate the numerical solution. The CH Equation is given by:

Here, “u” (Velocity) and “m” (Momentum) are both functions of time and space. The second equation is the relationship between velocity and momentum. Again, we will need to discretize both the time and space derivatives in order to numerically approximate the solution.

First, we will discretize the spatial derivatives. First, let us simplify our equation:

Thus, the equation rewritten is:

Now that we our equation is simplified, we can now use our finite difference method to numerically approximate the spatial solution. Let us replace our partial differentials with finite difference methods:

Thus, the equation rewritten is given by:

Now that we have our equation simplified and spatially discretized, we now need to discretize our equation in time. For this we will reuse the third order Runge-Kutta Method from the KDV Equation. Again, the third order (SSP) Runge-Kutta Method is given by:

Next, we need to discretize our relationship equation given by:

Once discretized we get:

In order to be able to solve this we will have to utilize our knowledge of linear algebra and solving matrices. First let us rewrite and simplify:



Now that we have our equation, all we need is our initial condition, our time and space intervals. Our initial condition for velocity will be given by the following piecewise function:

This piecewise function generates a wave with what is called a peakon or a peakon soliton. A peakon is a soliton with a discontinuous first derivative. Using the initial condition for velocity, we can use the relationship function/matrix and find the initial condition for momentum.

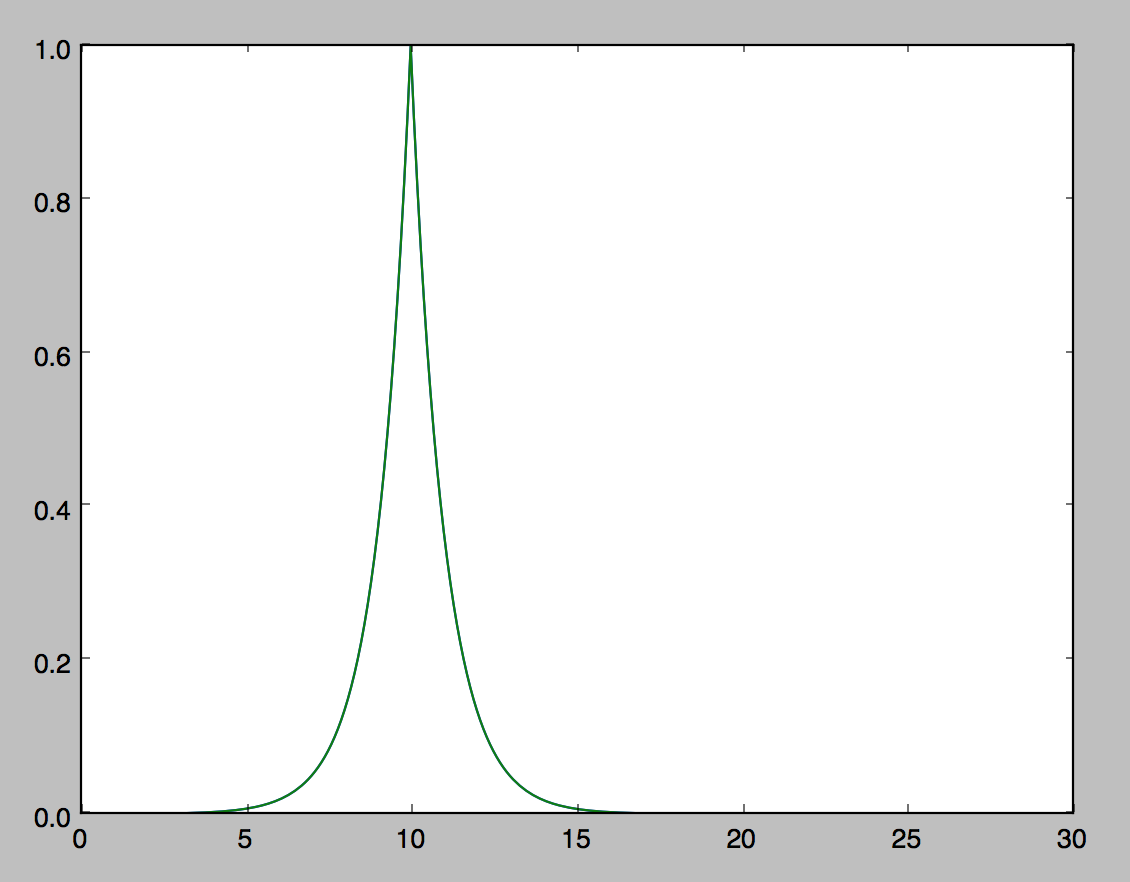
Space Interval: [0,30], step size: .1

Time Interval: [0,2], step size: .01

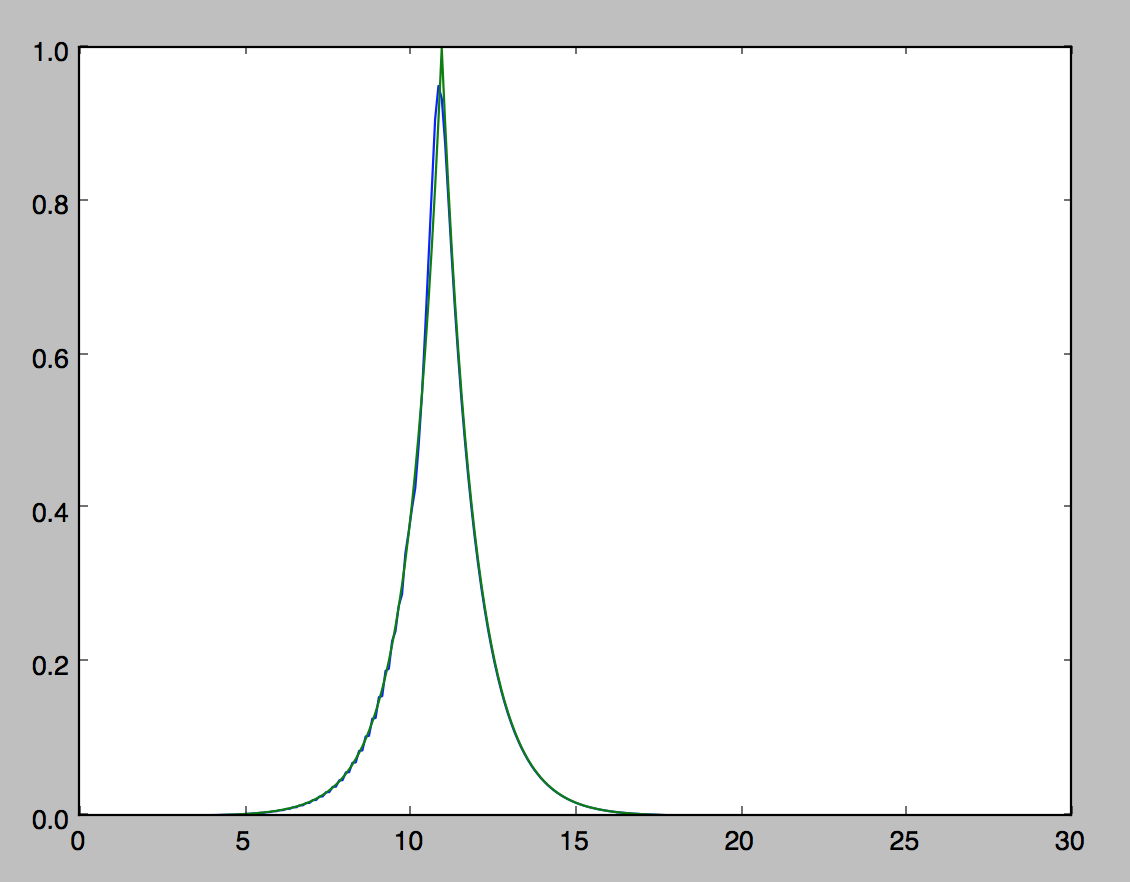
Again, we will test our solution against the actual solution. Now, the actual solution will be given by the following function:

Camassa-Holm Equation Solution:

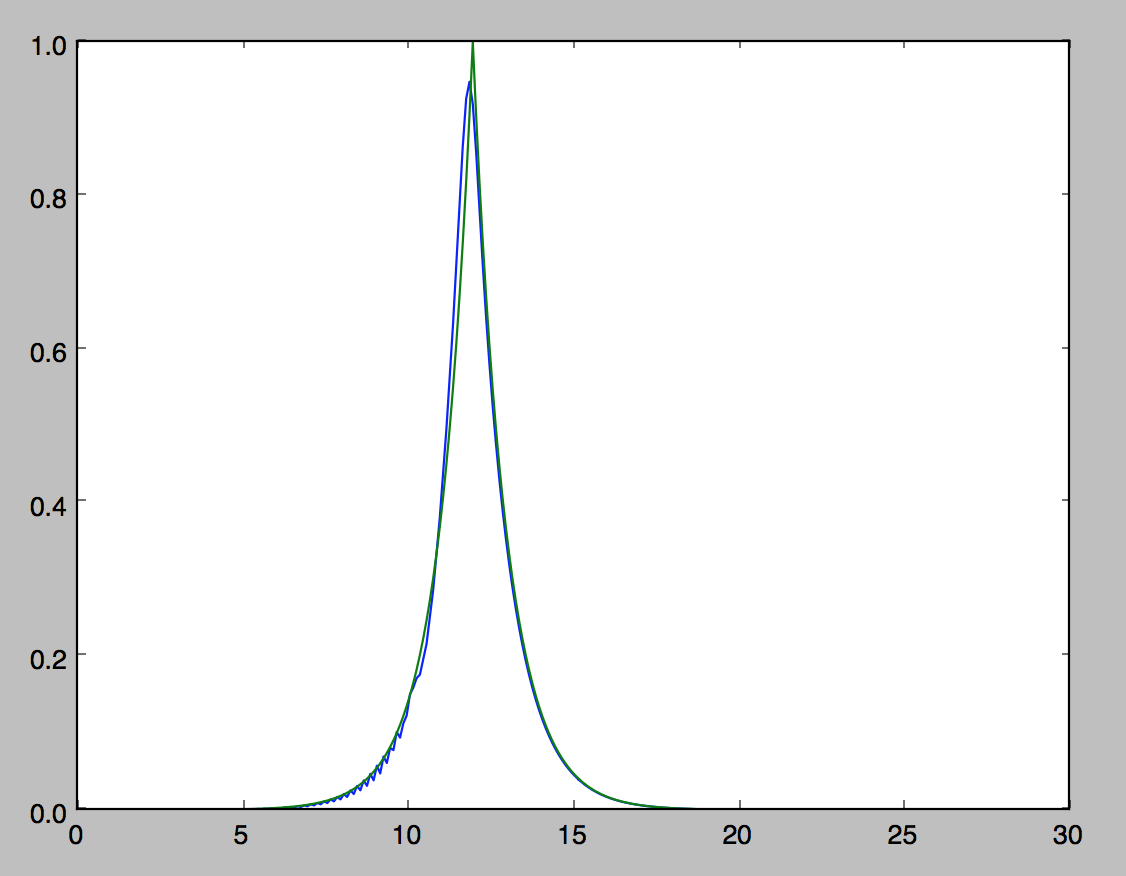
(Initial Condition) Velocity at Time = 0:



Velocity at Time = 1:



Velocity at Time = 2:



Analysis:

As we can see this equation is much better at preserving the shape of the wave. However, at time two we still start to see numerical diffusion starting to take form on the bottom left side of the wave. Overall, this solution is much more stable than the KDV equation.

Two Component Camassa-Holm (2CH) Equation:

Next we will test the Two Component Camassa-Holm (2CH) Equation using finite difference methods to approximate the numerical solution. The 2CH Equation is given by:

You may recognize the first portion of the equation as the CH Equation. Just like with the CH Equation, “u” (Velocity) and “m” (Momentum) are both functions of time and space. The second equation, again, is the relationship between velocity and momentum. However, there are slight differences between this equation (2CH) and the CH Equation. The 2CH Equation has a pressure term. This pressure term is given by:

The “p” stands for Rho (Density), and the “g” for the gravity constant. The third equation is a continuity expression for density, and that is given by:

Again, we will need to discretize both the time and space derivatives in order to numerically approximate the solution for the tsunami wave model.

First, we will discretize the spatial derivatives. First, let us simplify our equation:

Now that we our equation is simplified, we can now use our finite difference method to numerically approximate the spatial solution. Let us replace our partial differentials with finite difference methods:

Thus, the equation rewritten is given by:

Next we need to discretize our second equation (the continuity expression for Density) in space. Again the continuity expression for Density is given by:

After discretizing/replacing the partial differential equation with a finite difference method, we get the following equation:

Now that we have our equations simplified and spatially discretized, we now need to discretize our equations in time. For this we will reuse the third order Runge-Kutta Method from the KDV Equation. Again, the third order (SSP) Runge-Kutta Method is given by:

Next, we need to discretize our relationship equation given by:

Once discretized we get:

In order to be able to solve, we will have to utilize our knowledge of linear algebra and solving matrices. Refer to the written out form of the linear algebra required to solve this relationship equation previously mentioned.

Now that we have our equations, all we need is our initial conditions, our time and space intervals. For our initial condition here, we will use what is called a dam break initial condition. Our initial condition for density will be given by the following function:

The initial condition for velocity will be an array of zeros. Using the initial condition for velocity, we can use the relationship function/matrix and find the initial condition for momentum.

Space Interval: [-12,12], N = 200 (Number of steps taken)

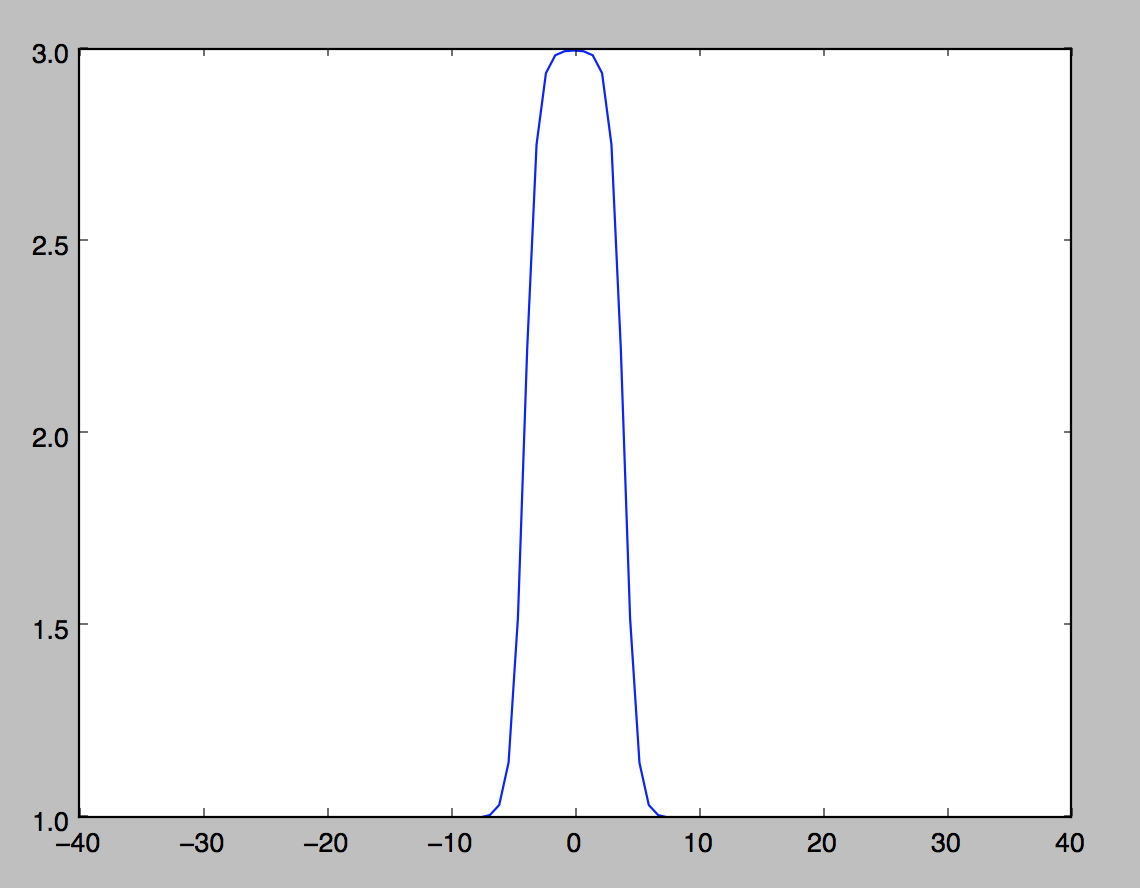
Time Interval: [0,2], step size: .01

However, for this model the actual solution is not known. Thus, we will test our solution by using numerical convergence. For numerical convergence we will test a variety of values for N (Number of points tested). First we will test 100 points, then 200 points, then 400 points, and finally we will test 800 points.

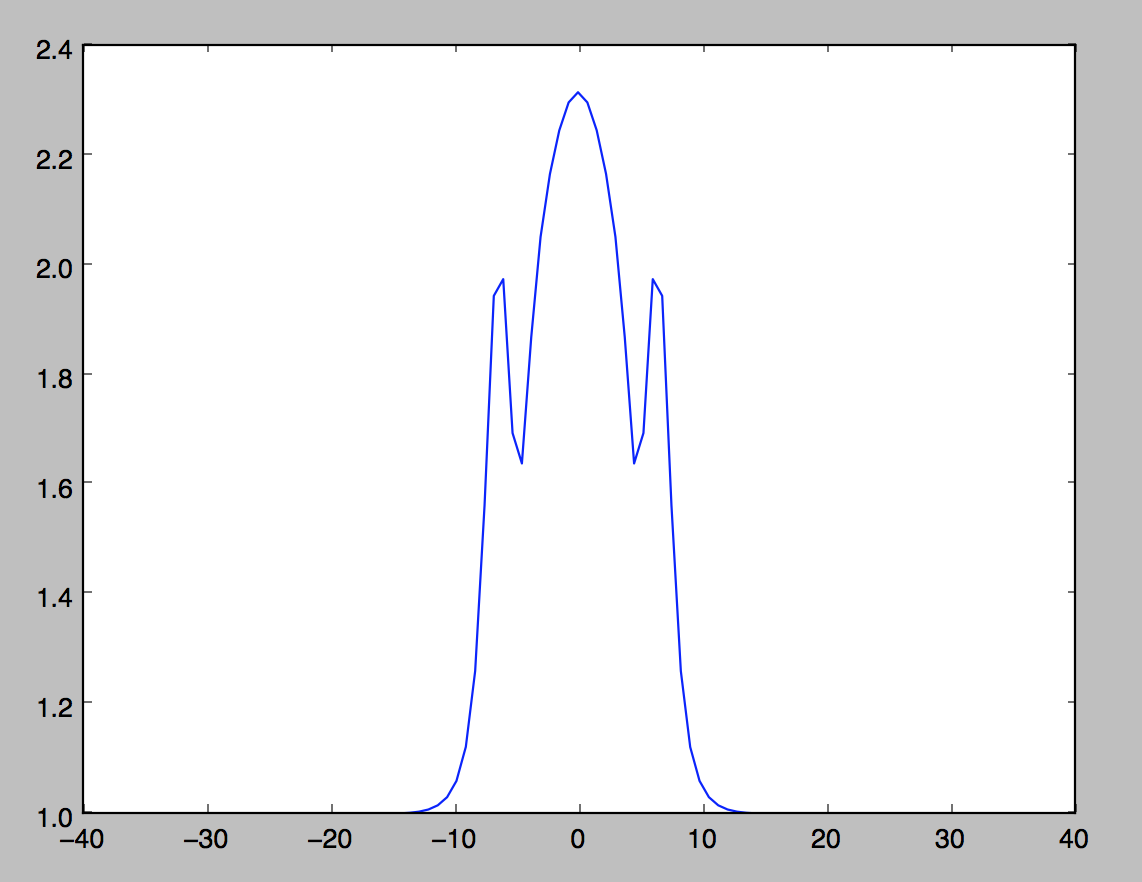
Two Component Camassa-Holm (2CH) Equation Solution:

*100 Points:*

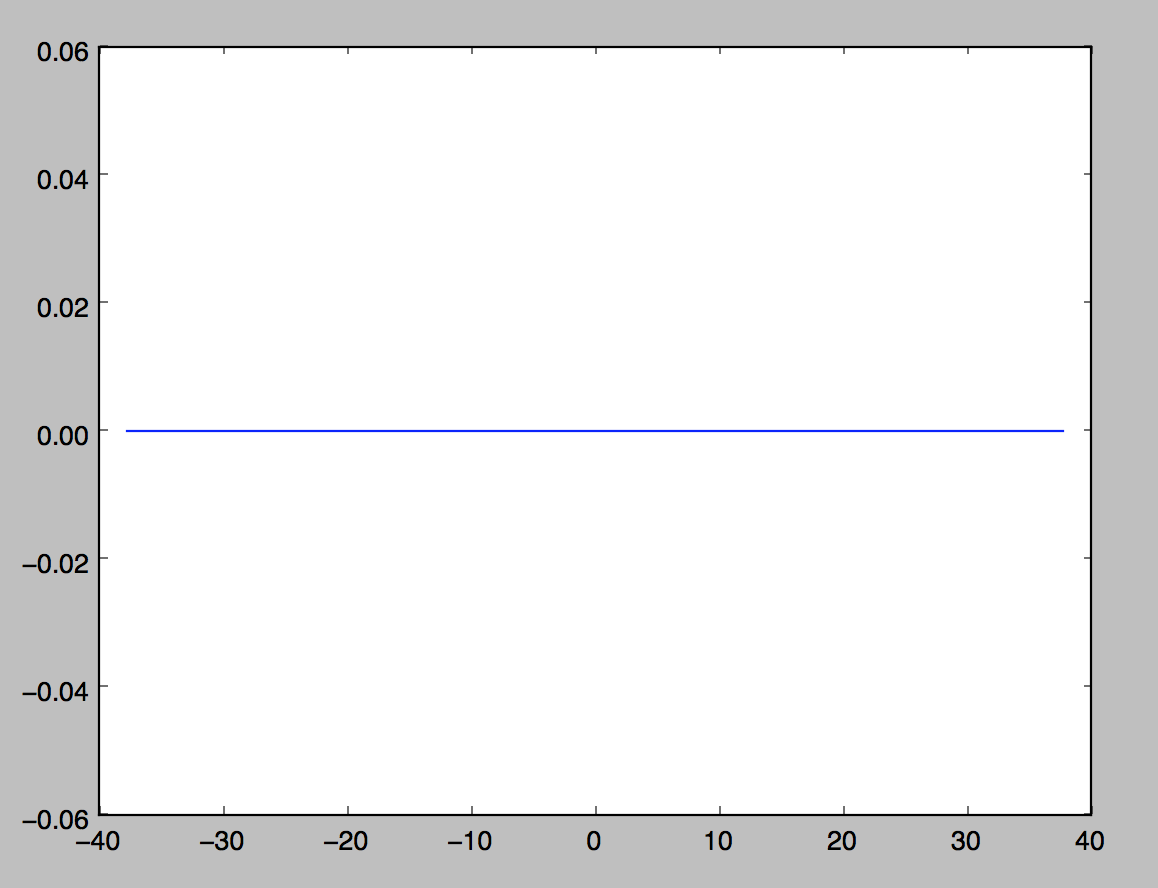
(Initial Condition) Density at Time = 0:



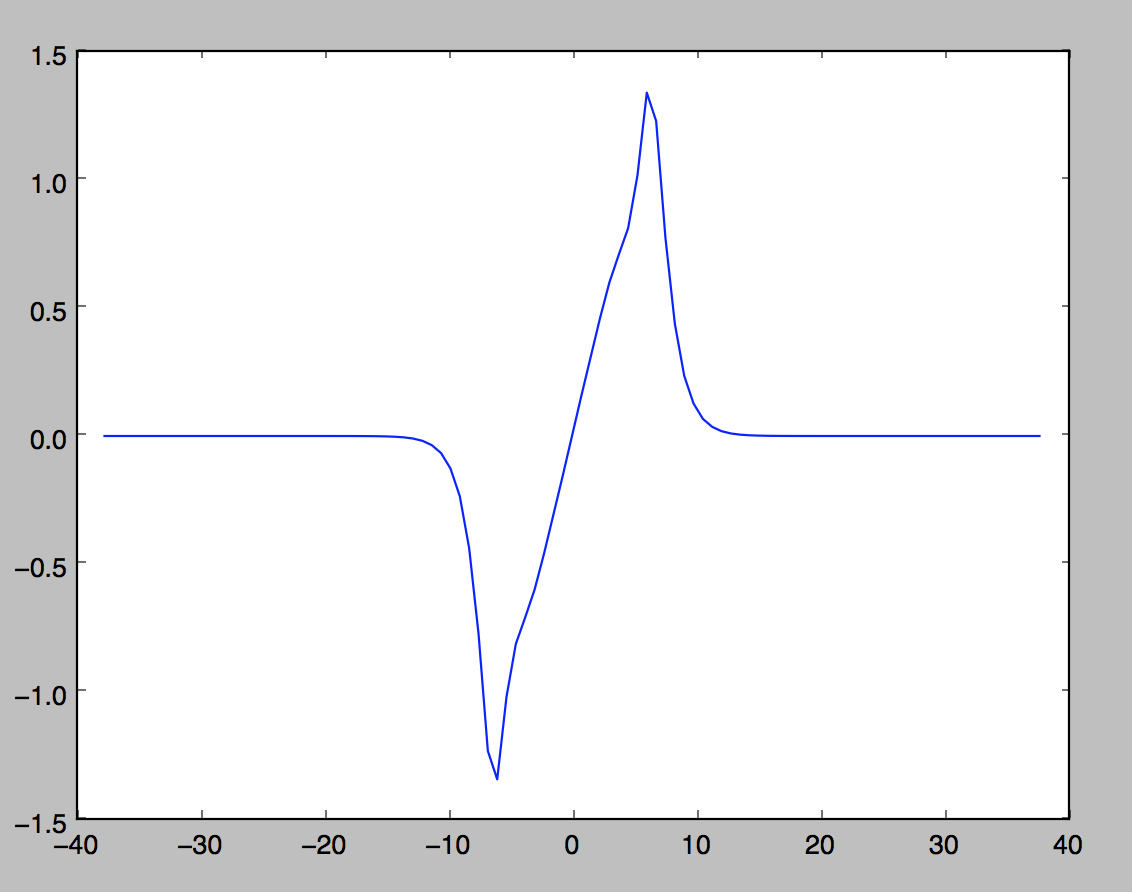
Density at Time = 2:



(Initial Condition) Velocity at Time = 0:

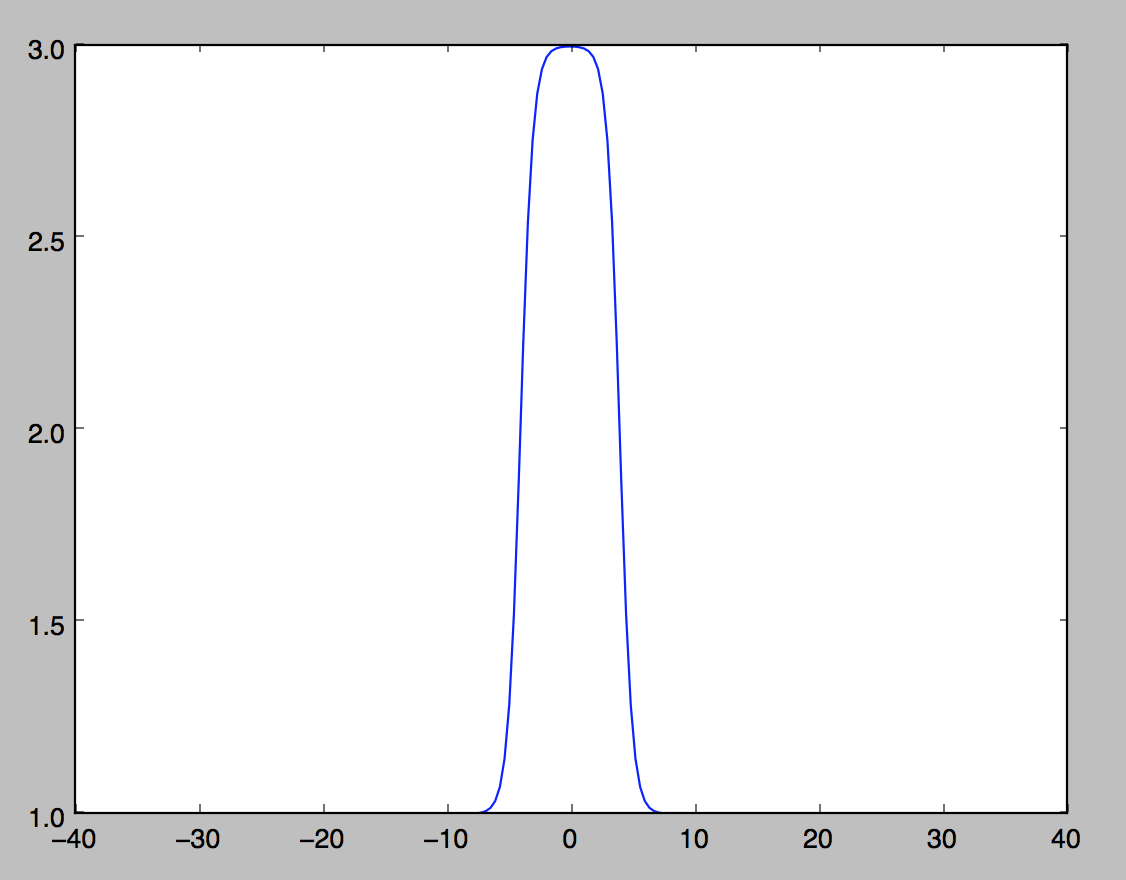


Velocity at Time = 2:

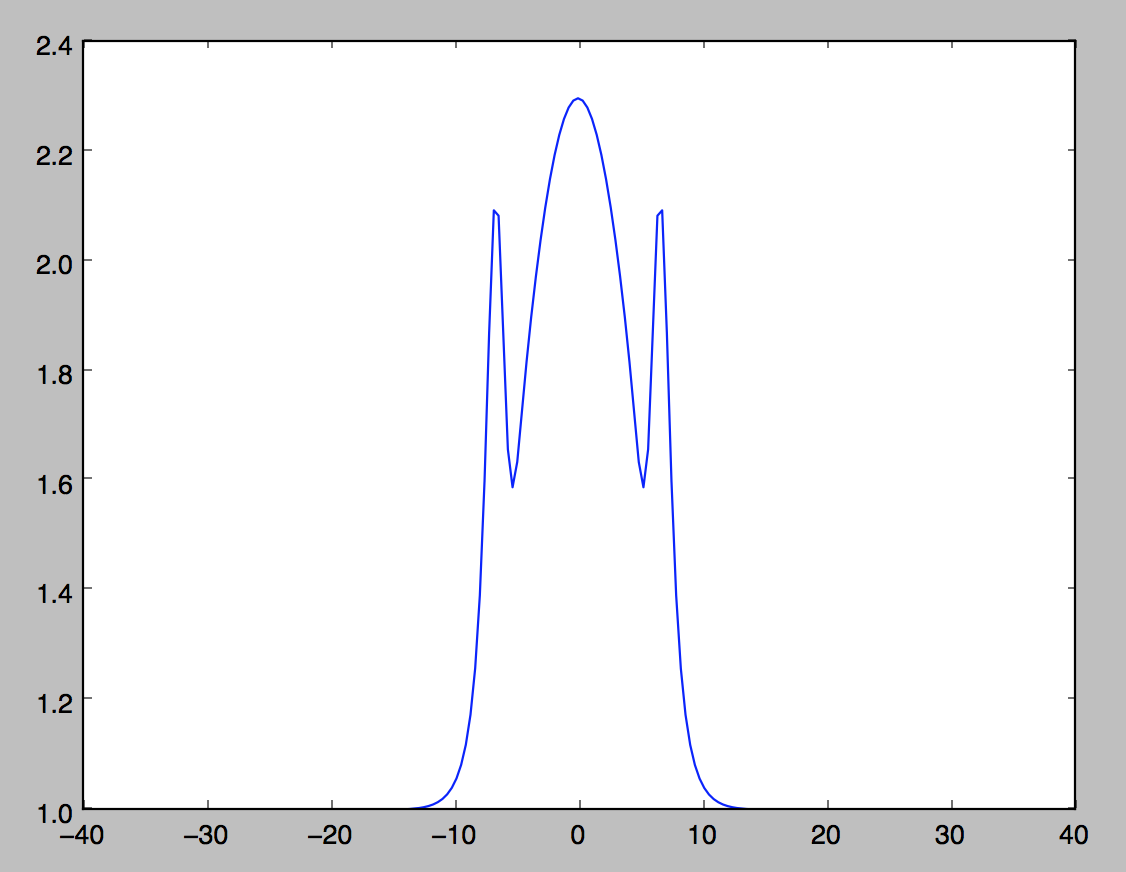


*200 Points:*

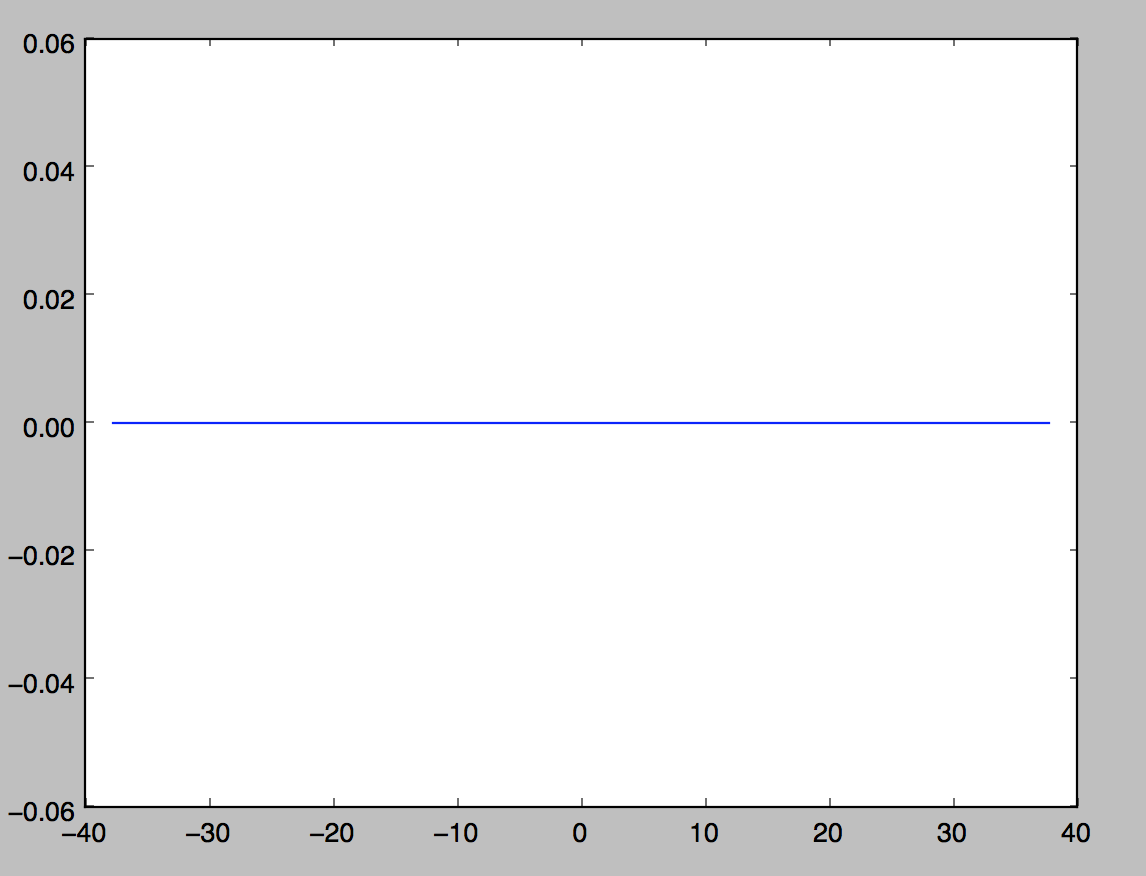
(Initial Condition) Density at Time = 0:



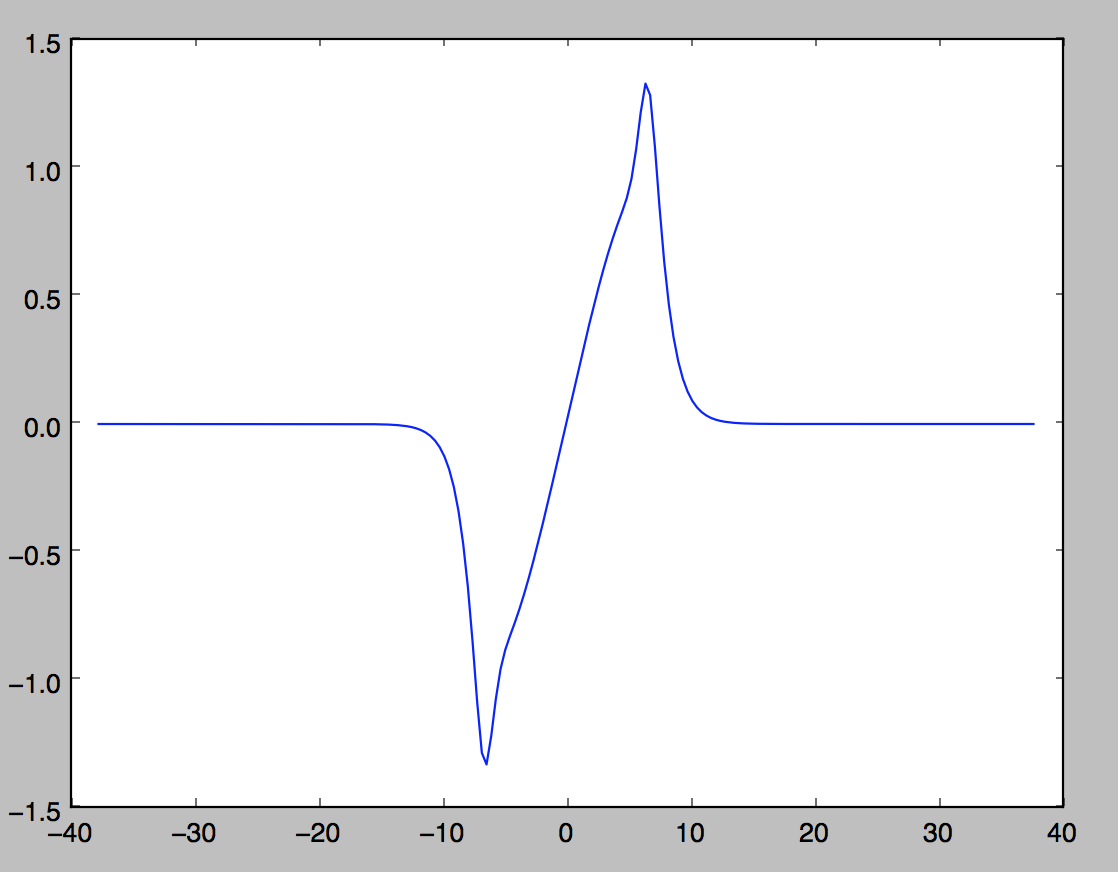
Density at Time = 2:



(Initial Condition) Velocity at Time = 0:

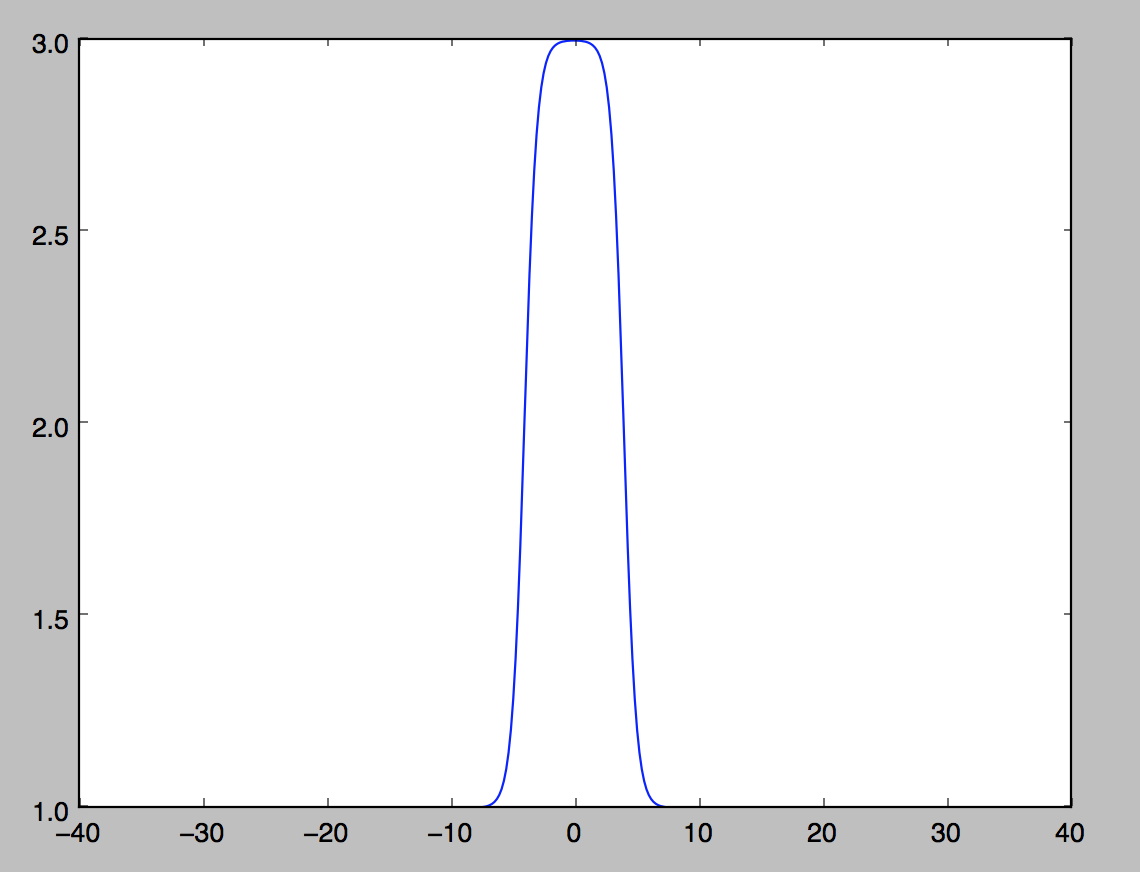


Velocity at Time = 2:

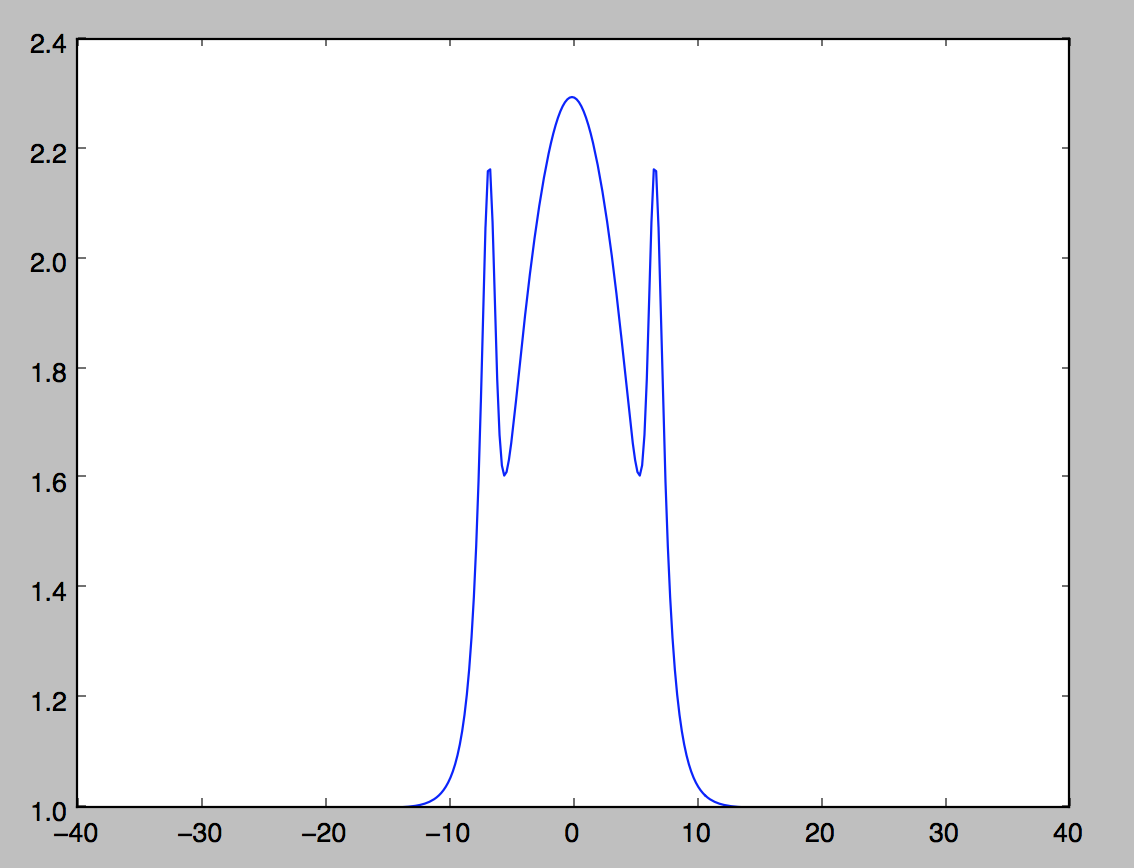


*400 Points:*

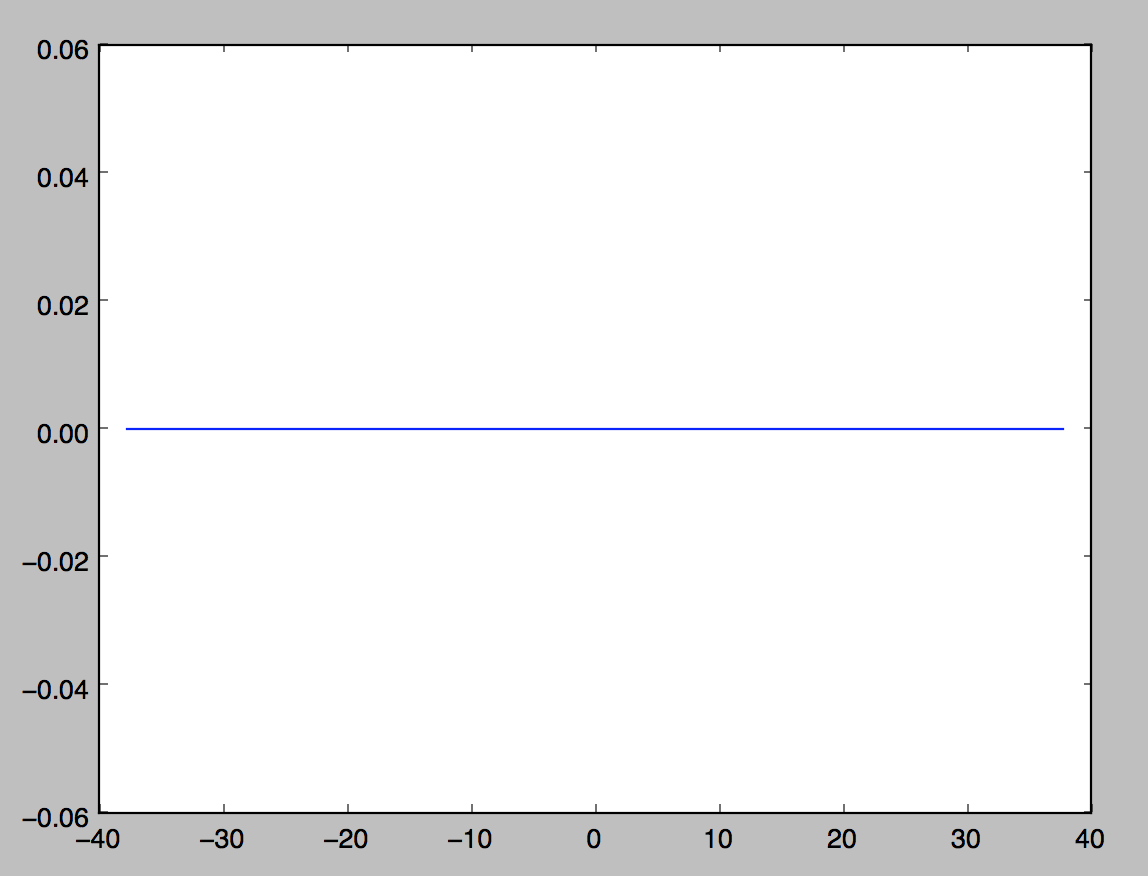
(Initial Condition) Density at Time = 0:



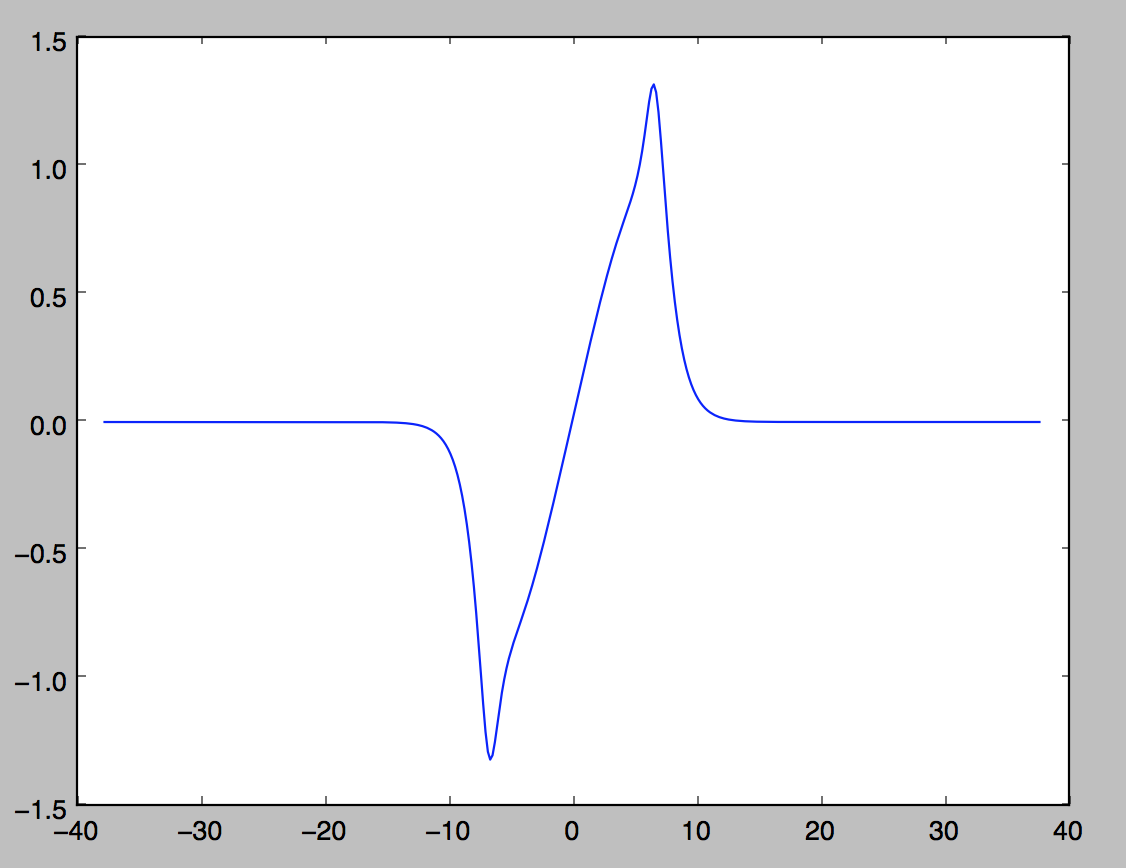
Density at Time = 2:



(Initial Condition) Velocity at Time = 0:

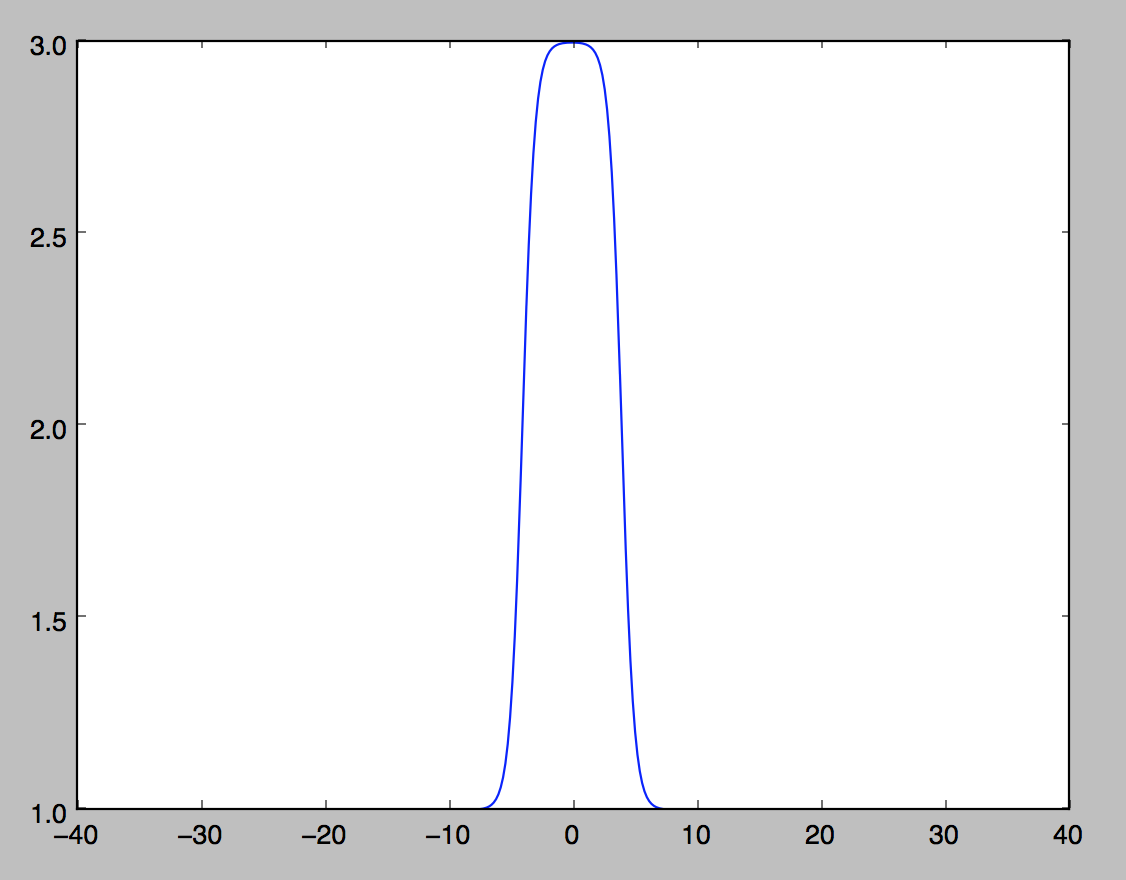


Velocity at Time = 2:

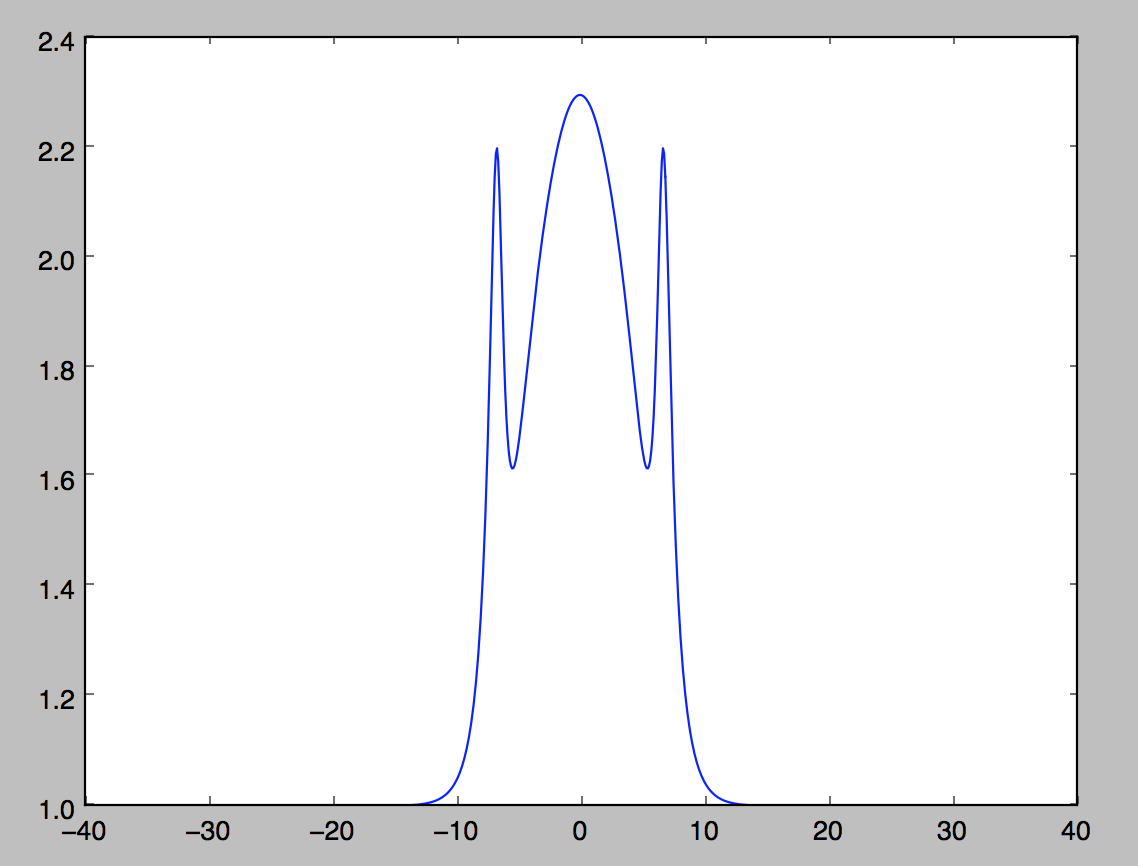


*800 Points:*

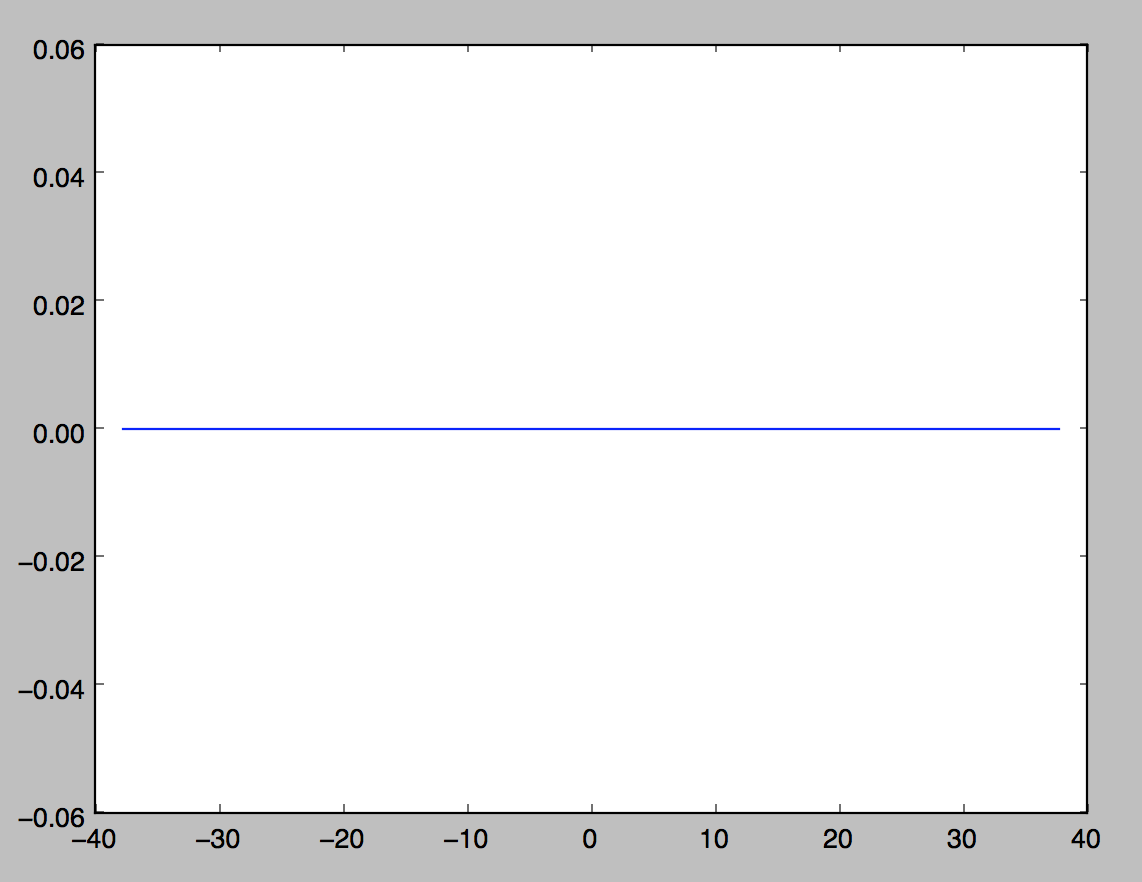
(Initial Condition) Density at Time = 0:



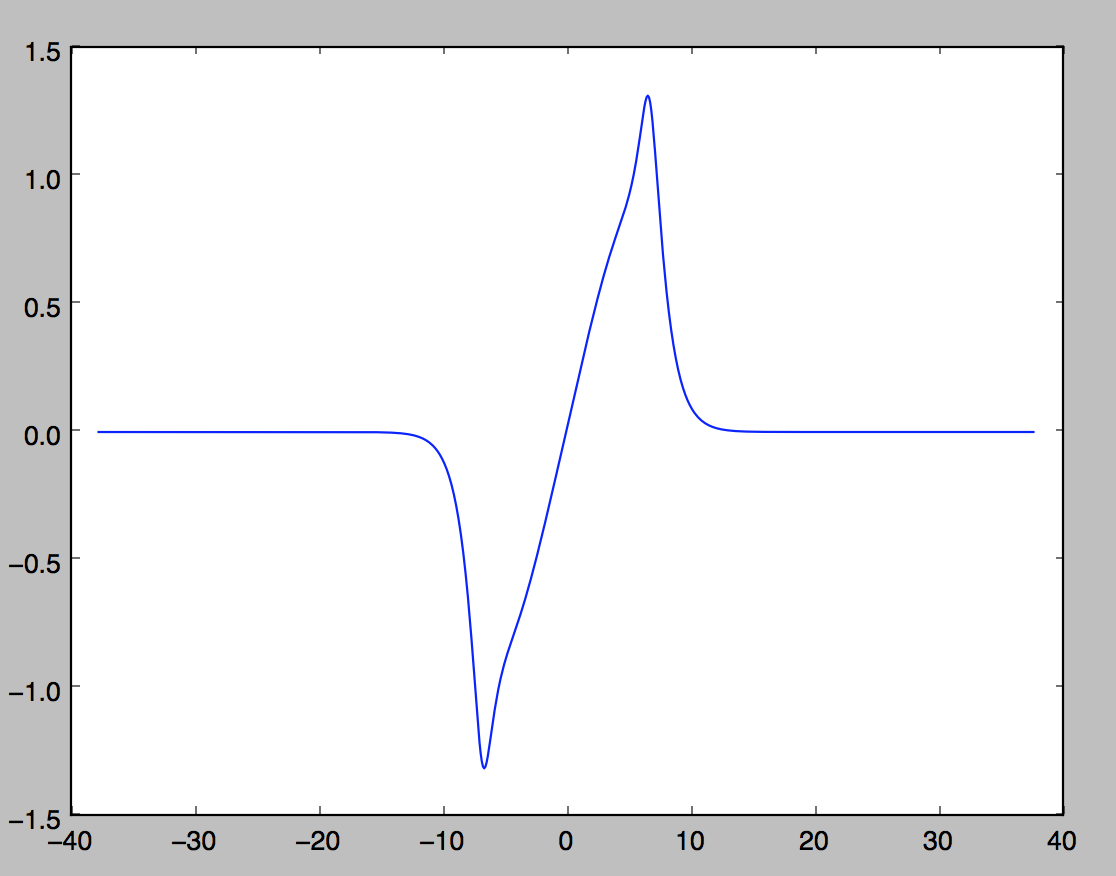
Density at Time = 2:



(Initial Condition) Velocity at Time = 0:



Velocity at Time = 2:



Analysis:

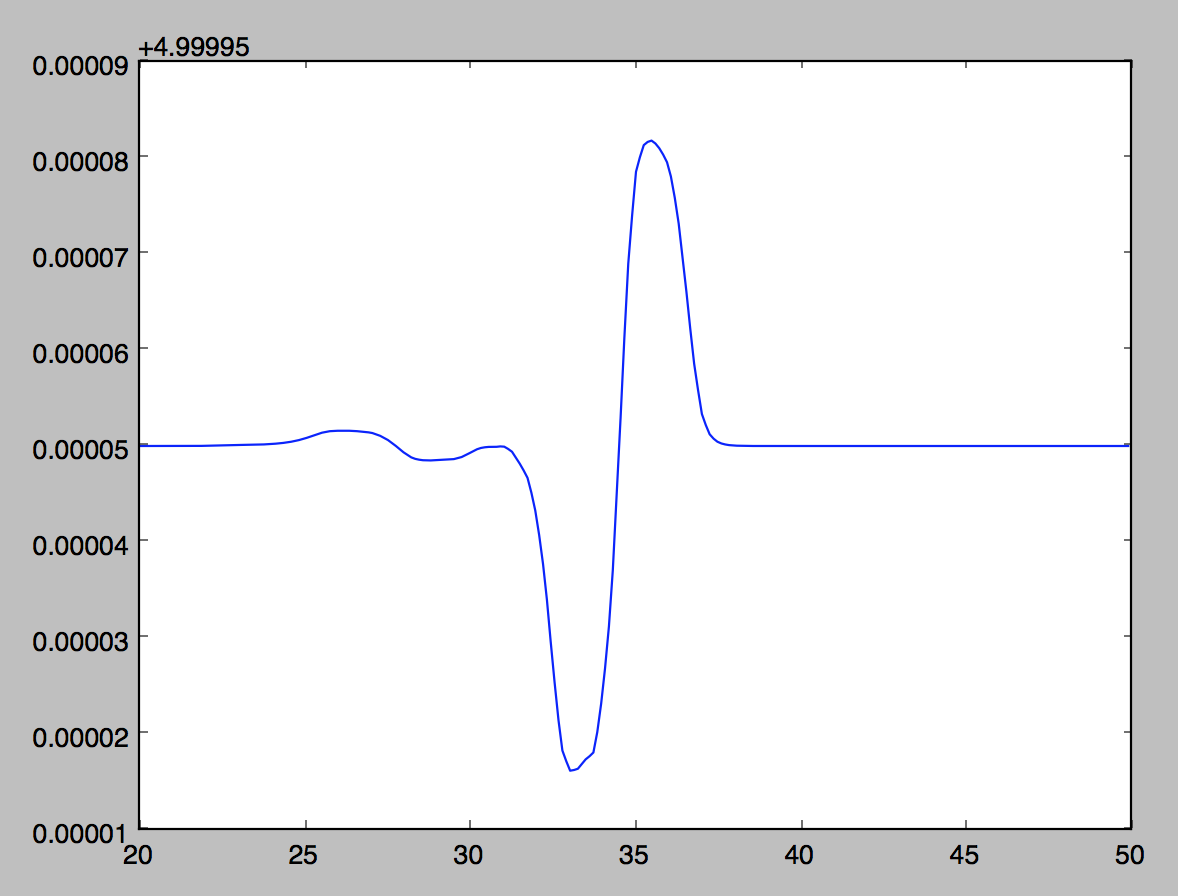
As we can see by testing numerical convergence and analyzing the solutions, the solutions all converge to the same wave shape. This is the best numerical solution for modeling tsunami waves thus far. Now, let us take this solution and test different initial conditions using actual tsunami wave data.

Testing Real Tsunami Data and Solution:

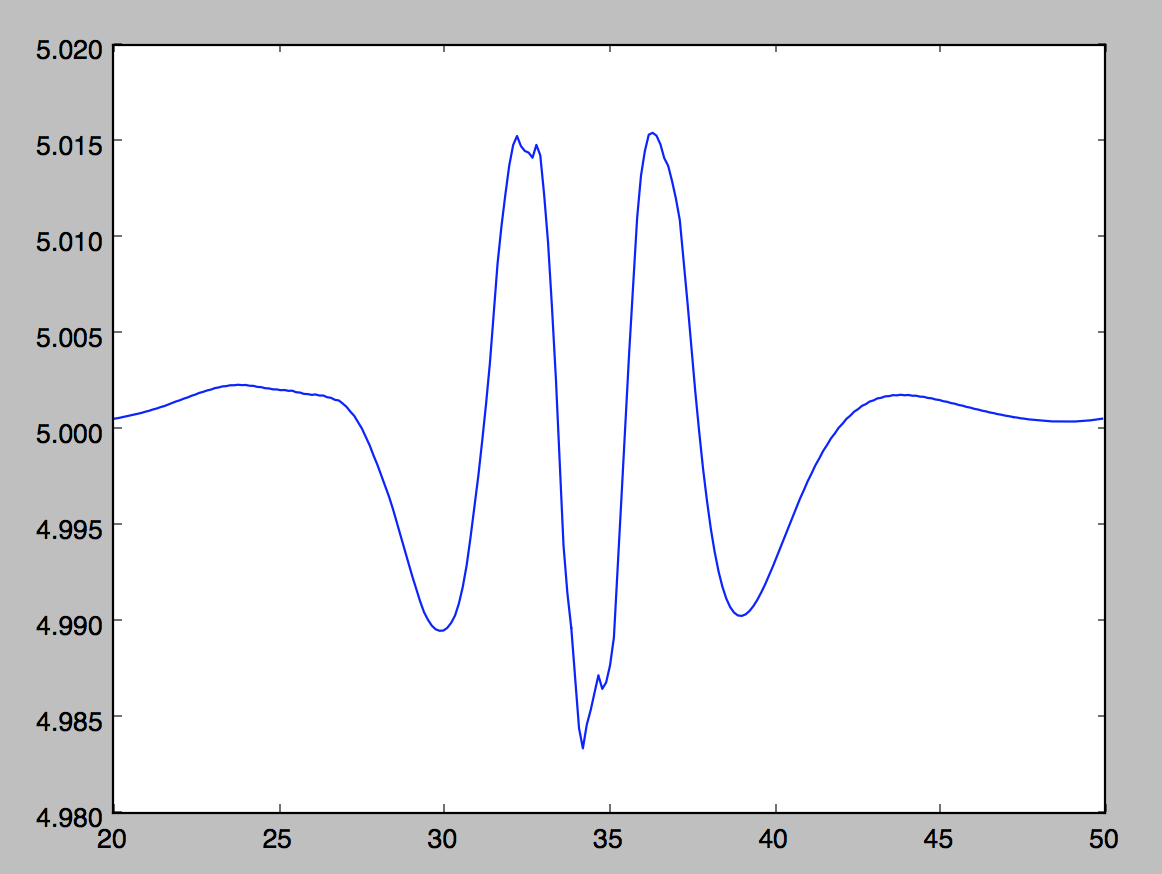
The data we will be using was collected from the NOAA Center for Tsunami Research at the Pacific Marine Environmental Laboratory. The data that was provided includes four sets. Each set includes the initial conditions for density and velocity. The first set has 256 data points for both density and velocity, the second has 512 data points for both density and velocity, the third has 1024 data points for both density and velocity, and finally, the last set has 4096 data points for both density and velocity.

Let us first test *256 Data Points*:

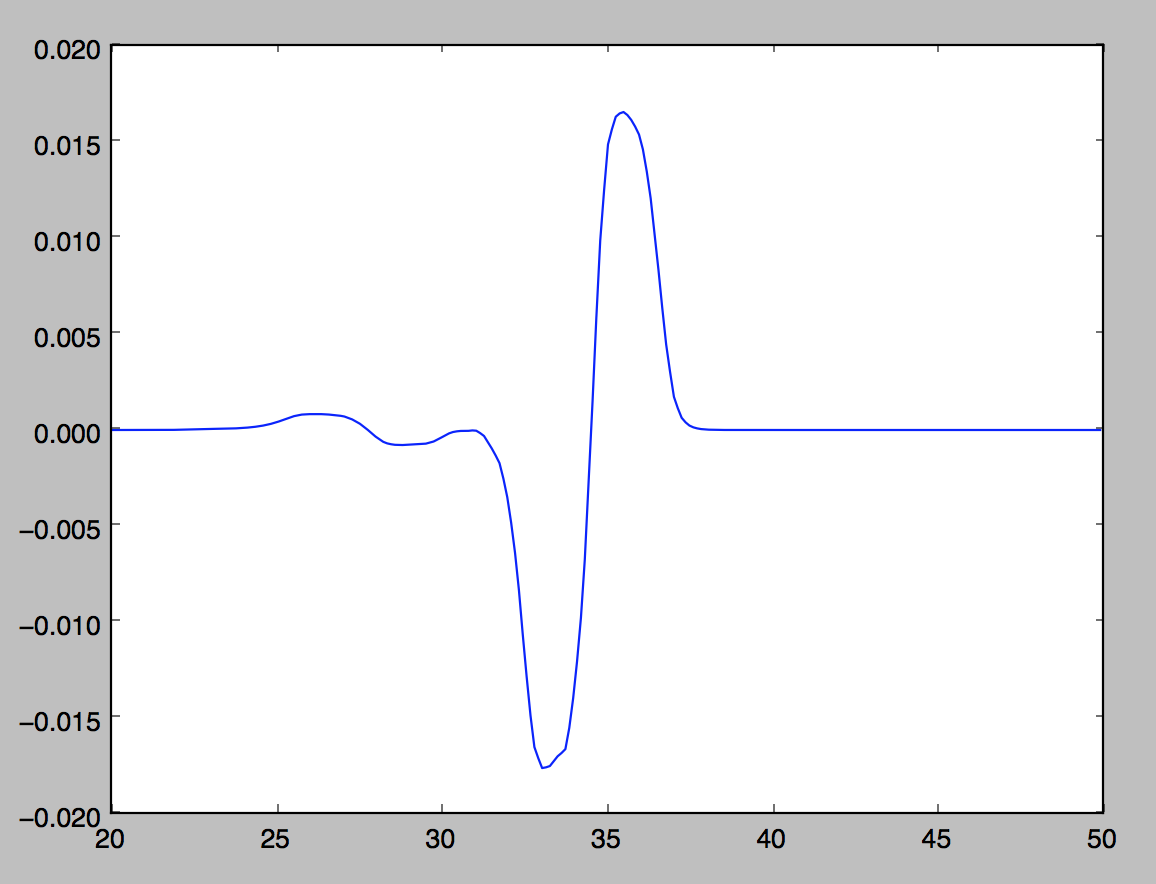
(Initial Condition) Density at Time = 0:



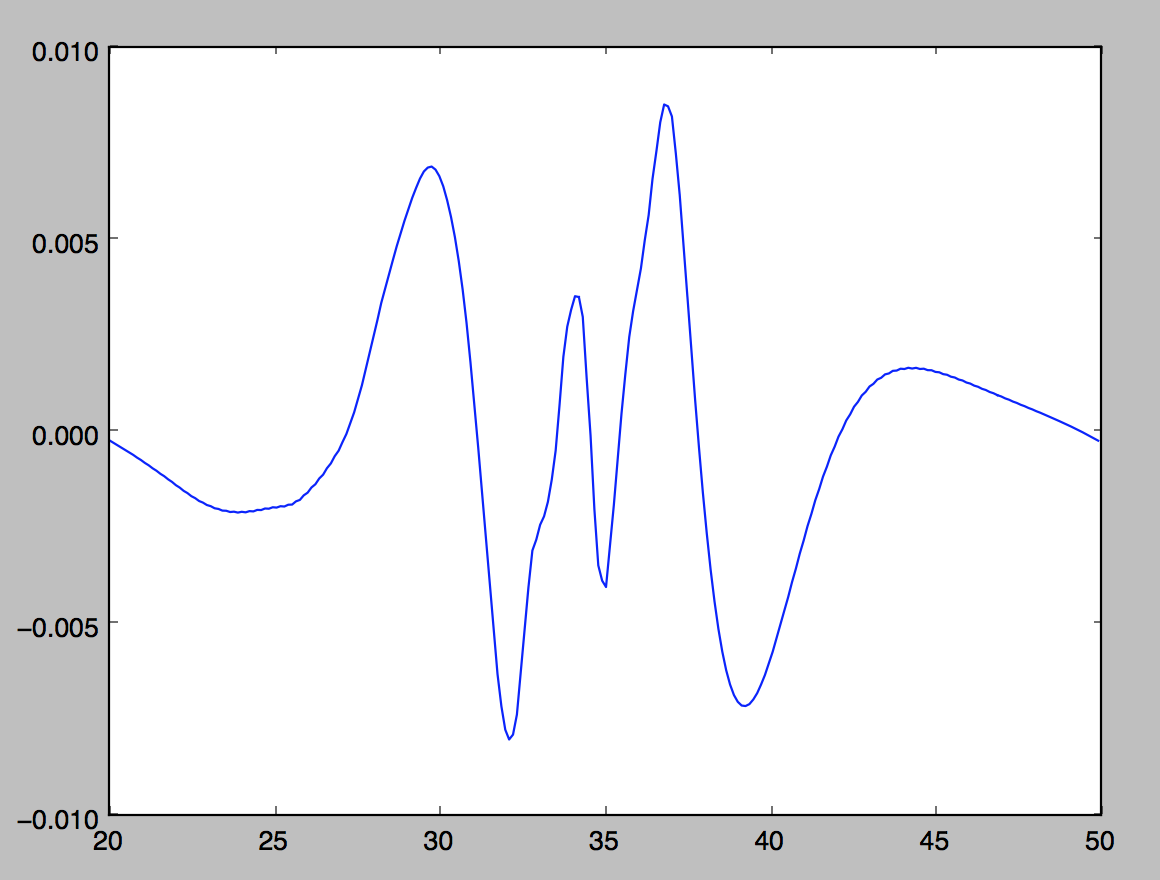
Density at Time = 2:



(Initial Condition) Velocity at Time = 0:

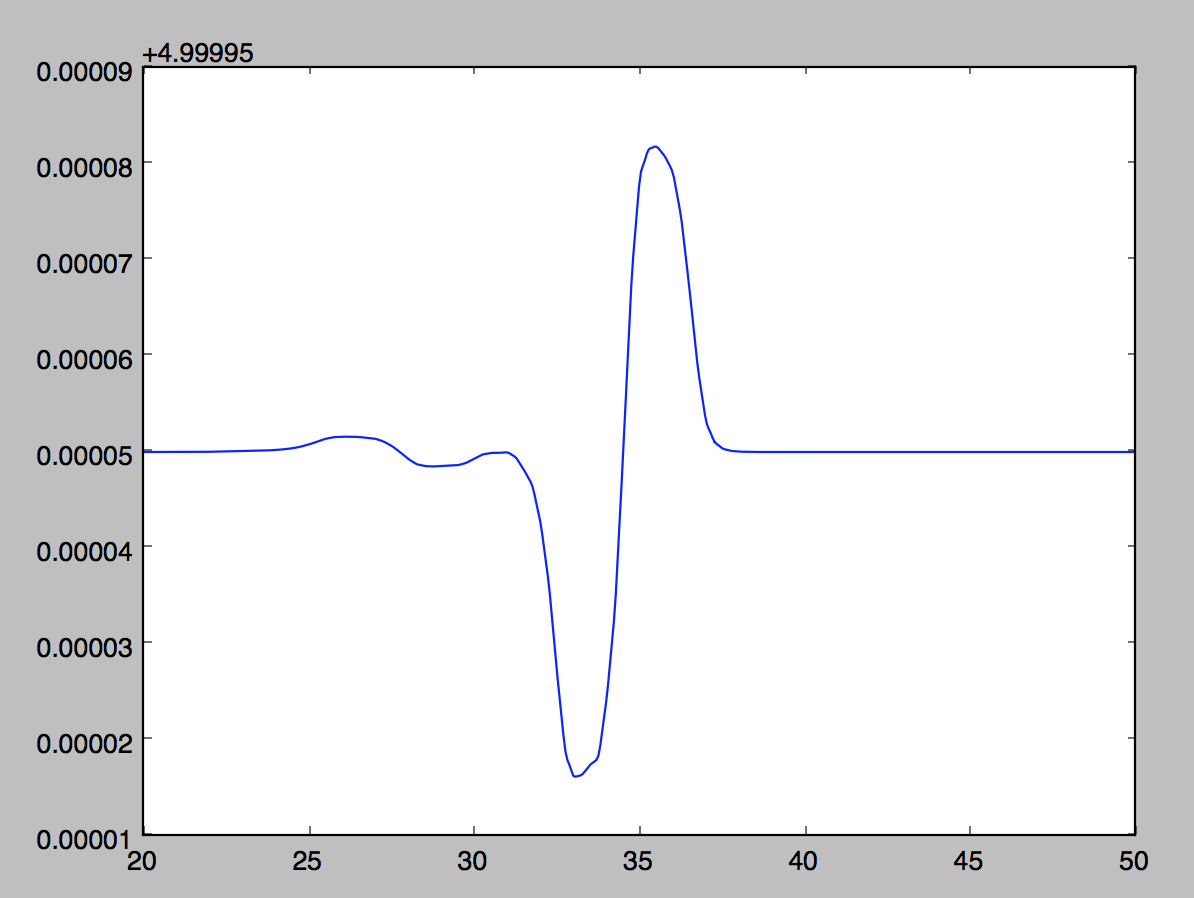


Velocity at Time = 2:

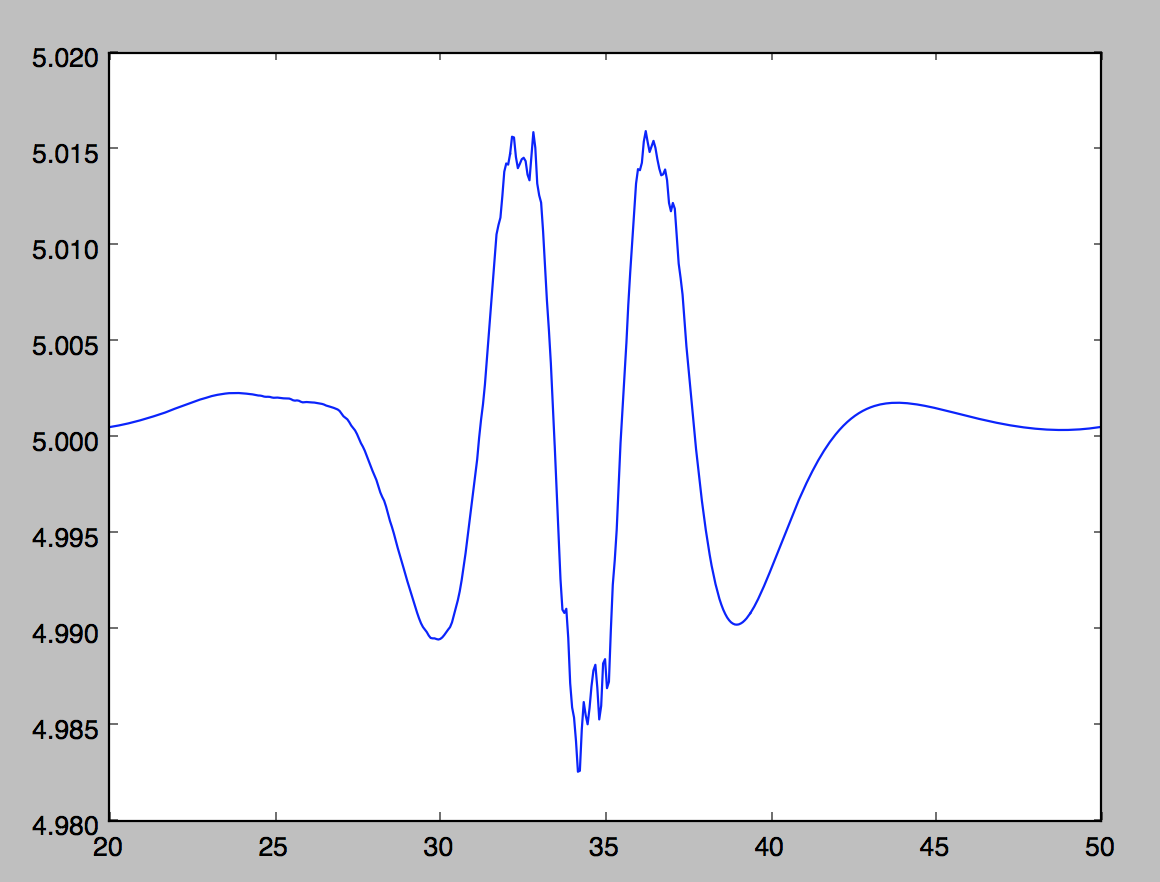


*512 Data Points:*

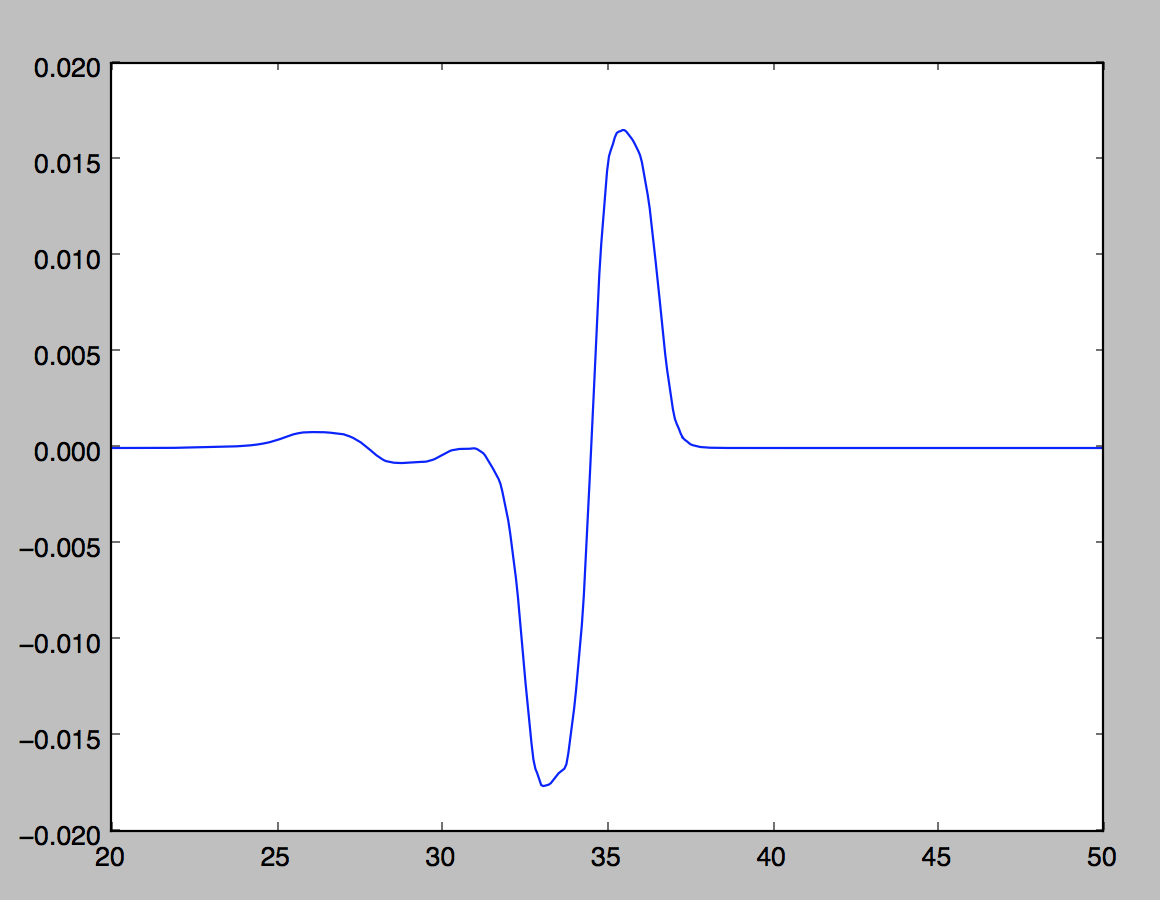
(Initial Condition) Density at Time = 0:



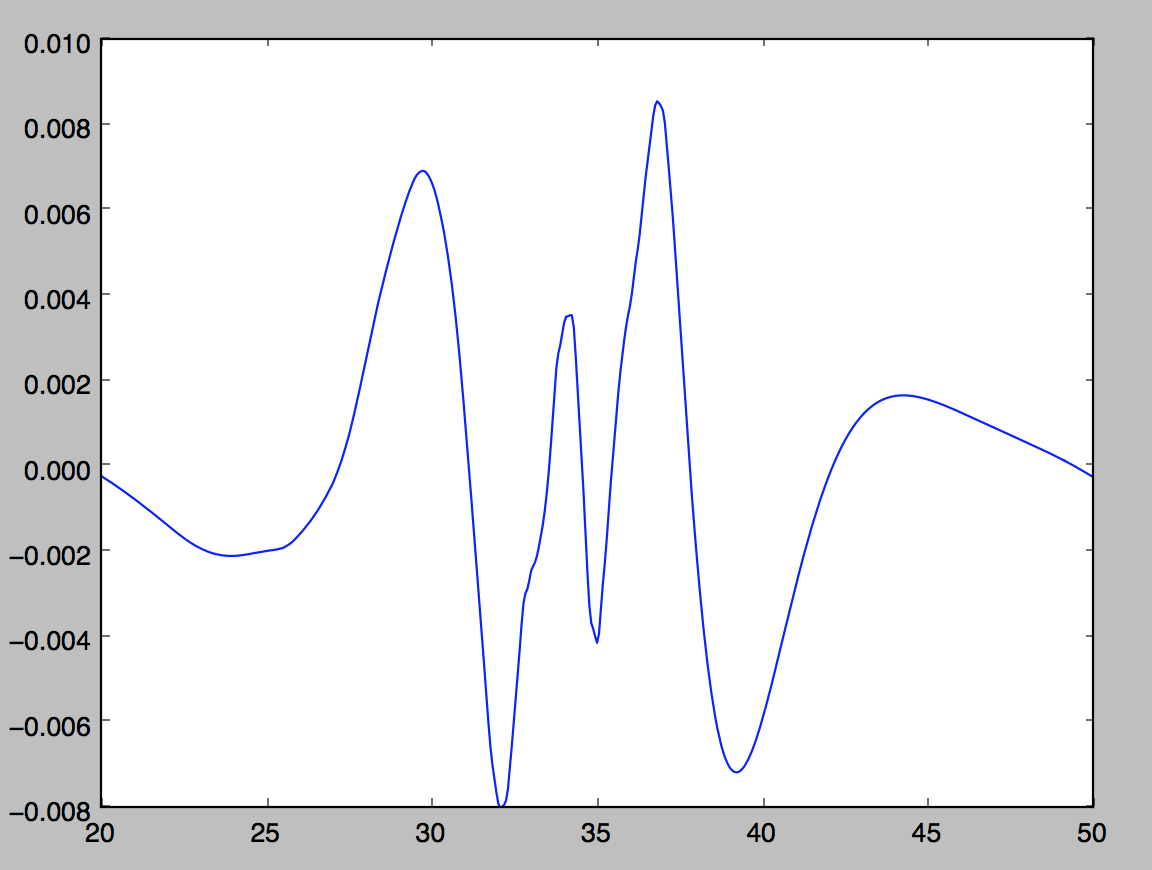
Density at Time = 2:



(Initial Condition) Velocity at Time = 0:



Velocity at Time = 2:



Analysis:

As we can see from testing the first two datasets, there is numerical diffusion that is distorting our solution. Let us attempt to fix this numerical diffusion. Often when we encounter numerical diffusion, we would do something with the second derivative to fix it. Let us add a length scale – bifurcation parameter to the following equation:

After is added:

That is, Alfa must be positive and way smaller than 1. Thus, we will try Alfa = .01

Another constant variable that we are able to change to help stabilize our solution is the gravity constant (g).

See attached file for data results.