Inverse Wave Propagation

Lava K.C.
Augsburg College
Minneapolis, MN 55454
kc@augsburg.edu

This work was supported by National Science Foundation grant RUI/CMG/ATM-0222148.

Abstract

Waves can be modeled mathematically and simulated using a computer in the forward direction with respect to time. Knowing various properties of the medium such as the density and the size, a source of a wave can be propagated through time. Using the existing forward direction wave simulation, we were able to find and simulate the original wave with very little information known about the wave itself.

1 Wave Propagation

1.1 What is Wave Propagation?

A wave may be defined as the propagation of disturbance carrying energy. We can model a wave mathematically by the following equation

$$\frac{\partial^2 \mathcal{U}}{\partial t^2} = \rho c^2 \left(\frac{\partial}{\partial x} \left(\frac{1}{\rho} \frac{\partial \mathcal{U}}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{1}{\rho} \frac{\partial \mathcal{U}}{\partial y} \right) \right)$$

where $\mathcal{U}$ is the pressure function of (x, y, t) x, y and t being the position and time respectively. ρ and c are the density of the medium and speed of the wave in the medium. There was an existing program that would simulate a wave in the forward direction. The program took in sets of parameters such as the length of time, the dimension of the space, the density of the field and the speed with which the wave propagated. The output could either be a 2 dimensional or 3 dimensional animation created by combining a sequence of jpeg images created in each time instance while the program was running. Each pixel on the screen represented one unit of dimensional space. The major problem in this simulation was what to do at the boundaries. The existing program simply reflected the wave when it hit the boundary.

Figure 1 illustrates how the propagated wave would look like after a certain period of time.

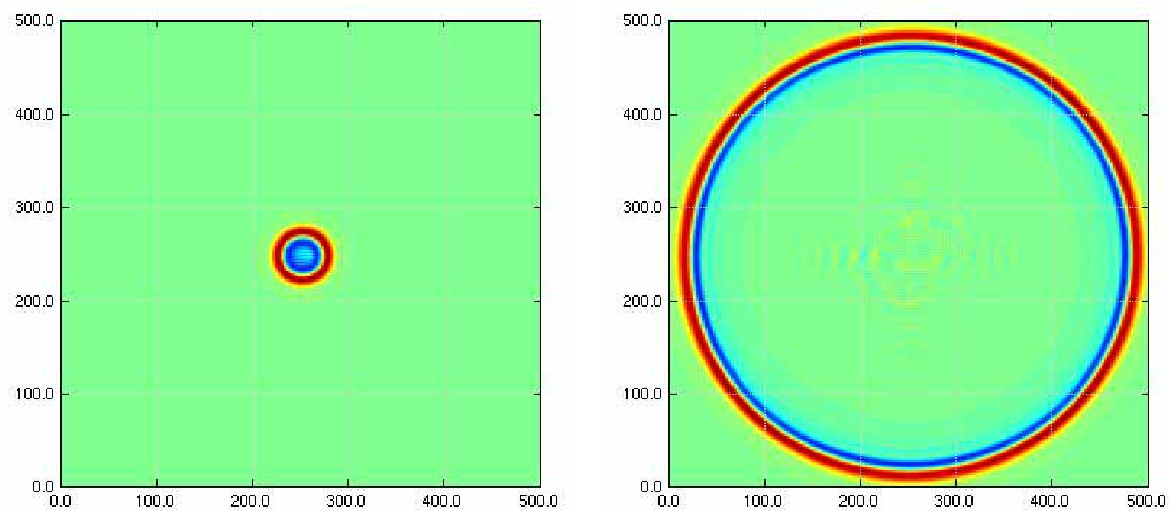

Figure 1: Initial wave (*left*) and wave after 50 time units (*right*).

1.2 What is Inverse Wave Propagation?

Inverse Wave Propagation is the process of trying to find the original wave that propagated in a medium while knowing very little information about the pressure the wave created.

For example : Suppose we record a voice in a normal room for a given time period with some type of pressure sensor or a microphone. Inverse wave propagation is the method of finding the properties of the medium in which the sound wave traveled. For our particular example, the speed of the sound in the medium after calculation should turn out to be close to the real speed of the sound in air and the density of the medium should be fairly equal to that of air.

Inverse Wave Propagation would be useful for scientists who are getting wave signals from space and want to find some properties of the medium in which the wave has traveled. It also can be used in underground explorations for gas and oil.

1.3 What does Inverse Wave Propagation Mean Mathematically?

Let the Forward Wave Propagation be represented by a function F(parameters) $\rightarrow \vec{d}$ where $\vec{d}$ is the final output. The parameters are the size, density of the medium, and the speed of the wave in the medium. Inverse Wave Propagation is the process of finding out the real wave with very little information about the wave.

To simplify the problem, it was decided that the density of the field would be assumed constant and both the density and size of the field is known. Now, the problem was simplified to estimate the speed of the wave. The speed field could be represented with a variable called $\vec{c}$. Using the equation F($\vec{c}$) $\rightarrow \vec{d}$, we can find $\vec{c}$ by simply solving the equation $F^{-1}(\vec{d}) \rightarrow \vec{c}$.

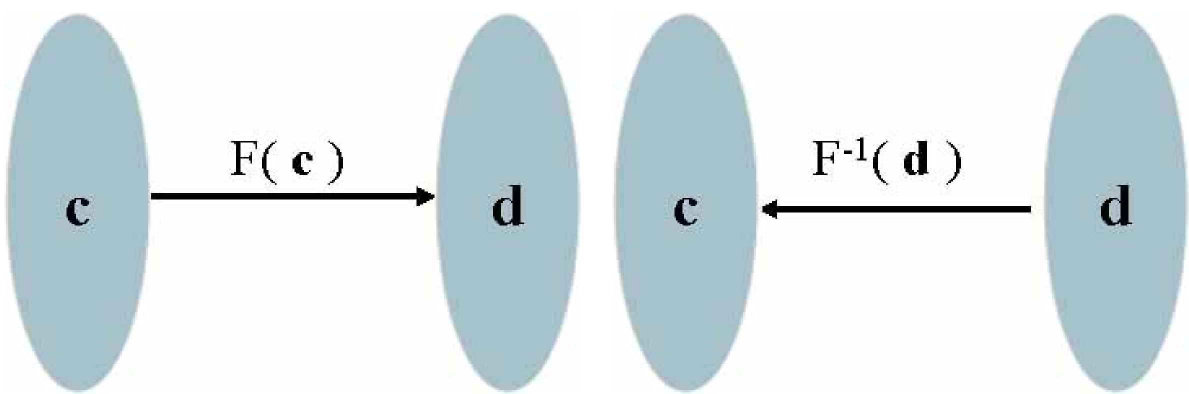

Figure 2: Figurative representation of the Forward (*left*) and Inverse (*right*) Wave Propagation.

Since F($\vec{c}$) did not have any closed form solution, the function F($\vec{c}$) was estimated using our Forward Wave Propagation program. Actually, even if F($\vec{c}$) had a formula the dimension of the parameter is too large to solve.

The concept of minimization can be used here. Instead of trying to find the inverse, we can try to minimize the misfit function. The misfit function may be represented as F($\vec{c}$)-$\vec{d}$ and is the difference between the actual pressure changes and the simulated pressure changes. We tried different methods to obtain the minimum misfit, as discussed below.

1.3.1 Monte-Carlo Gradient Method

In this process, the parameter $\vec{c}$ which is the speed of the wave in the medium was first randomly guessed. The forward wave was simulated, and the two data were compared. With these two sets of data we could calculate both F($\vec{c}$)-$\vec{d}$ and the gradient vector for $\vec{c}$. Following the direction of the gradient vector, we tried to minimize the misfit function.

The problem with this method was that the gradient vector step size was too small and the program could not complete the task within a day. Thus it was decided to implement a new method instead of the Monte-Carlo Gradient method.

1.3.2 Newton's Tangent Line Method for n dimension

This method is similar to the two-dimensional Newton Tangent Line method which is covered in Calculus. We first found the tangent vector for the misfit function and moved towards the direction of the tangent. The advantage of this method over the Monte-Carlo Gradient method was that this method selected the step size dynamically, a large step size at first and then small step-sizes when the misfit function was getting smaller. Newton's Tangent Line method was better than the Monte-Carlo Gradient method because it could complete a particular problem in a given time frame. Most of the time, the speed vector field $\vec{c}$ of the wave in the medium could be found which would produce the misfit function within 1% of the true value.

However, both the methods mentioned above are unreliable. This is because the Newton Tangent Line method and the Monte-Carlo Gradient method only give us a local maximum or a local minimum. We not only have to minimize the misfit function but also find the absolute minimum or absolute maximum to get a true value of $\vec{c}$. For this reason, we decided to use the Simulated Annealing method, which is described below.

1.3.3 Simulated Annealing Method

Simulated Annealing is a technique in which new solutions to the optimization problem are picked at random and checked for the least misfit. If the misfit turns out to be smaller than the smallest misfit we have observed thus far, the new solution is considered to be the best solution.

The pseudo code for this algorithm is as follows:

For j=1,n_{iter} do:

1. For k=1,2 do:
   (a) Compute $c^{\vec{new}}$
      i. Let *r* be a random number between 0 and 1
      ii. Let $p = sign(r - 0.5)T(1 + \frac{1}{T})^{(2r-1)}$

iii. Let $\vec{c_i}^{new} = c_i + p(\vec{c_i}^{max} - \vec{c_i}^{min})$

iv. if $\vec{c_i}^{new}$ does not satisfy $\vec{c_i}^{min} <= \vec{c_i} <= \vec{c_i}^{max}$ change it so it does

(b) If $G(\vec{c}^{new}) < G(\vec{c})$, then $\vec{c} = \vec{c}^{new}$ otherwise,

i. Let r be a random number between 0 and 1

ii. Let $dE = G(\vec{c}^{new}) - G(\vec{c})$

iii. Let $p = exp(-dE/T)$

iv. if $p > r$ then $\vec{c} = \vec{c}^{new}$

2. Let $T = T_0 exp(-C\sqrt{j})$

The input parameters are T_0, C, n_{iter} where T_0 is the initial temperature, C is the step size and n_{iter} is the amount of steps you want to apply in the annealing process. For our problem we selected n_{iter} to be between 500 and 1500.

The Simulated Annealing algorithm is a heuristic search algorithm that allows non-improving solutions to be accepted with decreasing probability over time. This method eliminates the problems caused by the previous methods because there is a chance of avoiding local minima or local maxima. This algorithm was relatively fast compared to the methods described previously. Figure 3 illustrates how fast this algorithm works.

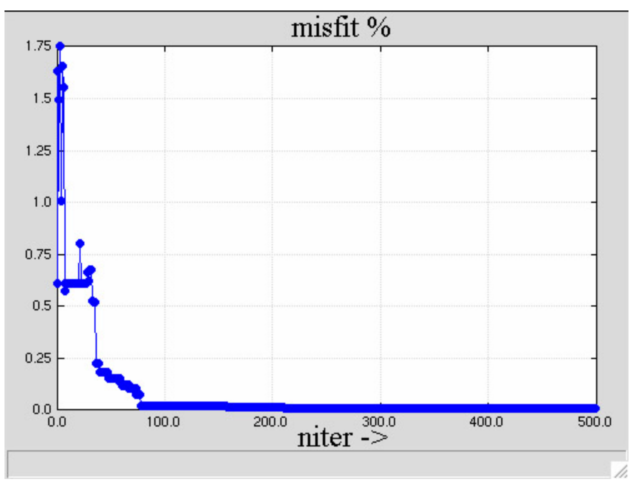

Figure 3: n_{iter} vs Misfit $\%$ comparison. Within 100 loops, the error was minimized to less than 1%.

1.4 Testing of The Program

To implement this program in the real world, we needed some test data with which to verify its accuracy. The sets of test data taken were from the Forward Wave Propagation program.

1.4.1 Calculating $\vec{d}$

A wave was propagated in a medium. There were sensors put in the program such that it recorded the pressure changes for some time.

1.4.2 Calculating $\vec{c}$

The recorded pressures, along with the dimension of the medium and the density vector, were transmitted to the Inverse Wave Propagation program. The Inverse Wave Propagation program tried to find the speed vector of the medium using the Simulated Annealing method.

1.4.3 Calculating F($\vec{c}$)-d

The new speed vector was fed to the Forward Wave Propagation software with the same sensing configuration. The new pressure changes were acquired. In other words, F($\vec{c}$) was estimated. This new data was then subtracted from the old data $\vec{d}$ and then the error was computed.

The results were very good. The error turned out to be less than 5% when compared to the original test data.

1.5 Did Changing the Type of Sensor Have an Effect on the Error?

In reality, we can not have a bar of sensor. We can have multiple sensors, for example microphones place at different places to record the pressure changes. So, instead of taking a single bar of sensors, we decided to take a set of point sensors as seen in the Figure 4. This allowed us the flexibility of making customized sensors. The set of points in the sensor set could be a line, a single point or a circle or simply four points on each side of the medium.

The initial assumption was that the error would decrease when more point sensors were used. This turned out to be false. Figure 5 will help in illustrating why this was happening.

1.6 Was the Error Computed a True Error Estimate?

The answer to this question is no. The error depended on the location of the sensor. For example, consider a scenario of a room where two types of medium fill it up. If the sensor was placed entirely in the side of the room with medium 1, our program did not guarantee the error of the other side of the room.

To fix this problem, we decided to place sensors on four sides of the room, instead of only one side. This reduced the speed error and misfit error can be seen on the Figure 7

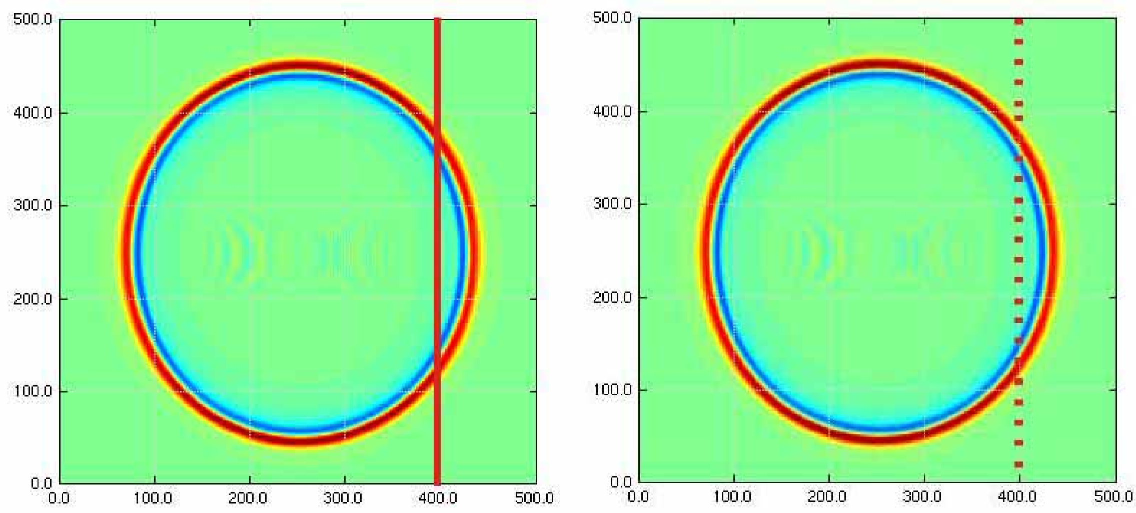

Figure 4: Two types of sensors. The single bar is shown on the left. The set of point sensors is shown on the right.

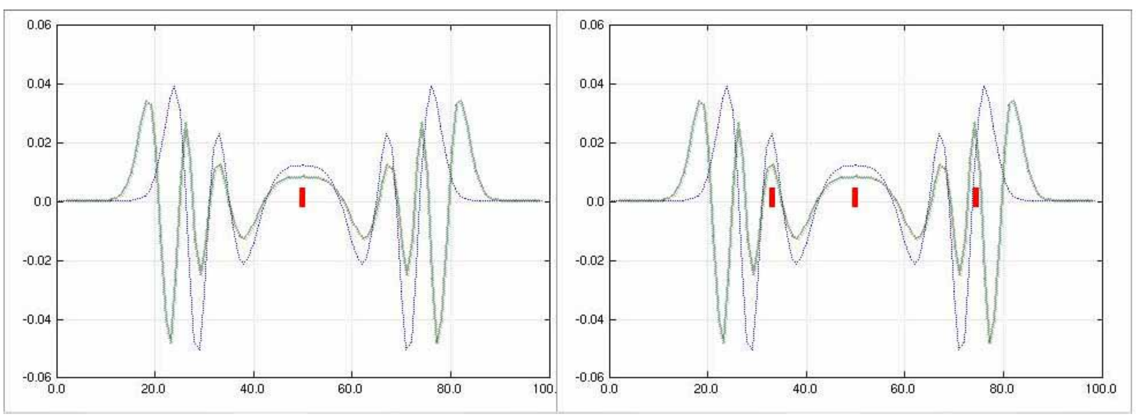

Figure 5: Using more sensors showed more errors because we were averaging errors from more places.

1.7 Conclusion:

After experimenting with different methods to solve the optimization problem, it was decided that the Simulated Annealing method was the best to use for Inverse Wave Propagation. One should also position the sensor on four sides to get the best possible estimate of the speed vector.

1.8 Acknowledgement:

I would like to thank Professor Nick Coult for help throughout this project.

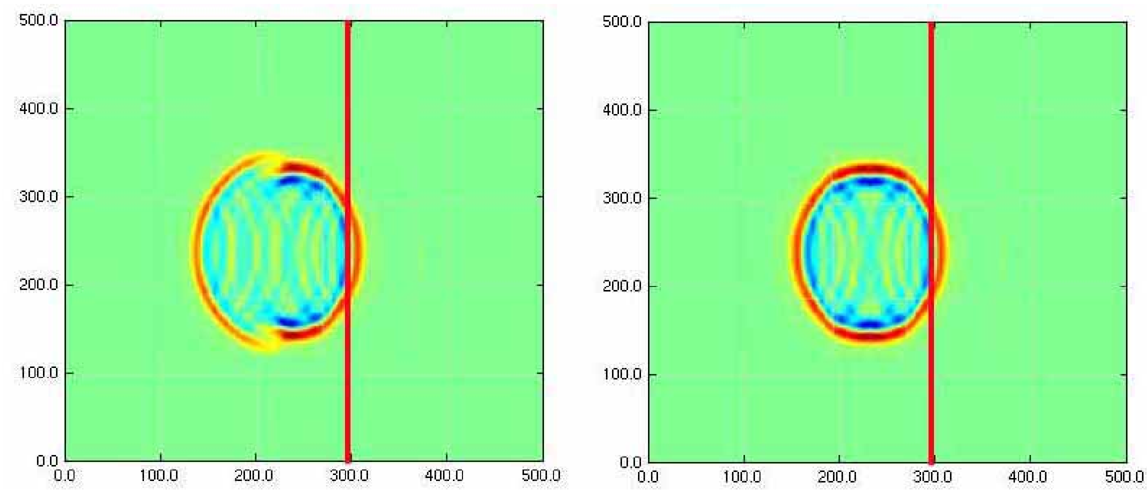

Figure 6: Miscalculation of error. Room with only one medium (*left*) and room with two types of medium (*right*). Due to the position of the sensor, the Simulated Annealing method only came up with one possible speed value instead of two.

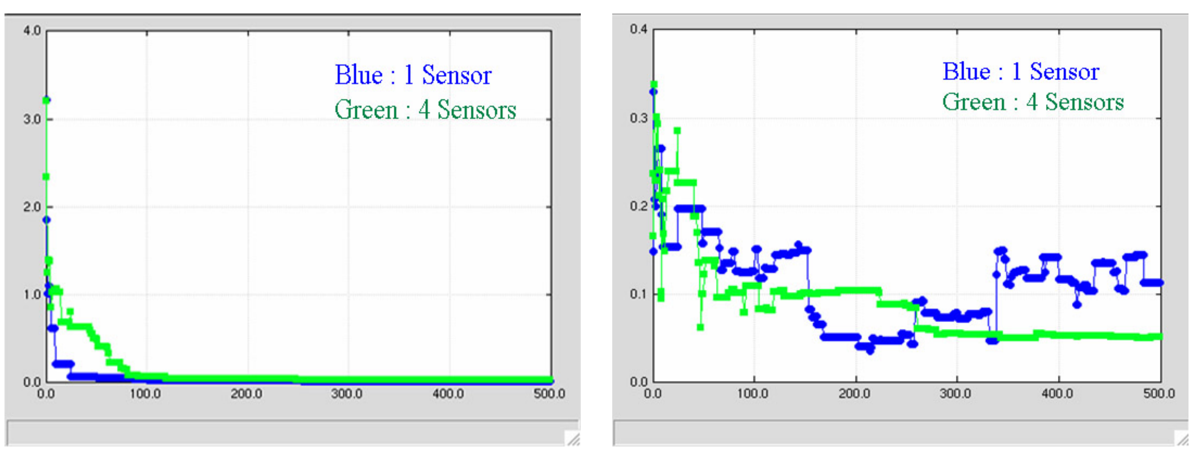

Figure 7: Comparison of the misfit error *(left)* and the velocity error *(right)* when using sensors on 1 side and 4 sides. Even though the misfit error graph may suggest that our $\vec{c}$ was correct, the velocity error graph suggests that we have large velocity error.