Finite Difference Analysis of 2-Dimensional Acoustic Wave with a Signal Function

Opiyo Richard Otieno [1], Alfred Manyonge [1], Owino Maurice [2] & Ochieng Daniel [1]
richardopiyo08@gmail.com[1],wmanyonge@gmail.com[1] & mauricearaka@yahoo.com[2]
[1] Dept of Pure & Applied Mathematics
Maseno University(Kenya)
[2] Dept of Mathematics & Computer Sciences
University of Kabianga (Kenya)

December 9, 2015

Abstract

This paper describes progress on a two dimensional numerical simulation of acoustic wave propagation that has been developed to visualize the propagation of acoustic wave fronts and to provide time-domain signal. In this exercise, we have simulated propagation of sound in such a medium using both explicit and Crank Nicolson finite difference schemes, we have also tested for stability of the developed schemes using Vonn Newmann and Matrix stability analysis together with its associated code in matlab. The stability analyses of the developed schemes revealed that Explicit scheme was conditionally stable while the Hybrid one (Crank Nicolson Scheme) was unconditionally stable, for all values of courant number r.
The rate of convergence of the algorithms depend on the truncation error introduced when approximating the partial derivatives, the Crank-Nicolson method converged at the rate of $(k^2 + h^2)$, which is a faster rate of convergence than either the explicit method, or the implicit method.

Keywords:

Acoustic wave, Finite difference approximation, Signal function, Crank Nicolson, Vonn Newman, Matrix stability analysis.

1 Introduction

When determining the acoustic properties of an environment, we are actually interested in the propagation of sound, given the properties and location of a sound source. Propagation of light or sound wave is of long standing interest in several branches of basic and applied physics, from old disciplines such as x-ray diffraction in crystallography, to the modern science of photonic crystals. Many problems in natural environment so involve wave propagation in periodic media. For example, nearly periodic sand bars are frequently found in shallow seas outside the surf zone; their presence changes the wave climate near the coast. The technology of remote-sensing, either by underwater sound or by radio waves from a satellite, depends on our understanding of scattering by the wavy sea surface.
Finite difference method is a key tool in numerical analysis and the motivation to study and learn this method is the fact that in Fluid dynamics, thermodynamics, solid mechanics etc. a large number of differential equations are found. And to solve all of them analytically is very difficult and at times impossible. As a result Finite Difference Methods provide sufficiently satisfactory accurate numerical solutions to such equations. Finite-difference modelling of wave propagation in heterogeneous media is a useful technique in a number of disciplines, including seismology and ocean acoustics. Sound is a longitudinal wave that is, waves of alternating pressure deviations from equilibrium causing local regions of compression and rarefaction as a result of vibrating objects. Sound is a wave which can be described as a disturbance that travels through a medium, transporting energy from one location to another location.
Many researchers have developed numerical interpretations of the wave equation suited to acoustics and seismic propagation. Hugh and Pat [13], developed second order finite difference scheme for modelling the acoustic wave equation in Matlab but their major limitation was, insufficient consideration of boundary conditions. Alford, Kelly and Boore [2], proposed that acoustic wave equation for homogeneous media can be approximated in rectangular co-ordinate system by the second and fourth order central difference. Although, one-way wave equation method in inhomogeneous media has been extensively studied in the literature, few detailed studies have been made on the implementation of source term and free boundary conditions. For this reason, Xie and Wu [29] integrated free surface boundary condition and the source term for one way elastic waves for decomposition of plane wave.
Charara and Tarantola [7], in their publication con-

sidered boundary conditions and source term for one-way acoustic depth extrapolation and they used a number of finite difference schemes and techniques namely, implicit finite difference scheme, central finite difference schemes and splitting methods. Seongjai [24], came up with fourth order implicit time stepping scheme for numerical solution of the acoustic wave equation as a variant of the conventional modified equation method, the scheme incorporated a locally one-dimensional (LOD) procedure with splitting error of $O(\Delta t^4)$. Walstijn and Kowalczyk [19], focused on compact stencil finite difference time domain (FDTD) scheme for approximating 2D wave equation in the context of digital audio.
This present work is a finite difference analysis of two dimensional acoustic wave equation with a signal function. Further, Von Neumann and matrix stability analyses criterion is done.

1.1 Finite Difference Method

The mathematical modelling of practical problems often involves the use of Partial Differential Equations. Very few of these equations can be solved analytically. For the acoustic wave equation described by a Partial Differential Equation , analytical solutions do exist but only for special or simple cases like the homogeneous case. However, for complex or sufficiently realistic models, it is necessary to resort to numerical methods.
The finite difference method is one of several techniques for obtaining numerical solutions to practical problems governed by Partial Differential Equations (PDE). In all numerical solutions the continuous partial differential equation (PDE) is replaced with a discrete approximation. In this context, the word *discrete* means that the numerical solution is known only at a finite number of points in the physical domain. The number of those points can be selected by the user of the numerical method. In general, increasing the number of points not only increases the resolution, but also the accuracy of the numerical solution. The discrete approximation results in a set of algebraic equations that are evaluated (or solved) for the values of the discrete unknowns. Figure 1 is a schematic representation of the numerical solution. The mesh is the set of locations where the discrete solution is computed. These points are called nodes, and if one were to draw lines between adjacent nodes in the domain the resulting image would resemble a net or mesh. Two key parameters of the mesh are $\Delta x \& \Delta z$, the local distance between adjacent points in space, and Δt, the local distance between adjacent time steps. For the case considered in this article Δx and Δz are uniform throughout the mesh. The core idea of the finite difference method is to replace continuous derivatives with difference formulas that involve only the discrete values associated with positions on the mesh. Applying the finite difference method to a differential equation involves replacing all derivatives with difference formulas. In the wave equation there are derivatives with respect to time, and derivatives with respect to space. Using different combinations of mesh points in the difference formulas results in different schemes. In the limit as the mesh spacing $(\Delta x, \Delta z)$ and (Δt) go to zero, the numerical solution obtained with any useful scheme will approach the true solution to the original differential equation. However, the rate at which the numerical solution approaches the true solution varies with the scheme. In addition, there are some practically useful schemes that can fail to yield a solution for bad combinations of $\Delta x, \Delta z$ and Δt.

1.2 Discretization Procedure

In developing the schemes, computational domain Ω is discretized with uniform grid with assumption that with uniform grid, both the space and time are adequate for the solution, it implies that $(\Delta x = \Delta z = h)$. Dividing the domain into a grid of Nx by Nz points, where Δx and Δz are the distance between points in the grid in the x and z axes respectively, to yield $x = n_x\Delta x$ and $z = n_z\Delta z$, where $n_x = 1, 2, ..., Nx$ and $n_z = 1, 2, ..., Nz$. Also, if Δt is the increment in time, then $t = k\Delta t$ where k is the time step with $k = 1, 2, \cdots, n$. Denoting the discrete approximation of $u(x, z, t)$ at the grid point (different points in space and time) as $(x_i = i\Delta x, z_j = j\Delta z, t^n = n\Delta t)$, then the acoustic wave field (numerical solution) can be specified as $u(x, z, t) \approx u^n_{i,j} = u(ih, jh, nk)$, for all $i = 1, 2, 3, \cdots, n_x$, $j = 1, 2, 3, \cdots, n_z$ and $n = 0, 1, 2, \cdots$

1.3 Finite Difference Approximations

Finite difference formulas are first developed with the dependent variable ϕ as a function of only one independent variable, x, i.e. $\phi = \phi(x)$. The resulting formulas are then used to approximate derivatives with respect to either space or time. By initially working with $\phi = \phi(x)$, the notation is simplified without any loss of generality in the result.

1.3.1 First Order Forward Difference

Consider a Taylor series expansion $\phi(x)$ about the point x_i

$$\phi(x+\Delta x) = \phi(x_i)+\Delta x\frac{\partial\phi}{\partial x}|_{x_i}+\frac{\Delta x^2}{2}\frac{\partial^2\phi}{\partial x^2}|_{x_i}+\frac{\Delta x^3}{3!}\frac{\partial^3\phi}{\partial x^3}|_{x_i}+\ldots \quad (1)$$

where Δx is a change in x relative to x_i. Solving for $\left(\frac{\partial\phi}{\partial x}\right)_{x_i}$ yields

$$\frac{\partial\phi}{\partial x}|_{x_i} = \frac{\phi(x+\Delta x)-\phi(x_i)}{\Delta x}-\frac{\Delta x}{2}\frac{\partial^2\phi}{\partial x^2}|_{x_i}-\frac{\Delta x^2}{3!}\frac{\partial^3\phi}{\partial x^3}|_{x_i}+\ldots \quad (2)$$

Notice that the powers of Δx multiplying the partial derivatives on the right hand side have been reduced by one. Let the approximate solution for the exact solution, i.e. $\phi_i \approx \phi(x_i)$ and $\phi_{i+1} \approx \phi(x_i + \Delta x)$, then equation (2) becomes;

$$\frac{\partial \phi}{\partial x}|_{x_i} \approx \frac{\phi_{i+1}) - \phi_i}{\Delta x} - \frac{\Delta x}{2}\frac{\partial^2 \phi}{\partial x^2}|_{x_i} - \frac{\Delta x^2}{3!}\frac{\partial^3 \phi}{\partial x^3}|_{x_i} + \dots \quad (3)$$

From the mean value theorem we can have for higher order derivatives

$$\frac{\Delta x^2}{2}\frac{\partial^2 \phi}{\partial x^2}|_{x_i} + \frac{(\Delta x)^3}{3!}\frac{\partial^3 \phi}{\partial x^3}|_{x_i} + \dots = \frac{\Delta x^2}{2}\frac{\partial^2 \phi}{\partial x^2}|_{\epsilon} \quad (4)$$

where $x_i \leq \epsilon \leq x_{i+1}$, therefore

$$\frac{\partial \phi}{\partial x}|_{x_i} \approx \frac{\phi_{i+1}) - \phi_i}{\Delta x} + \frac{\Delta x^2}{2}\frac{\partial^2 \phi}{\partial x^2}|_{\epsilon}$$

or equivalently;

$$\frac{\partial \phi}{\partial x}|_{x_i} - \frac{\phi_{i+1}) - \phi_i}{\Delta x} \approx \frac{\Delta x^2}{2}\frac{\partial^2 \phi}{\partial x^2}|_{\epsilon} \quad (5)$$

The term on the right hand side of Equation (5) is called the truncation error of the finite difference approximation. It is the error that results from truncating the series in Equation (3).
In general, notice that ϵ is not known. Furthermore, since the function $\phi(x,t)$ is also unknown, $\frac{\partial^2 \phi}{\partial x^2}$ cannot be computed. We apply the big O notation to express the dependence of the truncation error on the mesh spacing. Note that the right hand side of Equation (5) contain the mesh parameter Δx, which is chosen by the person using the finite difference simulation. Since this is the only parameter under the user's control that determines the error, the truncation error is simply written

$$\frac{\Delta x^2}{2}\frac{\partial^2 \phi}{\partial x^2}|_{\epsilon} = O(\Delta x^2)$$

The equals sign in this expression is true in the order of magnitude sense. In other words its not a strict equality, but rather, means that the left hand side is a product of an unknown constant and Δx^2. Although the expression does not give us the exact magnitude of $\frac{\Delta x^2}{2}((\frac{\partial^2 \phi}{\partial x^2})x_i)\epsilon$, it tells us how quickly that term approaches zero as Δx is reduced.
Using big O notation, Equation (3) can be written

$$\frac{\partial \phi}{\partial x}|_{x_i} = \frac{\phi_{i+1}) - \phi_i}{\Delta x} + O(\Delta x) \quad (6)$$

Equation (6) is called the *forward difference formula* for $\frac{\partial \phi}{\partial x}_{x_i}$ since it involves nodes x_i and x_{i+1}, hence, forward difference approximation has a truncation error that is $O(\Delta x)$. The size of the truncation error is (mostly) under our control because we can choose the mesh size Δx. The part of the truncation error that is not under our control is $\frac{\partial \phi}{\partial x}|_{\epsilon}$.

1.3.2 First Order Backward Difference

An alternative first order finite difference formula is obtained if the Taylor series like that in Equation (1) is written with a backward shift ($-\Delta x$). Using the discrete mesh variables in place of all the unknowns, one obtains

$$\phi_{i-1} = \phi_i - \Delta x\frac{\partial \phi}{\partial x}|_{x_i} + \frac{\Delta x^2}{2}\frac{\partial^2 \phi}{\partial x^2}|_{x_i} - \frac{(\Delta x)^3}{3!}\frac{\partial^3 \phi}{\partial x^3}|_{x_i} + \dots$$

Notice in this case the alternating signs of terms on the right hand side. Solving for $\frac{\partial \phi}{\partial x}|_{x_i}$, we arrive at

$$\frac{\partial \phi}{\partial x}|_{x_i} = \frac{\phi_i - \phi_{i-1}}{\Delta x} + \frac{\Delta x}{2}\frac{\partial^2 \phi}{\partial x^2}|_{x_i} - \frac{(\Delta x)^2}{3!}\frac{\partial^3 \phi}{\partial x^3}|_{x_i} + \dots$$

On using big O notation we get

$$\frac{\partial \phi}{\partial x}|_{x_i} = \frac{\phi_i - \phi_{i-1}}{\Delta x} + O(\Delta x) \quad (7)$$

This is called the *backward difference formula* because it involves the values of ϕ at x_i and x_{i-1}.
The order of magnitude of the truncation error for the backward difference approximation is the same as that of the forward difference approximation.

1.3.3 First Order Central Difference

Consider the Taylor series expansions for ϕ_{i+1} and ϕ_{i-1} as below;

$$\phi_{i+1} = \phi_i + \Delta x\frac{\partial \phi}{\partial x}|_{x_i} + \frac{\Delta x^2}{2}\frac{\partial^2 \phi}{\partial x^2}|_{x_i} + \frac{\Delta x^3}{3!}\frac{\partial^3 \phi}{\partial x^3}|_{x_i} + \dots \quad (8)$$

$$\phi_{i-1} = \phi_i - \Delta x\frac{\partial \phi}{\partial x}|_{x_i} + \frac{\Delta x^2}{2}\frac{\partial^2 \phi}{\partial x^2}|_{x_i} - \frac{(\Delta x)^3}{3!}\frac{\partial^3 \phi}{\partial x^3}|_{x_i} + \dots \quad (9)$$

Subtracting Equation (9) from Equation (8) yields

$$\phi_{i+1} - \phi_{i-1} = 2\Delta x\frac{\partial \phi}{\partial x}|_{x_i} + 2\frac{(\Delta x)^3}{3!}\frac{\partial^3 \phi}{\partial x^3}|_{x_i} \dots$$

Solving for $(\frac{\partial \phi}{\partial x})_{x_i}$ gives

$$\frac{\partial \phi}{\partial x}|_{x_i} = \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} - \frac{\Delta x^2}{3!}\frac{\partial^3 \phi}{\partial x^3}|_{x_i} + \dots$$

which results to;

$$\frac{\partial \phi}{\partial x}|_{x_i} = \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} + O(\Delta x^2) \quad (10)$$

which is the *central difference approximation* to $(\frac{\partial \phi}{\partial x})_{x_i}$.
To get good approximations to the continuous problem generally, small Δx is chosen. When $\Delta x << 1$, the truncation error for the central difference approximation goes to zero much faster than the truncation error in Equation (6) or Equation (7).

1.3.4 Second Order Central Difference

Finite difference approximations to higher order derivatives can be obtained with the additional manipulations of the Taylor Series expansion about $\phi(x_i)$. Adding Equation (9) and Equation (8) yields

$$\phi_{i+1} + \phi_{i-1} = 2\phi_i + (\Delta x)^2\frac{\partial^2 \phi}{\partial x^2}|_{x_i} + \frac{2(\Delta x)^4}{4!}\frac{\partial^4 \phi}{\partial x^4}|_{x_i} + \dots$$

Solving for $\left(\frac{\partial^2\phi}{\partial x^2}\right)_{x_i}$ gives;

$$\frac{\partial^2\phi}{\partial x^2}|_{x_i} = \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta x^2} - \frac{(\Delta x)^2}{12}\frac{\partial^4\phi}{\partial x^4}|_{x_i} + \dots$$

Using order notation

$$\frac{\partial^2\phi}{\partial x^2}|_{x_i} = \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta x^2} + O(\Delta x^2) \quad (11)$$

This is also called the *central difference approximation*, to the second derivative, whereas Equation (11) is the central difference approximation to the first derivative

1.4 Discretizing the acoustic equation

Generally in mathematical approach, the continuous formulation is transformed to a discrete formulation by replacing derivatives by say finite difference approximations while discretizing. The idea is to discretize the problem by choosing a step size h in both x and z and a step size k in t as in the solution procedure above. Then we try to approximate the acoustic potential (pressure) u on a grid of points. Therefore, we replace the continuous problem domain by a grid, or mesh, of discrete locations on Ω.

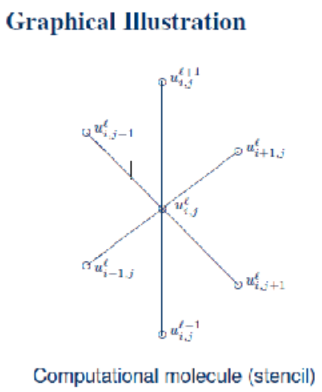


Figure 1.1: Computational molecule (stencil) in (x,z,t) space

The figure below clearly shows schematic representation of 2D (x,z,t) operator for discrete domain :

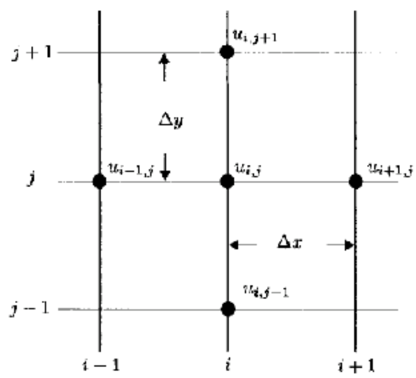


Figure 1.2: Representation in Grid (stencil) in (x,z,t) space

2 Numerical schemes

In this section, we develop the two numerical schemes that we shall use in this study, that is Central Difference Scheme (explicit) and Crank-Nicolson schemes (Hybrid) for the model equation

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial z^2}-\frac{1}{c^2(x,z)}\frac{\partial^2 u}{\partial t^2}=S(x,z,t)$$

which is a hyperbolic PDE, therefore we first discretize this equation by using the central difference approximation to the second derivative in u_{xx}, u_{zz} and u_{tt}

2.1 Central Difference Scheme(CDS) (Explicit)

Construction of the simple explicit scheme for the homogeneous 2-dimensional acoustic wave equation in rectangular coordinate is a fairly straight forward matter. Namely;

$$u_{tt}=c^2\left(u_{xx}+u_{zz}\right), \tag{2.1}$$

where $S=0$ which means that there is no supply of energy from the source. To develop explicit scheme for this equation, we discretize the terms in the homogeneous equation governed by (2.1) in the standard way by defining the central difference operators as follows:

$$D_x^2 u_{i,j}^n=\frac{u_{i+1,j}^n-2u_{i,j}^n+u_{i-1,j}^n}{(\Delta x)^2}$$

$$D_z^2 u_{i,j}^n=\frac{u_{i,j+1}^n-2u_{i,j}^n+u_{i,j-1}^n}{(\Delta z)^2}$$

Substituting these operators in equation (2.1), we arrive at:

$$u_{i,j}^{n+1}=2u_{i,j}^n-u_{i,j}^{n-1}+c^2k^2D_x^2u_{i,j}^n+c^2k^2D_z^2u_{i,j}^n$$

Systematic substitution yields;

$$\frac{u_{i,j}^{n+1}-2u_{i,j}^n+u_{i,j}^{n-1}}{(\Delta t)^2}=\frac{c^2}{(\Delta x)^2}\left(u_{i+1,j}^n-2u_{i,j}^n+u_{i-1,j}^n\right)$$

$$+\frac{c^2}{(\Delta z)^2}\left(u_{i,j+1}^n-2u_{i,j}^n+u_{i,j-1}^n\right). \tag{2.2}$$

We then express $u_{i,j}^{n+1}$ in terms of other terms to give;

$$u_{i,j}^{n+1}=2u_{i,j}^n-u_{i,j}^{n-1}+\frac{c^2(\Delta t)^2}{(\Delta x)^2}\left(u_{i+1,j}^n-2u_{i,j}^n+u_{i-1,j}^n\right)+$$

$$c^2(\Delta t)^2(\Delta z)^2\left(u_{i,j+1}^n-2u_{i,j}^n+u_{i,j-1}^n\right) \tag{2.3}$$

subscripts, i,j and superscript n represent the x,z and time co-ordinates respectively for a discrete grid of uniform spacing that is $\Delta x=\Delta z$ and for convenience, we introduce the substitution $\sigma=\left(\frac{c\Delta t}{\Delta x}\right)^2$, this yields;

$$u_{i,j}^{n+1}=2u_{i,j}^n-u_{i,j}^{n-1}+\sigma\left(u_{i+1,j}^n-2u_{i,j}^n+u_{i-1,j}^n\right)+$$

$$\sigma\left(u_{i,j+1}^n-2u_{i,j}^n+u_{i,j-1}^n\right). \tag{2.4}$$

therefore,

$$u_{i,j}^{n+1}=(2-4\sigma)\,u_{i,j}^n+\sigma u_{i+1,j}^n+\sigma u_{i-1,j}^n+$$

$$\sigma u_{i,j+1}^n+\sigma u_{i,j-1}^n-u_{i,j}^{n-1}. \tag{2.5}$$

Using the same reasoning we can extend this concept to non-homogeneous case below

$$u_{xx}+u_{zz}-\frac{1}{c^2(x,z)}u_{tt}=S(x,z,t)$$

as

$$\frac{c_{i,j}^2}{(\Delta x)^2}\left(u_{i+1,j}^n-2u_{i,j}^n+u_{i-1,j}^n\right)+$$

$$\frac{c_{i,j}^2}{(\Delta z)^2}\left(u_{i,j+1}^n-2u_{i,j}^n+u_{i,j-1}^n\right)-$$

$$\left(\frac{u_{i,j}^{n+1}-2u_{i,j}^n+u_{i,j}^{n-1}}{(\Delta t)^2}\right)=c_{i,j}^2S_{i,j}^n \tag{2.6}$$

Again by letting $\sigma=\left(\frac{c_{i,j}\Delta t}{\Delta x}\right)^2$ and subscripts, i,j and superscript n to represent the x,z and time coordinates respectively for a discrete grid of uniform spacing that is $\Delta x=\Delta z$ then, collecting the unknown terms that is $u_{i,j}^{n+1}$ on the left hand side gives;

$$u_{i,j}^{n+1}=(2-4\sigma)\,u_{i,j}^n+$$

$$\sigma\left(u_{i+1,j}^n+u_{i-1,j}^n+u_{i,j+1}^n+u_{i,j-1}^n\right)-u_{i,j}^{n-1}-$$

$$c_{i,j}^2\Delta t^2S_{i,j}^n, \tag{2.7}$$

which is the explicit scheme for the two dimensional acoustic wave with source term for all
$i=1,2,3,\ldots,M-1;\quad j=1,2,3,\cdots,N-1.$

2.2 Crank - Nicolson scheme

In Crank-Nicolson scheme, we replace the spatial coordinates u_{xx} and u_{zz} by the average of each central difference approximations at n^{th} time level and at $(n+1)^{th}$ time level. These yields in (1.3.2) as

$$\frac{c_{i,j}^2}{2(\Delta x)^2}\left[\left(u_{i+1,j}^{n+1}-2u_{i,j}^{n+1}+u_{i-1,j}^{n+1}\right)\right]+$$

$$\frac{c_{i,j}^2}{2(\Delta x)^2}\left[\left(u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n}\right)\right]+$$

$$\frac{c_{i,j}^2}{2(\Delta z)^2}\left[\left(u_{i,j+1}^{n+1}-2u_{i,j}^{n+1}+u_{i,j-1}^{n+1}\right)\right]+$$

$$\frac{c_{i,j}^2}{2(\Delta z)^2}\left[\left(u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n}\right)\right]-$$

$$\left(\frac{u_{i,j}^{n+1}-2u_{i,j}^n+u_{i,j}^{n-1}}{k^2}\right)=c_{i,j}^2\frac{\left(S_{i,j}^n+S_{i,j}^{n+1}\right)}{2} \tag{2.8}$$

In order to reduce the computing time, we adopt uniform grid spacing that is $\Delta x = \Delta z = h$ and $\Delta t = k$, now letting $r = \frac{c_{i,j}^2 \Delta t^2}{2\Delta x^2} = \frac{c_{i,j}^2 k^2}{2h^2}$ to give

$$\frac{c_{i,j}^2 k^2}{2h^2}\left[\left(u_{i+1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i-1,j}^{n+1}\right) + \left(u_{i,j+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j-1}^{n+1}\right)\right] + \frac{c_{i,j}^2 k^2}{2h^2}\left[\left(u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}\right) + \left(u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}\right)\right] - u_{i,j}^{n+1} + 2u_{i,j}^{n} - u_{i,j}^{n-1} = c_{i,j}^2 k^2 \frac{\left(S_{i,j}^n + S_{i,j}^{n+1}\right)}{2} \quad (2.9)$$

on collecting unknown terms $u_{i,j}^{n+1}$ on the left hand side gives the Implicit Crank-Nicolson scheme

$$-ru_{i+1,j}^{n+1} + (1+4r)u_{i,j}^{n+1} - ru_{i-1,j}^{n+1} - ru_{i,j+1}^{n+1} - ru_{i,j-1}^{n+1} = ru_{i+1,j}^{n} + (2-4r)u_{i,j}^{n} + ru_{i-1,j}^{n} + ru_{i,j+1}^{n} + ru_{i,j-1}^{n} - u_{i,j}^{n-1} - c_{i,j}^2 k^2 \frac{\left(S_{i,j}^n + S_{i,j}^{n+1}\right)}{2}, \quad (2.10)$$

for all
$i = 1, 2, 3, \cdots, M-1$ and $j = 1, 2, 3, \cdots, N-1$
Taking S to be a space function of x and z but not a function of time(t), then $S_{i,j}^n = S_{i,j}^{n+1}$, our Implicit Crank-Nicolson equation reduces to

$$-ru_{i+1,j}^{n+1} + (1+4r)u_{i,j}^{n+1} - ru_{i-1,j}^{n+1} - ru_{i,j+1}^{n+1} - ru_{i,j-1}^{n+1} = ru_{i+1,j}^{n} + (2-4r)u_{i,j}^{n} + ru_{i-1,j}^{n} + ru_{i,j+1}^{n} + ru_{i,j-1}^{n} - u_{i,j}^{n-1} - c_{i,j}^2 k^2 S_{i,j}^n \quad (2.11)$$

for all $i = 1, 2, 3, \cdots, M-1$; and $j = 1, 2, 3, \cdots, N-1$

3 Results

Accuracy and Stability analysis

3.1 Matrix stability of Explicit scheme

Matrix stability method considers the finite difference representation of both the PDE and boundary condition in a matrix form for which eigenvalue analysis is used to study stability, the theory behind this method is that the modulus of the eigenvalues of the amplification matrix should be less than unity. Employing matrix method to analyze stability of the scheme (2.7)and expanding this scheme by taking $i = 1, 2, 3, \ldots, M-1; \quad j = 1, 2, 3, \ldots, N-1$, and $r = \sigma = (\frac{c_{i,j}\Delta t}{\Delta x})^2$, generates the system of equations (see appendix) which can be expressed in matrix form as

$$U_{i,j}^{n+1} = AU_{i,j}^{n} - U_{i,j}^{n-1} + \mathbf{b},$$

where

$$u_{i,j}^{n+1} = \begin{bmatrix} u_{1,1} \\ u_{2,1} \\ \vdots \\ u_{1,2} \\ \vdots \\ u_{M-1,N-1} \end{bmatrix}^{n+1}$$

$$A = \begin{pmatrix} (2-4r) & r & \cdots & r & \\ r & (2-4r) & r & \cdots & r \\ \vdots & \ddots & \ddots & \cdots & \vdots \\ \cdots & r & \cdots & r & (2-4r) \end{pmatrix}$$

$$u_{i,j}^{n} = \begin{bmatrix} u_{1,1} \\ u_{2,1} \\ \vdots \\ u_{1,2} \\ \vdots \\ u_{M-1,N-1} \end{bmatrix}^{n}$$

$$b = \begin{pmatrix} ru_{0,1}^n + ru_{1,0}^n + ru_{0,1}^n - c_{1,1}^2 k^2 S_{1,1}^n \\ ru_{2,0}^n - c_{2,1}^2 k^2 S_{2,1}^n \\ \vdots \end{pmatrix}$$

and

$$u_{i,j}^{n-1} = \begin{bmatrix} u_{1,1} \\ u_{2,1} \\ \vdots \\ u_{1,2} \\ \vdots \\ u_{M-1,N-1} \end{bmatrix}^{n-1}$$

We realise some pattern and the resulting matrix $[(M-1) \times (N-1)] \times [(M-1) \times (N-1)]$ is of *block-tridiagonal* form as

$$G = \begin{bmatrix} C & D & & & \\ B & C & D & & \\ & \ddots & \ddots & \ddots & \\ & & & B & C, \end{bmatrix}$$

where B, C and D are $(M-1) \times (M-1)$ matrices, and there are N such C matrices on the diagonal. For this case, B and D are diagonal matrices whereas C is tridiagonal,

$$C = \begin{bmatrix} (2-4r) & r & & & \\ r & (2-4r) & r & & \\ & \ddots & \ddots & \ddots & \\ & & & r & (2-4r) \end{bmatrix},$$

$$B = D = \begin{bmatrix} r & & & \\ & r & & \\ & & \ddots & \\ & & & r \end{bmatrix}$$

Since C is tridiagonal matrix which is symmetric positive definite and is diagonally dominant, then C is non-singular thus there is a unique solution. The symmetry then implies that we have both a necessary condition for stability, therefore this scheme will always be stable for restricted values of r.

3.2 Von Neumann stability of Explicit scheme (CDS)

The von Neumann stability analysis is a way to determine when a particular numerical method is stable.

It looks at solutions of the form $a_j^n = \xi^n e^{ijkh}$, where $i = \sqrt{-1}$, j is our spatial index, k is the time index, and h is the spatial step. To do the analysis using this method, we simply substitute the above solution into the discretized form of the numerical method and determine where $|\xi|^2 \leq 1$. This tells us whether the amplitude of the wave is less than or equal to one. If the amplitude is greater than one, then the amplitude is increasing and will therefore eventually become unstable. Thus the method is stable at the values where $|\xi|^2 \leq 1$. In general, the Von Neumanns procedure introduces an error represented by a finite Fourier series and examines how this error propagates during the solution.

Stability being independent of source term, now getting the stability of explicit scheme using Von Neumann's method, we set $S = 0$ in the explicit scheme (2.7) to give the homogeneous equation;

$$u_{i,j}^{n+1} = (2-4\sigma)\, u_{i,j}^{n} +$$

$$\sigma \left(u_{i+1,j}^{n} + u_{i-1,j}^{n} + u_{i,j+1}^{n} + u_{i,j-1}^{n}\right) - u_{i,j}^{n-1} \quad (3.1)$$

Then using the fact that the solution of this constant coefficient differential equation is satisfied by the Fourier harmonics

$$U_{i,j}^{n} = \xi^n e^{i\beta mh} e^{i\gamma lh}.$$

where
β is time index in x
γ is time index in z
h is spatial step in x and z
m is spatial index in x
and l is spatial index in z.
Substituting in the homogeneous scheme (3.1), we get

$$\xi^{n+1} e^{i\beta mh} e^{i\gamma lh} = (2-4\sigma)\xi^n e^{i\beta mh} e^{i\gamma lh} +$$

$$\sigma\left[\xi^n e^{i\beta(m+1)h} e^{i\gamma lh} + \xi^n e^{i\beta(m-1)h} e^{i\gamma lh}\right] +$$

$$\sigma\left[\xi^n e^{i\beta mh} e^{i\gamma(l+1)h} + \xi^n e^{i\beta mh} e^{i\gamma(l-1)h}\right] -$$

$$\xi^{n-1} e^{i\beta mh} e^{i\gamma lh} \quad (3.2)$$

so that on dividing equation (3.2) by $\xi^n e^{i\beta mh} e^{i\gamma lh}$, we have;

$$\xi = (2-4\sigma) + \sigma\left[e^{i\beta h} + e^{-i\beta h} + e^{i\gamma h} + e^{-i\gamma h}\right] - \xi^{-1}$$

But $2\cos\theta = e^{i\theta} + e^{-i\theta}$ which reduces this equation to

$$\xi = (2-4\sigma) + \sigma\left[2(1 - 2\sin^2\frac{\beta h}{2}) + 2(1 - 2\sin^2\frac{\gamma h}{2})\right]$$

$$-\xi^{-1} \quad (3.3)$$

then;

$$\xi^2 - 2\left[1 - 2\sigma(\sin^2\frac{\beta h}{2} + \sin^2\frac{\gamma h}{2})\right]\xi + 1 = 0$$

we then let $g = \left[1 - 2\sigma(\sin^2\frac{\beta h}{2} + \sin^2\frac{\gamma h}{2})\right]$, to get;

$$\xi^2 - 2g\xi + 1 = 0,$$

where the i^{th} eigenvalue is given by

$$\xi_i = g \pm \sqrt{g^2 - 1}$$

Therefore, for stability, $|\xi_i| \leq 1; \quad i = 1, 2, \cdots, N$, this implies

$$-1 \leq 1 - 2\sigma(\sin^2\frac{\beta h}{2} + \sin^2\frac{\gamma h}{2}) \leq 1$$

which has non-trivial solution when $1 - 2\sigma(\sin^2\frac{\beta h}{2} + \sin^2\frac{\gamma h}{2}) \geq -1$, for this we get;

$$\sigma(\sin^2\frac{\beta h}{2} + \sin^2\frac{\gamma h}{2}) \leq 1,$$

since the maximum value of $\sin^2\frac{\beta h}{2}$ is unity, our equation reduces to $\sigma \leq \frac{1}{2}$ as stability condition. Therefore, convergence of the scheme follows the Courant et al. (1928) (C.F.L.) condition for convergence, which applies to explicit difference replacement of hyperbolic equations. It requires that (3.1) to be convergent when $0 \leq \sigma \leq \frac{1}{2}$. Thus, the stability condition coincides with the C.F.L. condition.

Crank-Nicolson scheme (Hybrid)

3.3 Matrix stability of Crank-Nicolson scheme

Similarly, we adopt the matrix method to analyze stability of the Crank-Nicolson scheme (2.11). We expand this scheme by taking $i = 1, 2, 3, \cdots, M-1; \quad j = 1, 2, 3, \cdots, N-1$, to get the system of equations which we can express in matrix form as

$$\begin{pmatrix} (1+4r) & -r & \cdots & -r \\ -r & (1+4r) & -r & \cdots \\ \vdots & \ddots & \ddots & \vdots \\ -r & \cdots & -r & (1+4r) \\ & -r & \cdots & (1+4r) \end{pmatrix}$$

$$\begin{bmatrix} u_{1,1} \\ u_{2,1} \\ \vdots \\ u_{1,2} \\ \vdots \\ u_{M-1,N-1} \end{bmatrix}^{n+1} =$$

$$\begin{pmatrix} (2-4r) & r & \cdots & r & \\ r & (2-4r) & r & \cdots & r \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ r & \cdots & r & (2-4r) & r \\ & r & \cdots & r & (2-4r) \end{pmatrix}$$

$$\begin{bmatrix} u_{1,1} \\ u_{2,1} \\ \vdots \\ u_{1,2} \\ \vdots \\ u_{M-1,N-1} \end{bmatrix}^n +$$

$$\begin{pmatrix} ru^{n+1}_{0,1} + ru^{n+1}_{1,0} + ru^n_{0,1} + ru^n_{1,0} - u^{n-1}_{1,1} - k^2 S^n_{1,1} \\ ru^{n+1}_{2,0} + ru^n_{2,0} - u^{n-1}_{2,1} - k^2 S^n_{2,1} \\ \vdots \\ ru^{n+1}_{0,2} + ru^n_{0,2} - u^{n-1}_{1,2} - k^2 S^n_{1,2} \\ \vdots \end{pmatrix}.$$

Which we can express in matrix form as

$$\mathbf{A}U^{n+1}_{i,j} = \mathbf{B}U^n_{i,j} + \mathbf{C}$$

$$U^{n+1}_{i,j} = (\mathbf{A}^{-1}\mathbf{B})U^n_{i,j} + \mathbf{A}^{-1}\mathbf{C} \cdots (3.3a)$$

A and **B** are block tridiagonal matrices. Thus, equation (3.3a) may be put in the form

$$(I - rA_{N-1})U^{n+1}_{i,j} = (2I + rA_{N-1})U^n_{i,j} + \mathbf{D},$$

where

$$A_{N-1} = \begin{bmatrix} -4 & 1 & 0 & \cdots & 0 \\ 1 & -4 & 1 & \cdots & 0 \\ 0 & 1 & -4 & 1 & \vdots \\ \vdots & \ddots & \cdots & 1 & -4 \\ 0 & \cdots & 0 & \cdots & -4 \end{bmatrix}$$

I is an $(N-1) \times (N-1)$ identity matrix. Thus,

$$U^{n+1}_{i,j} = \left[(2I + rA_{N-1})(I - rA_{N-1})^{-1}\right] U^n_{i,j} + \mathbf{E},$$

where $\mathbf{D} = \mathbf{A}^{-1}\mathbf{C}$ and $E = \mathbf{C}(I - rA_{N-1})^{-1}$.
In simpler form we write this equation as

$$U^{n+1}_{i,j} = PU^n_{i,j} + \mathbf{E}$$

In this case, $P = (2I + rA_{N-1})(I - rA_{N-1})^{-1}$ is the amplification matrix, and the stability condition is that absolute value of the eigenvalues of the amplification matrix should be less than or equal to 1, that is $\mid \lambda_i \mid \leq 1$. Since our Equation (2.11) is implicit and A and B are block tridiagonal matrices which are symmetric positive definite and are weakly diagonally dominant, then A and B are non-singular thus there is a unique solution, the symmetry then implies that we have both necessary and sufficient condition for stability, therefore this scheme will always be stable for all values of r since r has no restrictions (unconditionally stable).

3.4 Von Neumann stability of Crank-Nicolson scheme

To get stability of Crank Nicolson via this method, we set $S = 0$ since stability is independent of source term, then substitute $U^n_{i,j} = \xi^n e^{i\beta mh} e^{i\gamma lh}$ in the homogeneous equation (2.11)

$$-ru^{n+1}_{i+1,j} + (1+4r)u^{n+1}_{i,j} - ru^{n+1}_{i-1,j} - ru^{n+1}_{i,j+1} - ru^{n+1}_{i,j-1} = ru^n_{i+1,j} + (2-4r)u^n_{i,j} + ru^n_{i-1,j} + ru^n_{i,j+1} + ru^n_{i,j-1} - u^{n-1}_{i,j}, \quad (3.4)$$

which yields

$$-r\xi^{n+1} e^{i\beta(m+1)h} e^{i\gamma lh} + (1+4r)\xi^{n+1} e^{i\beta mh} e^{i\gamma lh} - r\xi^{n+1} e^{i\beta(m-1)h} e^{i\gamma lh} - r\xi^{n+1} e^{i\beta mh} e^{i\gamma(l+1)h} - r\xi^{n+1} e^{i\beta mh} e^{i\gamma(l-1)h} = r\xi^n e^{i\beta(m+1)h} e^{i\gamma lh} + (2-4r)\xi^n e^{i\beta mh} e^{i\gamma lh} + r\xi^n e^{i\beta(m-1)h} e^{i\gamma lh} + r\xi^n e^{i\beta mh} e^{i\gamma(l+1)h} + r\xi^n e^{i\beta mh} e^{i\gamma(l-1)h} - \xi^{n-1} e^{i\beta mh} e^{i\gamma lh} \quad (3.5)$$

Again dividing (3.5) by $\xi^n e^{i\beta mh} e^{i\gamma lh}$, we obtain

$$(1+4r)\xi - r\xi(e^{i\beta h} + e^{-i\beta h}) - r\xi(e^{i\gamma h} + e^{-i\gamma h}) = (2-4r) + r(e^{i\beta h} + e^{-i\beta h}) + r(e^{i\gamma h} + e^{-i\gamma h}) - \xi^{-1} \quad (3.6)$$

Recall that $\cos\theta = 1 - 2\sin^2\frac{\theta}{2} = \frac{e^{i\theta} + e^{-i\theta}}{2}$, therefore using this fact in (3.6), yields

$$(1+4r)\xi - r\xi\left(2(1 - 2\sin^2\frac{\beta h}{2}) + 2(1 - 2\sin^2\frac{\gamma h}{2})\right) = (2-4r) + r\left(2(1 - 2\sin^2\frac{\beta h}{2}) + 2(1 - 2\sin^2\frac{\gamma h}{2})\right) - \frac{1}{\xi}$$

After rearrangement, we get

$$\xi^2\left[1 + 4r(\sin^2\frac{\beta h}{2} + \sin^2\frac{\gamma h}{2})\right]$$

$$-\xi\left[2 - 4r(\sin^2\frac{\beta h}{2} + \sin^2\frac{\gamma h}{2})\right] + 1 = 0$$

which has a non-trivial solution when $-1 \leq \xi_i \leq 1$, where ξ_i is the magnification factor corresponding to eigenvalue, thus;

$$\xi_i = \frac{2 - 4r(\sin^2\frac{\beta h}{2} + \sin^2\frac{\gamma h}{2})}{2 + 8r(\sin^2\frac{\beta h}{2} + \sin^2\frac{\gamma h}{2})} \leq 1$$

Now for $\mid \xi_i \mid \leq 1$, we have $\sin^2\frac{\beta h}{2} = 1$ and $\sin^2\frac{\gamma h}{2} = 1$, therefore

$$\xi_i = \frac{1 - 4r}{1 + 8r}$$

Hence for stability $r > 0$, which makes ξ_i less than unity for all values of r implying unconditional stability throughout.

Analysis and Software

In this section we present an analysis of the numerical experiments. We also present and discuss the results obtained from these methods. We shall display these results using three- dimensional figures and graphs.

From the initial condition

$$u_t(x, z, 0) = 0$$

But since u_t is approximated using central difference i.e. (**??**), then central difference analogue of u_t yields

$$u_t \approx \frac{u_{i,j}^{n+1} - u_{i,j}^{n-1}}{2k} = 0$$

Taking $n = 0$, from initial condition we find

$$u_t \approx \frac{u_{i,j}^{0+1} - u_{i,j}^{0-1}}{2k} = 0,$$

where

$$k = \Delta t$$

Implying that

$$u_{i,j}^{1} = u_{i,j}^{-1}$$

Again, from the initial condition, $u(x, z, 0) = (\sin \pi x)(\sin \pi$
we get that

$$u(x, z, 0) \approx u_{i,j}^{0} = (\sin \pi x)(\sin \pi z)$$

At this point we developed a Matlab program that could give the pressure field as a function of x and z at varying time levels and results have been plotted for both equation (2.7) and (2.11).

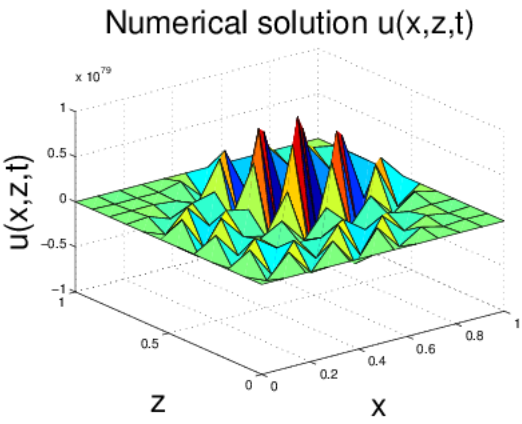


Figure 3.1: Numerical solution explicit scheme at c=1500,dt=0.5

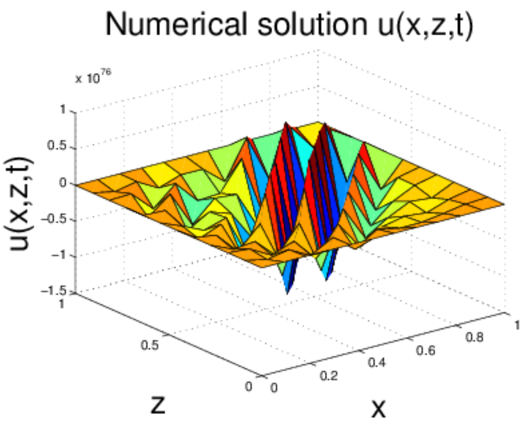


Figure 3.2: Numerical solution Crank Nicolson scheme at c=1500,dt=0.5

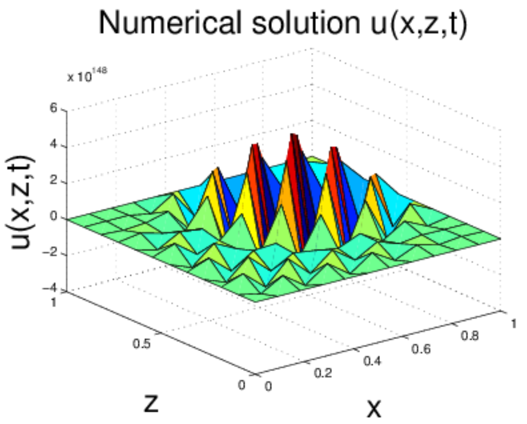


Figure 3.3: numerical solution of explicit scheme at t=10,c=1500.dt=0.5

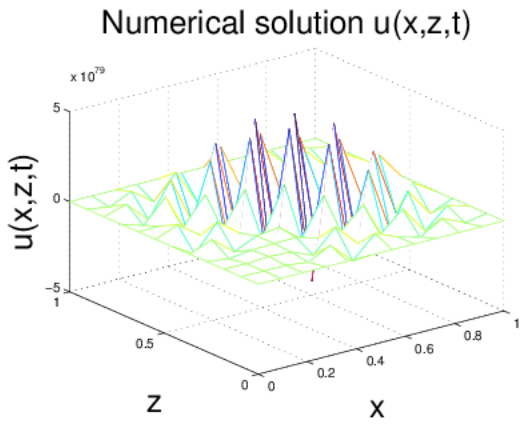


Figure 3.5: Numerical solution explicit scheme at c=1000,dt=0.8

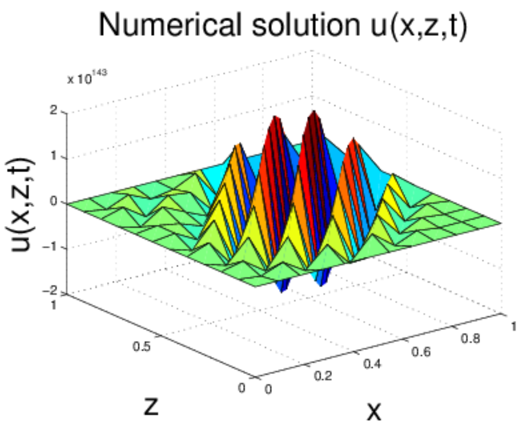


Figure 3.4: numerical solution of Crank Nicolson scheme at t=10,c=1500.dt=0.5

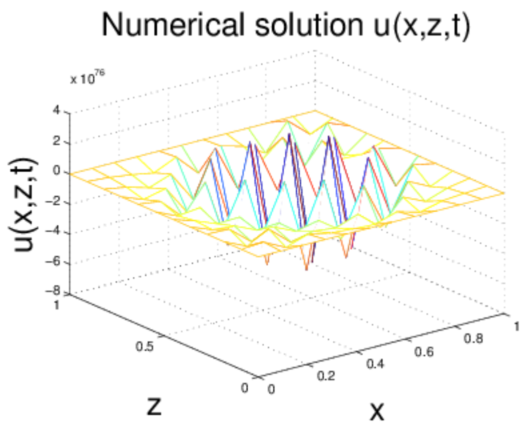


Figure 3.6: Numerical solution Crank Nicolson scheme at c=1000,dt=0.8

4 Discussions

In reality, sound propagation in elastic medium is damped, the amplitude of the pressure of the sound wave decreases with increasing distance from the sound source. Our results from the two numerical schemes (CDS) and (CNS) are confirming this since the displacement of the particles given by $u(x, z, t)$ is decreasing with an increase in the distance from the source (in this case t=0). The efficacy of a finite difference scheme is achieved with the increase of the grid points involved hence the increase in the accuracy of a finite difference scheme. In addition, the speed of sound reduces with increase in the distance from the source, this is evidenced by the reduction of the ripples as the propagation advances away from source see figure 4.1.3.

Conclusions

This study focussed on the second order acoustic equation with a signal function. Two numerical schemes namely Central Difference Scheme (Explicit scheme) and Hybrid scheme (Crank Nicolson Scheme) were developed and used in this study. The stability analyses of the developed schemes revealed that Explicit scheme was conditionally stable while the Hybrid one (Crank Nicolson Scheme) was unconditionally stable, for all values of courant number r.

The rate of convergence of the algorithms depends on the truncation error introduced when approximating the partial derivatives, the Crank-Nicolson method converges at the rate of $(k^2 + h^2)$, which is a faster rate of convergence than either the explicit method, or the implicit method. Further, since c is a function of (x, z), from the results it suffices to use the maximum sound velocity in the model.

The smaller the mesh sizes, the more finely the results, this makes the grid more finer thus improving the approximation around the boundary but at the cost of strongly increased computational time as evidenced by figures (4.1.5, 4.1.6).

5 Recommendations

We wish to recommend that further research can be undertaken to;

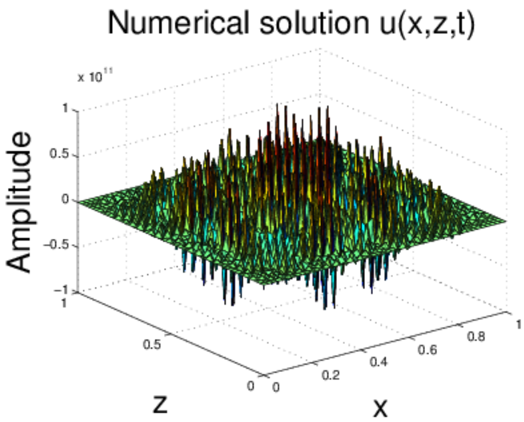
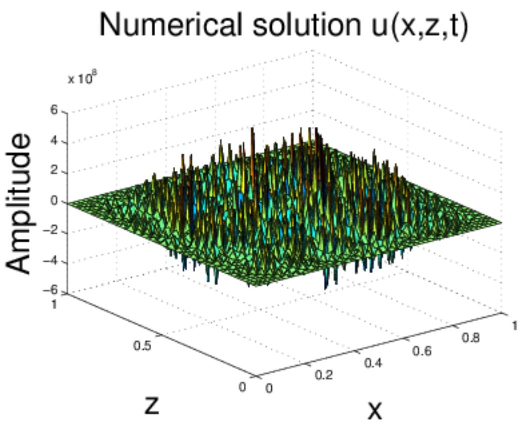


Figure 3.7: Numerical solution explicit scheme at t=5,c=0.5,dt=0.5



(1).png

Figure 3.8: Numerical solution Crank Nicolson scheme at t=5,c=0.5,dt=0.5

(i) Explore numerical solution to this problem using other methods like finite element and compare results.

(ii) Try out an analytical method via green's function.

References

[1] Alford, R. M., Kelly, K. R., and Boore, D. M. (1974): Accuracy of finite-difference modeling of acoustic wave propagation, Geophysics, **vol. 39**, pp 834-842.

[2] Alford, R. M., Kelly, K. R., and Boore, D. M. (1974): Accuracy of finite-difference modeling of acoustic wave propagation, Geophysics, **vol. 39**, pp 834-842.

[3] Bernatz, R. (2010): *Fourier series and numerical methods for partial differential equations*, Luther College, John Wiley & Sons, inc., publication.

[4] Borthen, J. (2010): *A numerical approximation of the wave equation*, Mathematics and Statistics, Georgetown University.

[5] Brekhovskikh, L. M., and Yu Lysanov, P. (2003): *Fundamentals of Ocean Acoustics*, 3rd edition, Springer-Verlag, NY.

[6] Cerjan, C., Kosloff, D., Kosloff, R., and Reshef, M. (1985): A nonreflecting boundary condition for discrete acoustic and elastic wave equations, Geopgysics, **vol. 50, no. 4**, pp.705-708.

[7] Charara, M. and Tarantola, A. (1996): Boundary conditions and the source term for one-way acoustic depth extrapolation, Geophysics,**vol. 61** , pp 244-252.

[8] Clay, C. S., Medwin, H. (1977): *Acoustical Oceanography: Principles and Applications*, John Wiley and Sons, New York, NY, pp. 88 and 98-99.

[9] Daniel, R. R. (2009): *The science and Application of acoustics second edition*, The City College of the City University of New York.

[10] David, H. and Robert, R. (1989): *Fundamental of physics*, 2nd Edition, John Wiley and sons, New York.

[11] Evans, C. L. (1997): *Partial Differential Equations*, American Mathematical Society.

[12] Frank J. F. (2000): *Foundations of engineering acoustics*, Institite of sound and vibration research, University of Southampton, U.K.

[13] Hugh D. G. and Pat, F. D. (2003): Finite difference modelling of the full acoustic wave equation in Matlab, CREWES Research Report **vol. 15**.

[14] Jain, M. K. (1991): *Numerical solution of Differential Equations*, Wiley Eastern Limited, New Delhi.

[15] Jensen, F., Kuperman, W., Porter, M., and Schmidt, H. (2000): *Computational Ocean Acoustics*, springer, New York, pp.11-12 and 52-54.

[16] John, H. M. (2001): *Numerical Methods for Mathematics, Science and Engineering*,2nd Edition, Prentice Hall of India. New Delhi.

[17] Kelly, K. R., Ward, R. W., Sven T. and Alford, R. M. (1976): Synthetic Seismograms: A finite-difference Approach, Geophysics, **vol. 41, no. 1**, p.2-27.

[18] Lines, L. R. Slawinski, R. and Bording, R. P. (1999): A recipe for stability of finite difference wave equation computations, Geophysics, **vol. 64**, pp 967-969.

[19] Maarten, V. W. and Kowalczyk, K. (2008): *On the numerical solution of the 2D wave equation with compact FDTD schemes*, Sonic Arts Research Centre, Queen's University Belfast, United Kingdom.

[20] Randall. J. L. (2005): *Finite difference Methods for Differential Equations*, lecture notes for Amath 585-586 University of Washington.

[21] Rao, S. K. (2004): *Introduction to Partial Differential Equations*, Prentice Hall of India, New Delhi.

[22] Richtmyer, R. D. and Morton, K. W. (1967): *Tridiagonal Algorithm*, Lecture notes.

[23] Rodney, F. W. C. (1990): *Underwater acoustic systems*, John Wiley and sons, New York.

[24] Seongjai, K. (2002): *Higher-Order Schemes for Acoustic Waveform Simulation*, Acoustic Waveform Simulation , Department of Mathematics, University of Kentucky, Lexington, Kentucky 40506-0027 USA, March.

[25] Serway, R. A. and Faughn, J. S. (1992): *Student Solutions, Manual and Study Guide to accompany College Physics*, 4th Edition. Saunders college Publishing.

[26] Strikwerda, J. C. (1989): *Finite Difference Schemes and Partial Differential Equations*, Wadsworth and Brook/Cole.

[27] Vetreno, J. R. (2007): *Analytic Models for Acoustic Wave Propagation in Air*, Master's Thesis, North Carolina State University.

[28] William, F. A. (1992): *Numerical methods for partial differential equations*, third edition, Academic press, New York.

[29] Xie, X. B., and Wu, R.S. (1997): *Free surface boundary condition and the source term for one-way elastic wave method*, Institute of Tectonics, University of California, Santa Crux.

6 Appendix

6.1 Matrix Generation

Set one, $j = 1$;

$$\begin{aligned}
u_{1,1}^{n+1} &= ru_{2,1}^{n} + (2-4r)u_{1,1}^{n} + ru_{0,1}^{n} + ru_{1,2}^{n} + ru_{1,0}^{n} - c_{1,1}^{2}(\Delta t)^{2}S_{1,1}^{n} - u_{1,1}^{n-1}\\
u_{2,1}^{n+1} &= ru_{3,1}^{n} + (2-4r)u_{2,1}^{n} + ru_{1,1}^{n} + ru_{2,2}^{n} + ru_{2,0}^{n} - c_{2,1}^{2}(\Delta t)^{2}S_{2,1}^{n} - u_{2,1}^{n-1}\\
u_{3,1}^{n+1} &= ru_{4,1}^{n} + (2-4r)u_{3,1}^{n} + ru_{2,1}^{n} + ru_{3,2}^{n} + ru_{3,0}^{n} - c_{3,1}^{2}(\Delta t)^{2}S_{3,1}^{n} - u_{3,1}^{n-1}\\
\vdots &= \vdots\ \vdots\\
u_{M-1,1}^{n+1} &= ru_{M,1}^{n} + (2-4r)u_{M-1,1}^{n} + ru_{M-2,1}^{n} + ru_{M-1,2}^{n} + ru_{M-1,0}^{n} -\\
&\qquad c_{M-1,1}^{2}(\Delta t)^{2}S_{M-1,1}^{n} - u_{M-1,1}^{n-1}
\end{aligned}$$

In set two, we set $j = 2$ to generate the systems of equations

$$\begin{aligned}
u_{1,2}^{n+1} &= ru_{2,2}^{n} + (2-4r)u_{1,2}^{n} + ru_{0,2}^{n} + ru_{1,3}^{n} + ru_{1,1}^{n} - c_{1,2}^{2}(\Delta t)^{2}S_{1,2}^{n} - u_{1,2}^{n-1}\\
u_{2,2}^{n+1} &= ru_{3,2}^{n} + (2-4r)u_{2,2}^{n} + ru_{1,2}^{n} + ru_{2,3}^{n} + ru_{2,1}^{n} - c_{2,2}^{2}(\Delta t)^{2}S_{2,2}^{n} - u_{2,2}^{n-1}\\
u_{3,2}^{n+1} &= ru_{4,2}^{n} + (2-4r)u_{3,2}^{n} + ru_{2,2}^{n} + ru_{3,3}^{n} + ru_{3,1}^{n} - c_{3,2}^{2}(\Delta t)^{2}S_{3,2}^{n} - u_{3,2}^{n-1}\\
\vdots &= \vdots\ \vdots\\
u_{M-1,2}^{n+1} &= ru_{M,2}^{n} + (2-4r)u_{M-1,2}^{n} + ru_{M-2,2}^{n} + ru_{M-1,3}^{n} + ru_{M-1,1}^{n} -\\
&\qquad c_{M-1,2}^{2}(\Delta t)^{2}S_{M-1,2}^{n} - u_{M-1,2}^{n-1}
\end{aligned}$$

Continuing in the same trend, we set $j = 3$ to give

$$\begin{aligned}
u_{1,3}^{n+1} &= ru_{2,3}^{n} + (2-4r)u_{1,3}^{n} + ru_{0,3}^{n} + ru_{1,4}^{n} + ru_{1,2}^{n} - c_{1,3}^{2}(\Delta t)^{2}S_{1,3}^{n} - u_{1,3}^{n-1}\\
u_{2,3}^{n+1} &= ru_{3,3}^{n} + (2-4r)u_{2,3}^{n} + ru_{1,3}^{n} + ru_{2,4}^{n} + ru_{2,2}^{n} - c_{2,3}^{2}(\Delta t)^{2}S_{2,3}^{n} - u_{2,3}^{n-1}\\
u_{3,3}^{n+1} &= ru_{4,3}^{n} + (2-4r)u_{3,3}^{n} + ru_{2,3}^{n} + ru_{3,4}^{n} + ru_{3,2}^{n} - c_{3,3}^{2}(\Delta t)^{2}S_{3,3}^{n} - u_{3,3}^{n-1}\\
\vdots &= \vdots\ \vdots\\
u_{M-1,3}^{n+1} &= ru_{M,3}^{n} + (2-4r)u_{M-1,3}^{n} + ru_{M-2,3}^{n} + ru_{M-1,4}^{n} + ru_{M-1,2}^{n} -\\
&\qquad c_{M-1,3}^{2}(\Delta t)^{2}S_{M-1,3}^{n} - u_{M-1,3}^{n-1}
\end{aligned}$$

Setting $j = 4$ yields

$$\begin{aligned}
u_{1,4}^{n+1} &= ru_{2,4}^{n} + (2-4r)u_{1,4}^{n} + ru_{0,4}^{n} + ru_{1,5}^{n} + ru_{1,3}^{n} - c_{1,4}^{2}(\Delta t)^{2}S_{1,4}^{n} - u_{1,4}^{n-1}\\
u_{2,4}^{n+1} &= ru_{3,4}^{n} + (2-4r)u_{2,4}^{n} + ru_{1,4}^{n} + ru_{2,5}^{n} + ru_{2,3}^{n} - c_{2,4}^{2}(\Delta t)^{2}S_{2,4}^{n} - u_{2,4}^{n-1}\\
u_{3,4}^{n+1} &= ru_{4,4}^{n} + (2-4r)u_{3,4}^{n} + ru_{2,4}^{n} + ru_{3,5}^{n} + ru_{3,3}^{n} - c_{3,4}^{2}(\Delta t)^{2}S_{3,4}^{n} - u_{3,4}^{n-1}\\
u_{4,4}^{n+1} &= ru_{5,4}^{n} + (2-4r)u_{4,4}^{n} + ru_{3,4}^{n} + ru_{4,5}^{n} + ru_{4,3}^{n} - c_{4,4}^{2}(\Delta t)^{2}S_{4,4}^{n} - u_{4,4}^{n-1}\\
\vdots &= \vdots\ \vdots\\
u_{M-1,4}^{n+1} &= ru_{M,4}^{n} + (2-4r)u_{M-1,4}^{n} + ru_{M-2,4}^{n} + ru_{M-1,5}^{n} + ru_{M-1,3}^{n} -\\
&\qquad c_{M-1,4}^{2}(\Delta t)^{2}S_{M-1,4}^{n} - u_{M-1,4}^{n-1}
\end{aligned}$$

```
%compute initial values of u
% Now because the pde is second order in time, we must specify initial
% values of the solution and its derivative
u_initslope = zeros(size(x)); % initial values of du/dt
figure(1);% plot for the initial values of u in surface form, so the first figure you obtain is for initial values
surf(x,z,u_init);
xlabel('x','Fontsize',24);
ylabel('z','Fontsize',24);
zlabel('u(x,z,U)','Fontsize',24);
title('initial values of u','Fontsize',24)
% keyboard
u_0 = u_init(2:nz+1,2:nx+1);%This generates interior values of u initially. Use MATLAB help to get detail of this notation
% u_U = zeros(length(x));
u_0 = u_0(:);%The interior initial values generated are stored as column vector(i.e transform elements from the interior matrix into a column vecto
%% Quantities necessary to assemble matrices. Remember we are solving a
%% system of equations in the form y=Ax where is a diagonal  sparse matrix.
%%
sigma = (c*dt/dx)^2;
phi =  (c*dt)^2;
Adiag = ones(N,1); % elements along the main diagonal of A
Asub = ones(N,1);  % elements along the lowest diagonal of A
Asub1 = ones(N,1); % elements along the diagonal just below the main diagonal of A
Asup = ones(N,1);  % elements along the uppermost diagonal of A
Asup1 = ones(N,1); % elements along the diagonal just above the main diagonal of A
%% We account for the zeros along the diagonals which happens to be the
%% Asub1 and Asup1 in this case.
for i = 1:nx
    Asub1(i*nz) = 0;
end
for i = 1:nx
    Asup1((i-1)*nz+1) = 0;
end
%% Main MATLABcode
```

This process is continued until $i = (M-1), \quad j = (N-1)$ as below

$$u^{n+1}_{1,N-1} = ru^n_{2,N-1} + (2-4r)u^n_{1,N-1} + ru^n_{0,N-1} + ru^n_{1,N} + ru^n_{1,N-2} -$$

$$c^2_{1,N-1}(\Delta t)^2 S^n_{1,N-1} - u^{n-1}_{1,N-1}$$

$$u^{n+1}_{2,N-1} = ru^n_{3,N-1} + (2-4r)u^n_{2,N-1} + ru^n_{1,N-1} + ru^n_{2,N} + ru^n_{2,N-2} -$$

$$c^2_{2,N-1}(\Delta t)^2 S^n_{2,N-1} - u^{n-1}_{2,N-1}$$

$$u^{n+1}_{3,N-1} = ru^n_{4,N-1} + (2-4r)u^n_{3,N-1} + ru^n_{2,N-1} + ru^n_{3,N} + ru^n_{3,N-2} -$$

$$c^2_{3,N-1}(\Delta t)^2 S^n_{3,N-1} - u^{n-1}_{3,N-1}$$

$$\vdots \;=\; \vdots \;\vdots$$

$$u^{n+1}_{M-1,N-1} = ru^n_{M,N-1} + (2-4r)u^n_{M-1,N-1} + ru^n_{M-2,N-1} + ru^n_{M-1,N} + ru^n_{M-1,N-2} -$$

$$c^2_{M-1,N-1}(\Delta t)^2 S^n_{M-1,N-1} - u^{n-1}_{M-1,N-1}$$

6.2 Matlab Programme