ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY

SCHOOL OF APPLIED NATURAL SCIENCE

DEPARTMENT OF MATHEMATICS

APPLICATION OF FINITE DIFFERENCE METHODS FOR SOLVING DAMPED WAVE EQUATION

A thesis submitted in partial fulfillment of the requirements for the Degree of Masters of Science in Numerical Analysis

BY: AYALEW MINDAHUN

January, 2018
Adama, Ethiopia

ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY

SCHOOL OF APPLIED NATURAL SCIENCE

DEPARTMENT OF MATHEMATICS

APPLICATION OF FINITE DIFFERENCE METHODS FOR SOLVING DAMPED WAVE EQUATION

A thesis submitted in partial fulfillment of the requirements for the Degree of Masters of Science in Numerical Analysis

BY: AYALEW MINDAHUN

Advisor: TEKLE GEMECHU (Ph. D)

January, 2018
Adama, Ethiopia

SIGNATURE PAGE (Approval sheet)

Name of student	Signature	Date
Advisor	Signature	Date
Examiner	Signature	Date
HoD	Signature	Date
School Dean	Signature	Date
SGS Dean	Signature	Date

Acknowledgement

First and above all, I praise God, the Almighty who gave me long life and granting me the capability to accomplish my study.

Secondly, my deepest thanks go to my advisor Dr. Tekle Gemechu for his support, advice, guidance and providing the necessary reference materials in the preparation of this thesis.

Also my heartily thanks go to Ababy Hailu who assisted me on assessing reference materials from internet and helped me by his constructive comments and supports to do my study strongly.

Contents

LIST OF SYMBOLS AND ACRONYMS

Δx: the distance between two grid points in the space

Δt: the distance between two time steps.

$O(\Delta x)^2$: second order spatial truncation error

$O(\Delta t)^2$: second order temporal truncation error

$u = u(x, t)$: the vertical displacement of the vibration of a string (analytic solution of the PDE)

u_i^n: the approximate numerical solution of the PDE which is obtained by solving the finite difference equation at n time level

$\hat{u}_i^n$: the exact solution of the PDE at n time level

(i, n): a grid point on the string

$\frac{\partial u}{\partial x}$: u_x: the first partial derivative of u with respect to x

$\frac{\partial^2 u}{\partial x^2}$: u_{xx}: the second partial derivative of u with respect to x

L_Δ: A finite difference operator

PDE: Partial Differential Equation

FDM: Finite Difference Method

DWE: Damped Wave Equation

FDTD: Finite-Difference Time-Domain

PDE: Partial Differential Equation

HPDE: Hyperbolic Partial Differential Equation

ICs: Initial Conditions

BCs: Boundary Conditions

LIST OF FIGURES

LIST OF TABLES

Abstract

Partial differential equations are difficult to solve analytically in general but, there exist a variety of numerical schemes that convert impossible calculus into myriad simple calculations that a computer can readily perform. This study presents an explanation of a time dependent damped wave equation from a vibrating string considering the transverse displacement of a plucked string and the subsequent vibration of the string, and application of finite difference methods for solving damped wave equation. The study is planned to find numerical solution of damped wave equation using explicit and implicit finite difference methods. The finite difference method discretizations proceed by replacing the derivatives in the damped wave equations by finite difference approximations. These give a large number of algebraic systems of equations to be solved. The computational experiments are performed writing MATLAB program or using Thomas Algorithm for explicit and implicit finite difference methods to generate numerical solutions to the damped wave equation. Stability, consistency and convergence of the methods are considered based on Δx *and* Δt *in the study. The methods approximate the solutions with consistency of the* $O\{(\Delta t)^2, (\Delta x)^2\}$*, for examining the accuracy of the results. Numerical examples are presented for each method. The results are compared with the analytic solution, obtained by the methods of separation of variable. Based on error analysis, we present results using tables and graphs.*

CHAPTER ONE

1 Introduction

1.1 Background of the study

The study of waves helps to model many things that are useful in everyday life i.e., musical instruments, engineering devices for weather fore cast, tsunami and earth quake detection devices as well as basic communication devices, and so on. This study presents a numerical simulation of mechanical waves from a vibrating string. In particular, damped wave equation. A vibrating string is just a model of many objects that vibrates in nature. Most vibrations result into wave motion which carries energy that can be controlled for human benefits. A vibrating string presents a better and an initial point to the study of waves because the variables can be easily manipulated.

In the study by Saberi Najafi & Izadi (2014), comparison of two explicit FDMs was presented to approximate the numerical solution of one dimensional damped wave equation. In their work, they studied the three-level explicit and two-level explicit FDMs taking different values of time step,Δt, space size,Δx and damping coefficient k (which is a small positive constant) to approximate the numerical solution. Considering all the results obtained, they arrived at the conclusion that although the two methods work with high accuracy, but the three-level explicit method is better than the two-level explicit method for solving the one dimensional damped wave equation.

1.2 Statement of the problem

In the study by Saberi Najafi & Izadi (2014), the implicit FDM was not presented to approximate the numerical solution of one dimensional damped wave equation in comparison with the explicit FDM. So, in this study a thorough evaluation of the model is treated applying the explicit and implicit FDMs taking different values of time step, Δt and space size, Δx considering a stretched string L of length of negligible weight. Supposing the two ends of the string are firmly secured at some supports so they are not move and assuming the set-up has damping considering friction, i.e., the reaction of the surrounding medium (air for example) and density of the string on the motion of the string, and the string is set to vibrate by displacing it from its equilibrium position and then releasing it. Then, the vertical displacement of the string at any time, $t > 0$ and $0 < x <$ L, is given by the displacement function $u\ (x\ , t)$.

And we want to determine the subsequent motion of the string by finding the approximate numerical solution of $u\ (x\ , t)$, i.e., $u\ (x_i\ , t_n)$, for $t > 0$, $0 < x <$ L. To find $u\ (x_i\ , t_n)$, we solve the equation subject to the BCs and ICs. The work also includes a systematic exploration of stability, consistency and convergence of the methods.

Finally, we compare the analytic solution with the numerical solutions of the methods, and errors of the methods with each other. We can then determine which method is better and more efficient to find the numerical solution of damped wave equation.

And owing this, the study attempts to answer the following questions.

- To what extent the methods are stable and which method is more stable?
- To what extent the methods are consistent and which method is more consistent?
- To what extent the methods converge and which method is more convergent?
- Which method is more appropriate to solve damped wave equation?
- Which method has less error?

1.3 Objective of the study

1.3.1 General objective

The general objective of this study is to apply the explicit and implicit FDMs for solving one dimensional damped wave equation taking different values of time step, Δt and space size, Δx.

1.3.2 Specific objectives

The specific objectives of the study are:

- To apply the explicit and implicit FDMs to determine the subsequent motion of the string by finding the approximate numerical solution of one dimensional damped wave equation subject to its ICs and BCs in different cases.
- To determine the stability, consistency and convergence of the methods.
- To determine which method is more efficient in solving damped wave equation.

1.4 Significance of the study

At the end, the outcome of this study may

- develop skills in the study,
- be used as additional reference material for students and anyone who needs to conduct further study on application of FDMs on damped wave equation in 1D and 2D, and
- improve the application of FDMs for scientific investigation in the area of applied mathematics.

1.5 Delimitation/scope of the study

In this study, we plan to apply the explicit and implicit finite difference schemes to approximate the numerical solution of damped wave equation considering the damped vibration of a string with fixed ends taking different values of time step,Δt & step size,Δx and damping coefficient k.

1.6 Limitation of the study

The limitations of this study are:

- lack of internet access to get adequate reference material and to review the literature and
- lack of knowledge and experience of working with MATLAB program.

1.7 Organization of the study

This document has been organized as in the following. In the first chapter background of the study, statement of the problem, question, purpose of the study, objectives of the study, significance of the study, limitation and delimitation of the study are presented. The second chapter deals with review of the related literature. The third chapter focuses on methods and procedures. The fourth chapter deals with results and discussion. The fifth chapter is about conclusion and recommendations.

CHAPTER TWO

2 Review of the related literature

2.1 Partial Differential Equations (PDEs)

Partial differential equations (PDEs) occur in many branches of applied mathematics, for example, in hydrodynamics, elasticity, quantum mechanics and electromagnetic theory, S.S. Sastry (2006). Moreover, PDE is a many-faceted subject that was created to describe the mechanical behavior of objects such as vibrating strings. The vibration of a string is time and space dependent, a typical example of multidimensional systems. Lutz & Rudolf (2000) noted that "Multidimensional physical phenomena depending on time and space are commonly described by PDEs", Mango, J.M. et al. (2014). PDEs are models of various physical and geometrical problems, arising when the unknown function (the solution), call it *u*, which depend on two or more variables, usually on time *t* and one or several space variables.

We say that a PDE is linear if it is of the first degree in the unknown function *u* and its partial derivatives, with coefficients depending only on the independent variables. Otherwise we call it non-linear. We call a linear PDE homogeneous if each of its terms contains either *u* or one of its partial derivatives. Otherwise we call the equation non-homogeneous, Erwin Kreyszing (2006) & C. Chapra (2010). In PDE, there exist three types of equation namely elliptic, parabolic and hyperbolic equations. The general linear PDE of the second order in two independent variables is of the form:

$$A(x, y)\frac{\partial^2 u}{\partial x^2} + B(x, y)\frac{\partial^2 u}{\partial x \partial y} + C(x, y)\frac{\partial^2 u}{\partial y^2} + F\left(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right) = 0 \qquad (2.1)$$

Such a PDE is said to be:

Elliptic, if $B^2 - 4AC < 0$,

e.g.: Two-dimensional Laplace equation; $u_{xx} + u_{yy} = 0$

Two-dimensional Poisson equation; $u_{xx} + u_{yy} = f(x, y)$

Parabolic, if $B^2 - 4AC = 0$

e.g.: One-dimensional heat equation; $u_t = c^2\, u_{xx}$

Hyperbolic, if $B^2 - 4AC > 0$

e.g.: One-dimensional wave equation; $u_{tt} = c^2\, u_{xx}$, C. Chapra (2010).

2.2 Wave Motion

Waves are classified as either mechanical or electromagnetic. Mechanical waves require some kind of elastic material medium of propagation. An elastic medium is characterized by a continuum of points where a displacement of one point (source) immediately experiences reaction forces from neighboring points. The reaction forces at the neighboring produce forces on their neighboring points and by continuing this process the initial displacement gets propagated into the elastic medium by way of neighboring points being affected as a function of time. An electromagnetic wave differs from a mechanical wave in that it can propagate through a vacuum.

The wave motion is the movement of a disturbance from some source. In the process energy and momentum are transferred from the source. It can be transverse or longitudinal, Dan Russell (2016). Figure 1(a) illustrates a long string which is given an initial displacement and then suddenly released. The elastic properties of the string cause forces which try to pull/restore the initial displacement of the string back to its equilibrium position. The restoring forces affect points immediately to the right of the displacement. These neighboring points are pulled up ward and down ward and so the displacement moves to the right at a definite speed which depends up on the material properties of the string. This is an example of a transverse wave where the displacement of the elastic medium is perpendicular to the direction of wave propagation. A vibrating string is an example of transverse wave motion.

In Figure 1(b) there is illustrated masses connected together by springs to form a chain. The masses might represent atoms and the springs might represent the forces between the atoms. When one mass is displaced longitudinally there is created spring elongations and compressions which produce forces that act on the neighboring masses. If the left most mass is given an initial displacement to the right, the compression disturbance is created which in turn propagates to the right along the chain. This is an example of longitudinal wave. Sound wave is an example of longitudinal wave.

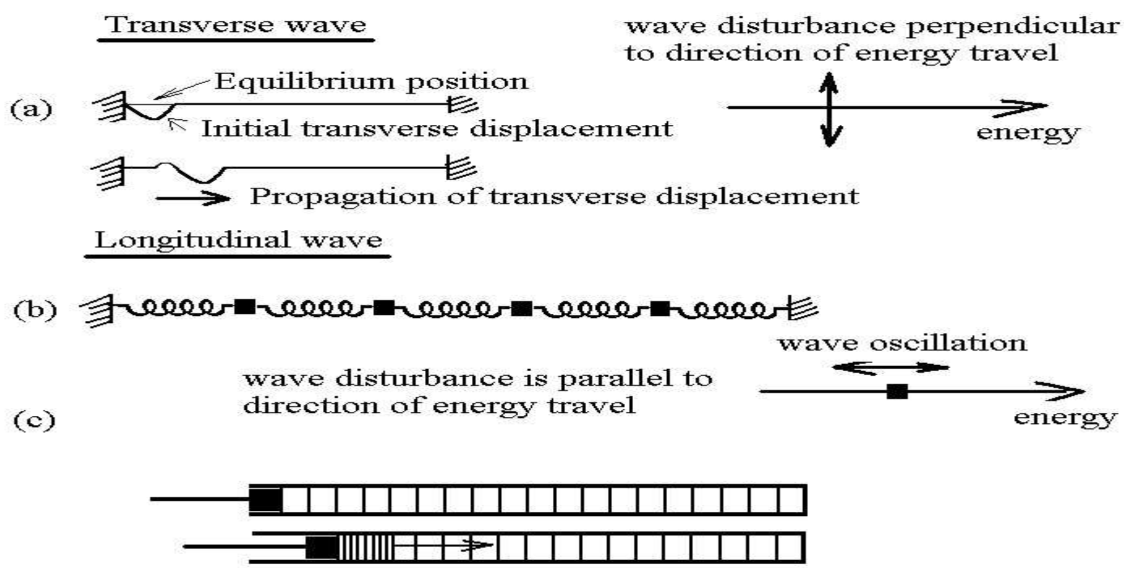


Figure 1: Transverse and longitudinal wave motion

Figure 1(c) illustrates a gas with density ρ inside a circular pipe with a cross-sectional area A. Imagine a piston given an impulsive force F which causes the piston to move with velocity c over a time interval Δt. This sudden motion compresses the gas inside the pipe and generates a compression wave moving to the right. This is an example of a longitudinal wave, Nathaniel Page Sites (2003).

2.3 Wave Equation

As N. Yener (2011) stated quoting X. Lu (2010), wave equation is an extremely important mathematical model and it is widely used by physics and engineers in describing the propagation of water waves, sound waves, electromagnetic waves, seismic waves, gravity waves, gas dynamics, the motion of vibrating string and membrane. However, this study will focus on the damped wave by simulating one dimensional wave on a vibrating string. The vibrating string is a simple physical model that can be described with the wave equation. The solution consists of a number of characteristics states with which the string oscillates. A variety of engineering systems such as vibrations of rods and beams, motion of fluid waves, transmission of sound and electrical signals can be characterized by this model, C. Chapra (2010).

2.4 Damped wave

A damped wave is a wave whose amplitude of oscillation decreases with time, eventually going to zero. In a damped wave equation we consider friction, i.e., the reaction of the surrounding medium (air resistance for example) and density of the string on the motion of the string. The effect of friction is, of course, to damp out the free vibration. For small amplitude, motion of this reaction opposes the motion of each element of the string and is proportional to the element's velocity. The friction term k opposes motion of the string and means that eventually vibrations decay with time, G. Duffy (1998). As stated by Dean G. Duffy (1998) quoting the first published solution by Kirchhoff G. (1857): on the motion of electricity in wires, "The damped wave equation first arose in the mathematical description of the telegraph, it is generally known as the equation of telegraphy.'' This term is also refers to an early method of radio transmission produced by spark gap transmitters, which consisted of a series of damped electromagnetic waves. Information was carried on this signal by telegraphy, turning the transmitter on and off (on-off keying) to send messages in Morse code. Damped waves were the first practical means of radio communication, used during the wireless telegraphy era which ended around 1920, Siwiak et al. (2004). Jradel M., (2008) stated that when the neural fields are formulated to predict neural activity using brain anatomy, one is led to the damped wave equation. One common application of the damped wave equation is in the study of musical instruments, Peterson, M.R (2004), such as the violin, classical guitar. Other physical phenomena that have been modeled by the damped wave equation include potential differences in nerve axons and fluid flow. In such a way that the damped wave equation has great application in science and technology.

2.5 Finite Difference Methods (FDMs)

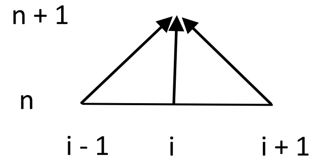
In real world, most of the problems in science and engineering, for example problems described by PDEs are complicated enough that they can only be solved numerically using FDMs, Jain (2003). For systems in one dimension, like the transverse motion of a vibrating string, the use of FDMs leads to a recurrence equation that simulates the propagation along the string, A. Chaigne (1992).

Among the number of numerical methods available for the solution of PDEs, FDMs are more gainfully employed than others and particularly well suited and reliable for solving hyperbolic equations in the time domain, A.R. Mitchell, D.F. Grifliths (1980) & S.S Sastry (2006). A finite difference scheme is produced when the partial derivatives in the PDE governing a physical phenomenon are replaced by a finite difference approximation. The result is a single algebraic equation or a system of algebraic equations which, when solved, provide an approximation to the solution of the original PDE at selected points of a solution grid, Gilberto E. Urroz (2004).

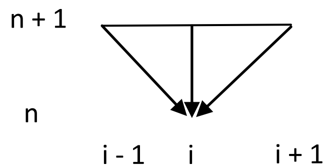
The most common FDMs for the numerical solution of PDE are explicit, implicit (simple) and Crank-Nicolson FDMs. These are closely related but differ in stability, consistency, accuracy and execution speed.

In this study, we are restricted to use explicit and implicit FDMs in solving one dimensional damped wave equation. In explicit finite difference schemes, the wave at $n + 1$ time step depends explicitly on the wave at n time step. The main advantage of explicit FDM is that it is relatively simple and computationally fast. However, the main drawback is that stable solutions are obtained only when $0< c\frac{\Delta t}{\Delta x} \leq 1$. If the condition is not satisfied, the solutions become unstable. In implicit finite difference schemes, the spatial derivative $\frac{\partial^2 u}{\partial x^2}$ is evaluated at the new time step. The main advantage of this method is that there is no restriction on the time step. Taking large time step, however, may result in an inaccurate solution.

(a) *Explicit scheme*:



(b) *Implicit scheme*:



(c) *Crank-Nicolson scheme*:

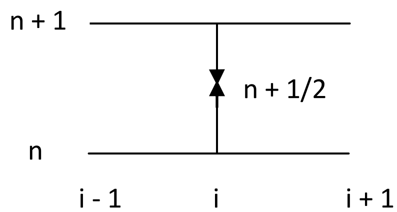


Figure 2: Explicit, Implicit and Crank-Nicolson schemes

The application of FDM to a particular differential equation problem includes the following steps:

1. Construction of discrete finite-difference scheme of the problem:

- coverage of the computational domain by a space-time grid, i.e. discretize the domain,
- approximations to derivatives, functions, initial and/or boundary conditions all at the grid points, i.e. replace the derivatives by difference equations, and
- construction of a linear system of the finite-difference (i.e. algebraic) equations.

2. Analysis of the finite-difference scheme:

- order of approximation,
- stability,
- consistency, and convergence.

3. Numerical computations (tests).

The FD solution of a PDE in the space-time domain is referred to as the Finite Difference Time-Domain (FDTD) method, Yee (1966). The domains are discretized into discrete space and time domains, with spacing of Δx and Δt, respectively. This is convenient to do space and time derivations. A discretized PDE, which is a function of space and time, will be formulated and discrete grids assigned. The computation then advances a discrete time, Δt, which is called a time step. It defines how far in time the model advances in an iteration. The time step determines the stability of the computation. Too large time steps cause instabilities in the simulation with inaccurate results. Overly small time steps waste computational time to achieve the same result as a coarser time step. This is therefore a trade-off for the FDTD method.

FDM works by replacing the region over which the independent variables in the PDE are defined by the finite grid of points at which the dependent variable is approximated. The solution domain of the problem is covered by a mesh of grid lines,

$$x_i = i\Delta x,\ i = 0, 1, 2, \ldots, \mathrm{M} \text{ and } t_n = n\Delta t,\ n = 0, 1, 2, \ldots, \mathrm{N} \qquad (2.2)$$

parallel to the space and time coordinate axes, respectively.

Approximations u_i^n to $u\ (i\Delta x\ ,\ n\Delta t)$ are calculated at the point of intersection of these lines, namely, $(i\Delta x\ ,\ n\Delta t)$ which is referred to as the $(i\ ,\ n)$ grid point. The constant spatial and temporal grid-spacing are $\Delta x = \frac{\mathrm{L}}{\mathrm{M}}$ and $\Delta t = \frac{\mathrm{T}}{\mathrm{N}}$, respectively, where L and T are the physical domains of space and time while M and N are the number of discrete points sampled over these domains, *Jake Wood*.

FDMs require the following initial information:

1. PDE, e.g. the 1D damped wave equation: $\frac{\partial^2 u}{\partial t^2} + 2\mathrm{k}\frac{\partial u}{\partial t} = \mathrm{c}^2\frac{\partial^2 u}{\partial x^2}$
2. Space domain, e.g. $0 \leq x_i \leq \mathrm{L}$, where $x_i \equiv i\Delta x$, i = 0,1,2,…,M, and L = some upper bound, e.g. π
3. Time domain, e.g. $0 \leq t_n \leq \mathrm{T}$, where $t_n \equiv n\Delta t$, $\mathrm{n} = 0,1,2, \ldots$, N, and T = sometime in the future, T can be arbitrarily large

4. ICs, the value of $u\ (x\ ,\ t)$ at t = 0, i.e. $u^{0}_{0,1,2,\ldots,\mathrm{M}}$, e.g. $u\ (x\ ,\ 0) = f(x)$ and $\frac{\partial u}{\partial t}\ (x\ ,\ 0) = g(x)$, for $0 \leq x \leq \mathrm{L}$
5. BCs, which specify the value and/or the derivative of $u\ (x\ ,\ t)$ at the ends or edges of the space domain. In one special dimension, there are two edge values for any time t, t: u_0^n and u_m^n. Their values are predetermined by specific functions of time; $u_0^n = f(n)$ and $u_m^n = \mathrm{g}(n)$.
   a) Dirichlet BCs are the simple case where both $f\,(n)$ and $g\,(n)$ are constants.
   b) Neumann BCs allow boundary values to fluctuate by setting u_0^n and u_m^n such that they obey the equation $u_x = 0$, *Jake Wood*.

❖ **Taylor Series Expansions:** It gives us a means to predict a function value at one point in terms of a function value and its derivatives at another point

- ***Forward Expansion***:

$$T_1 = \ u_{i+1}^n = u_i^n + \Delta x \frac{\partial u_i^n}{\partial x} + \frac{\Delta x^2}{2!}\frac{\partial^2 u_i^n}{\partial x^2} + \frac{\Delta x^3}{3!}\frac{\partial^3 u_i^n}{\partial x^3} + \ldots \tag{2.3}$$

- ***Backward Expansion***:

$$T_2 = \ u_{i-1}^n = u_i^n - \Delta x \frac{\partial u_i^n}{\partial x} + \frac{\Delta x^2}{2!}\frac{\partial^2 u_i^n}{\partial x^2} - \frac{\Delta x^3}{3!}\frac{\partial^3 u_i^n}{\partial x^3} + \ldots \tag{2.4}$$

Note: *For small* Δx *higher order terms can be neglected.*

$$u_{i+1}^n = u_i^n + \Delta x \frac{\partial u_i^n}{\partial x} + \frac{\Delta x^2}{2!}\frac{\partial^2 u_i^n}{\partial x^2} + \ldots + \frac{\Delta x^n}{n!}\frac{\partial^n u_i^n}{\partial x^n} + \frac{\Delta x^{n+1}}{(n+1)!}\frac{\partial^{n+1} u_i^n}{\partial x^{n+1}} \tag{2.5}$$

n-order accurate Truncation error

$$T_1: \frac{\partial u_{i,n}}{\partial x} = \frac{u_{i+1,n} - u_{i,n}}{\Delta x} - \frac{\Delta x}{2!}\frac{\partial^2 u_{i,n}}{\partial x^2} - \frac{\Delta x^2}{3!}\frac{\partial^3 u_{i,n}}{\partial x^3} - \ldots \tag{26}$$

Forward difference Truncation error $O\ (\Delta x)$

$$T_2: \frac{\partial u_{i,n}}{\partial x} = \frac{u_{i,n} - u_{i-1,n}}{\Delta x} \tag{2.7}$$

Backward difference Truncation error $O\ (\Delta x)$

$$T_1 - T_2: \frac{\partial u_{i,n}}{\partial x} = \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} - \frac{\Delta x^2}{3!}\frac{\partial^3 u_i^n}{\partial x^3} - \ldots \tag{2.8}$$

Central difference Truncation error $O\ (\Delta x)^2$

$$T_1 + T_2: \frac{\partial^2 u_{i,n}}{\partial x^2} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2} - \frac{2\Delta x^2}{4!}\frac{\partial^4 u_i^n}{\partial x^4} + \ldots \tag{2.9}$$

Central difference Truncation error $O\ (\Delta x)^2$, Shipra Agarwal (pp.9-10).

2.6 Thomas Algorithm for Tri-diagonal System

Consider the system of linear simultaneous algebraic equations given by

$$AX = b, \tag{2.10}$$

where A is a tri-diagonal matrix, X = $[x_1, x_2, \ldots, x_n]^T$ and b = $[b_1, b_2, \ldots, b_n]^T$. Hence we consider an n x n tri-diagonal system of equations given by

$$\begin{bmatrix} d_1 & u_1 & 0 & \cdots & & 0 \\ l_2 & d_2 & u_2 & \ddots & & \vdots \\ 0 & l_3 & d_3 & u_3 & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ & & & & & u_{n-1} \\ 0 & \cdots & 0 & & l_n & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ . \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ . \\ . \\ . \\ b_n \end{bmatrix} \tag{2.11}$$

The system of equations given by (2.11) is solved using Thomas Algorithm which is described in three steps as follows.

Step 1: Set $y_1 = d_1$ and complete

$$y_i = d_i - \frac{l_i u_{i-1}}{y_{i-1}}, \text{ for } i = 2, 3, \ldots, n$$

Step 2: Set $z_1 = \frac{b_1}{d_1}$ and complete

$$z_i = \frac{b_i - l_i z_{i-1}}{y_i}, \text{ for } i = 2, 3, \ldots, n$$

Step 3: $x_i = z_i - \frac{u_i x_{i+1}}{y_i}$, for $i = n-1, n-2, \ldots, 1$, where $x_n = z_n$, Rao V. Dukkipati (2010).

CHAPTER THREE

3 Methods and procedures

In this chapter we mainly focus on the study design, source of information, methods of data analysis, procedures and instrumentation.

3.1 Study design

The design of this study is documentary and experimental, to solve finite difference approximation on a discretized domain to get the numerical solution of one dimensional damped wave equation of a piece of thin flexible vibrating string which is firmly secured at the ends $x = 0$, $x = L$, at some supports; assuming the set-up has damping considering friction, i.e., the reaction of the surrounding medium, air resistance and density of the string on the motion of the string.

In the study, we design to apply explicit and implicit finite difference schemes in solving damped wave equation; by defining PDE, FDMs, discretizing the domain, replacing PDE by discrete functions, solving systems of equations using MATLAB code or Thomas Algorithm. Stability, consistence, convergence and truncation error of each method is analyzed. Tables & graphs are presented working with MATLAB code.

3.2 Source of information

To conduct this study, we use different reference books and some down-loaded materials, such as journals and articles that are related to the applications of FDMs on partial differential hyperbolic equations and damped wave equation. Applied numerical methods working with MATLAB and other MATLAB books are also used for the study.

3.3 Data analysis

In this study, analysis of numerical solutions of damped wave equation of each method is done. Stability, consistence, convergence and truncation error of each method is analyzed. Tables and graphs are presented working with MATLAB code.

3.4 Procedures of the study

1. demonstrating the PDE,
2. discretization of the domain,
3. replacing the partial derivatives by difference approximations,
4. solving Tri-diagonal system,
5. writing Mat Lab code,
6. comparisons of the solutions, and so on are procedures of the study.

3.5 Instrumentation

The way of approximation of the numerical solution of wave equation is introduced in the study using explicit and implicit finite difference methods. We show how to use the these methods to get the numerical solutions of damped wave equations of a piece of a thin flexible vibrating string, representing them by tri-diagonal matrix for their solutions. Stability, convergence and consistency of the solution of each method are analyzed. Computing the solutions of each method is performed working with MATLAB code or Thomas Algorithm. Plotting the graphs of solutions of each method and exact solution is performed working with MATLAB code. Finally, the exact and numerical solutions of the methods can be compared using tables and graphs.

CHAPTER FOUR

4 Results and Discussion

4.1 A mathematical model of damped wave equation on a vibrating string

Consider a piece of thin flexible (elastic) string, such as a violin string. We place the string along the x-axis, stretch it to length L, and firmly secured (fasten) it at the ends x = 0 and x = L, at some supports, so they will not move. We then distort the string initially from a position u = f and u_t = g parallel to the y axis at some instant, call it t = 0, we release it and allow it to vibrate. Then taking the motion of such a vibrating string and starting from general principles, a mathematical model of small transverse vibrations of an elastic string, such as a violin string can be derived. In order to model a wave traveling in a medium, we need to derive some equations governing the motion of such materials. The displacement (deflection) of the string at the point $x \in (0\ , L)$ and time $t > 0$ is given by u (x , t).

We make the following physical assumptions in order to proceed the derivation. The mass per unit length of the string, ρ [gm/cm], is constant (homogeneous material).The tension T is a force and assume that when the string is plucked the tension remain constant throughout the string. This is the same as assuming that the displacement is small for a homogeneous string. Also, assume that the force of gravity is much weaker than tension ($\rho Lg \ll$ T) so that it does not affect the motion of the string appreciably and can, therefore, be neglected, Ravi P. Agarwal & Donal O' Regan (2009).

The string is perfectly flexible, i.e., the string does not resist the motion of the wave. This means that the forces exerted on the string are tangent at the points where they act. We assume a damping force which is proportional to the velocity of the string.

We introduce the following notations:

$u = u\,(x\,,\,t\,)$: Transverse displacement of the string (cm),

$\rho = \rho\,(x)$: Linear density of string (gm/cm),

T = T (x): Tension in string at pint x (dyne),

$\omega = \omega\,(x, t\,)$: Transverse external force per unit length (dyne/cm),

$u_t = \frac{\partial u}{\partial t}$: Transverse velocity of the string (cm/sec),

$u_{tt} = \frac{\partial^2 u}{\partial t^2}$: Transverse acceleration of the string (cm/sec^2),

β = Linear velocity damping force per unit length (dyne/cm^2).

To derive the model, we consider an element of the string with density ρ per unit length located between x and $x + \Delta x$, and the forces acting on a small portion of the string (a line element $[x\,,\,x + \Delta x]$ of length $0 < \Delta x \ll 1$) as illustrated in Figure 3. Because of the model assumptions, these will be tensile forces, i.e., tangential to the curve of the string, Erwin Kreyszing (2006).

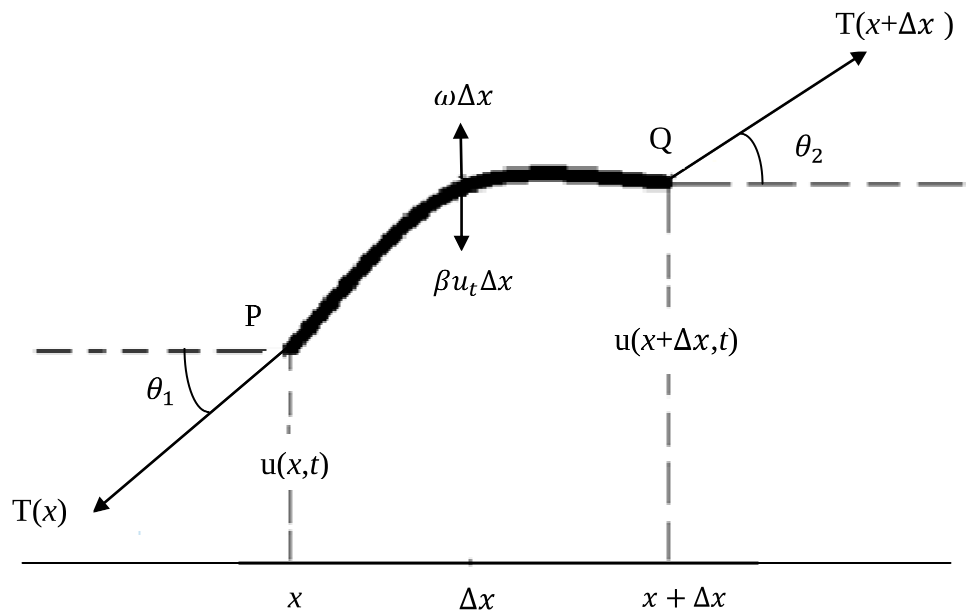


Figure 3: Section of vibrating string showing forces exerted on it

We find the derivatives $\frac{\partial u}{\partial x}$ (x, t) and $\frac{\partial u}{\partial x}$ $(x + \Delta x, t)$ represent the slopes of the tangent lines at the end points of Δx section. In terms of the angles $\theta_1$and θ_2 these slopes are represented,

$$\tan\theta_1 = \frac{\partial u}{\partial x}(x, t) \text{ and } \tan\theta_2 = \frac{\partial u}{\partial x}(x + \Delta x, t) \quad (4.1)$$

The external forces acting on this Δx section of the string are approximated by,

$\omega\Delta x$ = External force (dyne) and

$\beta u_t\, \Delta x$ = Damping force (dyne).

Our next assumption is that a point on the string moves only in the vertical direction. This means that the sum of the forces on the small piece of the string in x-direction is zero. We may equivalently assume that the horizontal component of the tension is constant. Thus, we have

$$T(x+\Delta x)\cos\theta_2 - T(x)\cos\theta_1 = 0 \quad (4.2)$$

This implies

$$T(x)\cos\theta_1 = T(x+\Delta x)\cos\theta_2 = T \tag{4.3}$$

Or simply $T_1\cos\theta_1 = T_2\cos\theta_2$ = T, where T_1 = T(x), T_2 = T(x+Δx) and T is a constant.

In the vertical direction we apply Newton's second law of motion that mass times acceleration must equal to the sum of the forces acting on the string. Defining the linear density ρ to be the mass of the string per unit length, the mass of the small piece of the string we are considering is given by $\rho\Delta x$. We sum forces at the center point of the string element to obtain

$$T(x+\Delta x)\sin\theta_2 - T(x)\sin\theta_1 + \omega(x+\frac{\Delta x}{2})\Delta x - \beta\Delta x\frac{\partial u}{\partial t}(x+\frac{\Delta x}{2}, t) = \rho(x + \frac{\Delta x}{2})\Delta x\frac{\partial^2 u}{\partial t^2}(x+\frac{\Delta x}{2}, t), \tag{4.4}$$

Where simply

$$T_2\sin\theta_2 - T_1\sin\theta_1 + \omega\Delta x - \beta\Delta x\frac{\partial u}{\partial t} = \rho\Delta x\frac{\partial^2 u}{\partial t^2} \tag{4.5}$$

$$\Rightarrow T\frac{\sin\theta_2}{\cos\theta_2} - T\frac{\sin\theta_1}{\cos\theta_1} + \omega\Delta x - \beta\Delta x\frac{\partial u}{\partial t} = \rho\Delta x\frac{\partial^2 u}{\partial t^2} \tag{4.6}$$

$$\Rightarrow T\tan\theta_2 - T\tan\theta_1 + \omega\Delta x - \beta\Delta x\frac{\partial u}{\partial t} = \rho\Delta x\frac{\partial^2 u}{\partial t^2} \tag{4.7}$$

Dividing both sides of (4.7) by Δx and T, we have

$$\frac{\tan\theta_2 - \tan\theta_1}{\Delta x} + \frac{\omega}{T} - \frac{\beta}{T}\frac{\partial u}{\partial t} = \frac{\rho}{T}\frac{\partial^2 u}{\partial t^2} \tag{4.8}$$

It follows from (4.1) that (4.8) becomes

$$\left[\frac{\frac{\partial u}{\partial x}(x+\Delta x,t) - \frac{\partial u}{\partial x}(x,t)}{\Delta x}\right] + \frac{\omega}{T} - \frac{\beta}{T}\frac{\partial u}{\partial t} = \frac{\rho}{T}\frac{\partial^2 u}{\partial t^2} \tag{4.9}$$

This implies

$$\left[\frac{\frac{\partial u}{\partial x}(x+\Delta x,t) - \frac{\partial u}{\partial x}(x,t)}{\Delta x}\right] = \frac{\rho}{T}\frac{\partial^2 u}{\partial t^2} - \frac{\omega}{T} + \frac{\beta}{T}\frac{\partial u}{\partial t} \tag{4.10}$$

In the limit as $\Delta x \to 0$, the left hand side of (4.10) becomes a partial derivative of u with respect to *x*, and because of our considerations were independent of the horizontal position of the line element, we finally obtain

$$\frac{\partial^2 u}{\partial x^2} = \frac{\rho}{T}\frac{\partial^2 u}{\partial t^2} - \frac{\omega}{T} + \frac{\beta}{T}\frac{\partial u}{\partial t} \tag{4.11}$$

$$\frac{\rho}{T}\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + \frac{\omega}{T} - \frac{\beta}{T}\frac{\partial u}{\partial t}$$

$$\frac{\partial^2 u}{\partial t^2} = \frac{T}{\rho}\frac{\partial^2 u}{\partial x^2} + \frac{\omega}{\rho} - \frac{\beta}{\rho}\frac{\partial u}{\partial t}$$

$$\frac{\partial^2 u}{\partial t^2} = c^2\frac{\partial^2 u}{\partial x^2} + \frac{\omega}{\rho} - \frac{\beta}{\rho}\frac{\partial u}{\partial t} \tag{4.12}$$

Where $c^2 = \frac{T}{\rho}$, c $= \sqrt{\frac{T}{\rho}}$ [m/sec], wave speed, which is a constant dependent on the physical conditions of the problem. Various special cases of (4.12) can be listed, Dean G. Duffy (1998).

4.1.1 Case1: One dimensional wave equation

The density, ρ and tension, T are constants with $\beta = 0$ and $\omega = 0$, we obtain the one dimensional wave equation. The wave equation will model the position $u = u(x, t)$ of any point *x* of the string, where $0 < x <$ L at any time $t > 0$. In order to get a particular solution to 1D wave equation, we must specify BCs and ICs. Since the string is fastened at the ends $x = 0$ and $x =$ L, i.e., the displacement *u* is zero, we have the two **boundary conditions**:

$$u(0, t) = 0 \text{ and } u(L, t) = 0, \text{ for } t > 0$$

Furthermore, the form of the motion of the string will depend on its initial shape of the string i.e., *its initial displacement* (displacement at time, $t = 0$), call it $f(x)$, and *its initial velocity* (velocity at $t = 0$), call it $g(x)$. We thus have the two **initial conditions:**

$$u(x, 0) = f(x) \text{ and } \frac{\partial u}{\partial t}(x, 0) = g(x), \text{ for } 0 < x < L$$

Our model for the vibrating string is then given by the following initial and boundary value problems.

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}, \qquad (4.13)$$

Subject to BCs: $u(0, t) = u(L, t) = 0,\ t > 0,$ (4.14)

and ICs: $u(x, 0) = f(x)$ and $\frac{\partial u}{\partial t}(x, 0) = g(x),\ 0 < x < L$ (4.15)

4.1.2 Case 2: One dimensional damped wave equation

The reaction of the surrounding medium, air resistance for example, density of the string and non-elastic effects in the string will contribute to reduce the amplitudes of the wave so that the motion dies out after some time, which is damping. Our model does not attain the same amplitude as the measured data over time. In fact the amplitude decreases over time. This physical phenomenon is known as damping. One way to take damping into account in our model is to add in a damping force. A damping force is a force that resists the motion of the wave acting in the opposite direction of the motion. We would expect a model for a damped wave traveling in a string to be similar to the un damped model with some additional decay factor. Assuming that the damping force at a point on the string is proportional to the velocity of the string, u_t and the density, ρ and tension, T are constants with $\beta/\rho = 2k$ and $\omega = 0$, we arrive at the telegraph equation, i.e., the wave equation with damping.

$$\frac{\partial^2 u}{\partial t^2} + 2k\frac{\partial u}{\partial t} = c^2 \frac{\partial^2 u}{\partial x^2}, \qquad (4.16)$$

Subject to BCs: $u(0, t) = u(L, t) = 0,\ t > 0,$ (4.17)

and ICs: $u(x, 0) = f(x)$ and $\frac{\partial u}{\partial t}(x, 0) = g(x),\ 0 < x < L$ (4.18)

where, the physical constant T/ρ is denoted by c^2 (instead of c), that is $c^2 = T/\rho$ to indicate that this constant is positive, a fact that will be essential to the form of the solutions, Donald C. Armstead (2004).

- c = $\sqrt{T/\rho}$, wave speed, depends on the physical condition of the problem.
- k is the damping coefficient (friction term), normally be determined from the physical experiments of the string.
- The term $2k\,\frac{\partial u}{\partial t}$, presents a damping force proportional to the elements velocity $\frac{\partial u}{\partial t}$.

Its solution is the displacement function $u(x\ ,\ t)$ defined for all values of x from 0 to L and all $t > 0$, satisfying the BCs and ICs, Donald C. Armstead (2004), Christoph Kirsch (2011). The problem is to determine the vibrations of the string, that is, to find the vertical displacement of the string, $0 < x < L$ (or its deflection at any point x), given by the displacement function $u\ (x\ ,\ t)$ at any time, $t > 0$, that satisfies the one dimensional damped wave equation (4.16) subject to the BCs and ICs given by equations (4.17) and (4.18). “One dimensional” means that the equation involves only one space variable, x.

4.2 Finite difference methods for damped wave motion

4.2.1 Simulation of damped waves on a finite string

We begin our study of wave motions by simulating one dimensional damped wave on a finite string; say on a guitar or violin string considering the reaction of the surrounding medium, air resistance, density of the string and non-elastic effects in the string. Let the string be in the deformed state coincide with the interval [0 , L] on the x-axis, and let $u\ (x\ ,\ t)$ be the displacement at time t in the y-direction of a point initially at x. The displacement function u is governed by the mathematical model that we have derived in the equation (4.16). The solution $u(x\ ,\ t)$ varies in space and time and describes waves that are moving with velocity c. The PDE problems (4.16), (4.17), (4.18) will now be discretized in space and time by a finite difference method.

4.2.2 Discritizing the domain

The temporal domain [0 , T] is represented by a finite number of mesh points

$$0 = t_0 < t_1 < t_2 < \cdots < t_{N-1} < t_N = T \tag{4.19}$$

Similarly, the spatial domain [0 , L] is replaced by a set of mesh points

$$0 = x_0 < x_1 < x_2 < \cdots < x_{M-1} < x_M = L \tag{4.20}$$

One may view the mesh as two-dimensional in the x, t plane, consisting of points $(x_{i\,,}\, t_n)$, with

$i = 0, 1, 2, \ldots, M-1, M$ and $n = 0, 1, 2, \ldots, N-1, N$

4.2.3 Uniform meshes

For uniformly distributed mesh points we can introduce the constant mesh spacing's Δt and Δx. We have that

$$x_i = i\Delta x\,,\ \ i = 0\,, 1, 2, \ldots, M \text{ and } t_n = n\Delta t\,,\ \ n = 0\,, 1, 2, \ldots, N \tag{4.21}$$

We also have that

$$\Delta x = x_i - x_{i-1},\ i = 0, 1, 2, \ldots, M-1, M \text{ and } \Delta t = t_n - t_{n-1},\ n = 0, 1, 2, \ldots, N-1, N \tag{4.22}$$

4.2.4 The discrete solution

The solution $u\ (x\ ,\ t)$ is sought at the mesh points. We introduce the mesh function u_i^n, which approximates the exact solution at the mesh points $(x_{i\,,}\, t_n)$ for $i = 0, 1, \ldots, M$ and $n = 0, 1, \ldots, N$. Using the finite difference method, we shall develop algebraic equations for computing the mesh function. One can often refers to the algebraic equations as discrete equations, finite difference equations or a finite difference scheme.

4.2.5 Fulfilling the equation at the mesh points

For a numerical solution by the finite difference method, we relax the condition that (4.16) holds at all points in the space-time domain (0 , L) x (0 , T] to the requirement that the PDE is fulfilled at the interior mesh points (x_i , t_n):

$$\frac{\partial^2 u}{\partial t^2}(x_i , t_n) + 2k\frac{\partial u}{\partial t}(x_i, t_n) = c^2\frac{\partial^2 u}{\partial x^2}(x_i, t_n),\ 0 < x_i < L,\ t_n > 0 \qquad (4.23)$$

for i = 1, 2, …, M−1 and n = 1, 2, …, N−1. For n = 0 we have the ICs $u = f(x)$ & $\frac{\partial u}{\partial t} = g(x) = 0$, and at the boundaries i = 0 & i = M we have the BC u = 0.

4.2.6 Replacing derivatives by finite differences

The second order derivatives can be replaced by central differences, which yield more accurate approximation for both time & space derivatives. The second order derivative in the t-direction is

$$\frac{\partial^2 u}{\partial t^2}(x_i , t_n) \approx \frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{\Delta t^2}, \qquad (4.24)$$

and a similar approximation of the second order derivative in the x-direction reads

$$\frac{\partial^2 u}{\partial x^2}(x_i , t_n) \approx \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} \qquad (4.25)$$

And the first order derivative can be replaced by central difference approximations as follows respectively, CF Chan Man Fong D De Kee (2003).

$$\frac{\partial u}{\partial t}(x_i , t_n) \approx \frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} \qquad (4.26)$$

4.2.7 Algebraic version of the 1D damped wave equation

We can now replaced the derivatives in (4.16) by central difference approximations and get

for **explicit** method

$$\frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{\Delta t^2} + 2k\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = c^2\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2}, \qquad (4.27)$$

and for **implicit** method

$$\frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{\Delta t^2} + 2k\frac{u_i^{n+1} - u_i^n}{2\Delta t} = c^2\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta x^2} \quad (4.28)$$

4.2.8 Algebraic version of the initial conditions

We also need to replace the derivative in the initial condition (4.18) by a finite difference approximation. A central difference $\frac{\partial u}{\partial t}(x_i, t_n) \approx \frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t}$ is appropriate. Writing out $\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = 0$ and ordering the terms gives

$$u_i^{n+1} = u_i^{n-1}, \text{ for } i = 0, 1, 2, \ldots, \text{M and } n = 0 \quad (4.29)$$

The other initial condition can be computed by

$$u_i^0 = u(x_i, 0) = f(x_i),\ i = 0, 1, 2, \ldots, \text{M}. \quad (4.30)$$

4.2.9 Formulating a recursive algorithm

We assume that u_i^{n-1} and u_i^n are already computed for i = 0, 1, 2, …, M. The only unknown quantity in (4.27) and (4.28) is therefore, u_i^{n+1}, which can be solved. Given that u_i^{n-1} and u_i^n we find new values at the next time level, Hans Petter Langtangen (2013).

4.3 Explicit and Implicit FDMs to solve damped wave equation

We now consider equation (4.16), subject to the conditions (4.17) and (4.18) [in order for (4.16) to be solvable]. Select an integer $\text{M} > 0$ and an integer $\text{N} > 0$ to define the x-axis and the t-axis grid points using $\Delta x = \frac{\text{L}}{\text{M}}$ and $\Delta t = \frac{\text{T}}{\text{N}}$ respectively. Then the mesh points (x_i, t_n) are defined by

$$x_i = i\Delta x,\ \text{i} = 0, 1, 2, \ldots, \text{M} \quad (4.31)$$

$$t_n = n\Delta t,\ \text{n} = 0, 1, 2, \ldots, \text{N}. \quad (4.32)$$

At any interior mesh points (4.16) becomes (4.23).

4.3.1 Explicit Finite Difference Method

Consider the following central difference approximations of the derivatives

$$\frac{\partial^2 u}{\partial t^2}(x_i, t_n) = \frac{u(x_i, t_{n+1}) - 2u(x_i, t_n) + u(x_i, t_{n-1})}{\Delta t^2} - \frac{(\Delta t)^2}{12}\frac{\partial^4 u}{\partial t^4}(x_i, \mu_n) + O(\Delta t^2) \tag{4.33}$$

$$\frac{\partial u}{\partial t}(x_i, t_n) = \frac{u(x_i, t_{n+1}) - u(x_i, t_{n-1})}{2\Delta t} - \frac{(\Delta t)^2}{6}\frac{\partial^3 u}{\partial t^3}(x_i, \mu_n) + O(\Delta t^2) \tag{4.34}$$

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_n) = \frac{u(x_{i+1}, t_n) - 2u(x_i, t_n) + u(x_{i-1}, t_n)}{\Delta x^2} - \frac{(\Delta x)^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i, t_n) + O(\Delta x^2), \tag{4.35}$$

where $\mu_n \epsilon (t_{n-1}, t_{n+1})$ and $\xi_i \epsilon (x_{i-1}, x_{i+1})$ for n = 1, 2, …, N−1and i = 1, 2, …, M−1.

Replacing these terms in (4.16) we obtain,

$$\frac{u(x_i, t_{n+1}) - 2u(x_i, t_n) + u(x_i, t_{n-1})}{\Delta t^2} - \frac{(\Delta t)^2}{12}\frac{\partial^4 u}{\partial t^4}(x_i, \mu_n) + O(\Delta t^2) + 2\mathrm{k}\Big[\frac{u(x_i, t_{n+1}) - u(x_i, t_{n-1})}{2\Delta t} - \frac{(\Delta t)^2}{6}\frac{\partial^3 u}{\partial t^3}(x_i, \mu_n)$$

$$+ O(\Delta t^2)\Big] = c^2\Big[\frac{u(x_{n+1}, t_n) - 2u(x_i, t_n) + u(x_{n-1}, t_n)}{\Delta x^2} - \frac{(\Delta t)^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i, t_n) + O(\Delta x^2)\Big] \tag{4.36}$$

By omitting all terms of $\{(\Delta t)^2, (\Delta x)^2\}$ we have

$$\frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{\Delta t^2} + 2\mathrm{k}\Big[\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t}\Big] = c^2\Big[\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2}\Big]$$

$$\Rightarrow u_i^{n+1} - 2u_i^n + u_i^{n-1} + \mathrm{k}\Delta t(u_i^{n+1} - u_i^{n-1}) = \lambda^2(u_{i+1}^n - 2u_i^n + u_{i-1}^n),$$

where $\lambda = c\,\frac{\Delta t}{\Delta x}$ (4.37)

$$\Rightarrow u_i^{n+1} = \frac{1}{1+\mathrm{k}\Delta t}[\lambda^2 u_{i-1}^n + 2(1-\lambda^2)u_i^n + \lambda^2 u_{i+1}^n)] - \frac{1-\mathrm{k}\Delta t}{1+\mathrm{k}\Delta t}u_i^{n-1}$$

$$\Rightarrow u_i^{n+1} = \mathrm{a}[\lambda^2 u_{i-1}^n + 2(1-\lambda^2)u_i^n + \lambda^2 u_{i+1}^n)] - \mathrm{b}u_i^{n-1}, \tag{4.38}$$

where $\mathrm{a} = \frac{1}{1+\mathrm{k}\Delta t}$ and $\mathrm{b} = \frac{1-\mathrm{k}\Delta t}{1+\mathrm{k}\Delta t}$ (4.39)

Equation (4.38) holds for each n = 1, 2, …, N and i = 1, 2, …, M−1.

The total of the truncation errors in using (4.38) instead of (4.16) is,

$$\mathrm{E}\{u_i^n\} = \tau_i^n = \frac{1}{12}\left[\Delta t^2 \frac{\partial^4 u}{\partial t^4}(x_i\,,\mu_n) + 2\mathrm{k}\Delta t^2 \frac{\partial^3 u}{\partial t^3}(x_i\,,\mu_n) - \mathrm{c}^2 \Delta x^2 \frac{\partial^4 u}{\partial t^4}(\xi_i\,,t_n) + O\{(\Delta t)^2,(\Delta x)^2\}\right] \quad (4.40)$$

This may be written as

$$\mathrm{E}\{u\} = O\{(\Delta t)^2,(\Delta x)^2\} \quad (4.41)$$

In this method the IC, $g(x) = 0$. In (4.38), when $n = 0$ we need u_i^{-1} which is calculated as

$$\frac{\partial u}{\partial t}(x_i\,,0) = g(x) = 0 \;\Rightarrow\; \frac{\partial u}{\partial t}(x_i\,,0) \;\Rightarrow\; \frac{u_i^1 - u_i^{-1}}{2\Delta t} = 0 \;\Rightarrow\; u_i^1 = u_i^{-1} \quad (4.42)$$

Then from (4.38) when $n = 0$, we have

$u_i^1 = \mathrm{a}[\lambda^2 u_{i-1}^0 + 2(1-\lambda^2)u_i^0 + \lambda^2 u_{i+1}^0)] - \mathrm{b}u_i^{-1}$,

$\Rightarrow (1+b)u_i^1 = \mathrm{a}[\lambda^2 u_{i-1}^0 + 2(1-\lambda^2)u_i^0 + \lambda^2 u_{i+1}^0)]$

$\Rightarrow u_i^1 = \frac{\mathrm{a}}{1+b}[\lambda^2 u_{i-1}^0 + 2(1-\lambda^2)u_i^0 + \lambda^2 u_{i+1}^0)]$

$$\Rightarrow u_i^1 = \frac{1}{2}[\lambda^2 u_{i-1}^0 + 2(1-\lambda^2)u_i^0 + \lambda^2 u_{i+1}^0)], \quad (4.43)$$

where $\frac{\mathrm{a}}{1+b} = \frac{1}{2}$, for $i = 1, 2, \ldots, \mathrm{M}-1$. (4.44)

Then we use (4.43) together with the IC, $u\,(x_i\,,0) = f(x_i)$, $0 \le x_i \le \mathrm{L}$ to get u_i^1 for each $i = 1, 2, \ldots, \mathrm{M}-1$and $n = 0$; then go on with (4.38) for $n = 1, 2, \ldots, \mathrm{N}$.

For *n* = 0: $u_i^1 = \frac{1}{2}[\lambda^2 u_{i-1}^0 + 2(1-\lambda^2)u_i^0 + \lambda^2 u_{i+1}^0)]$,

$i = 1$, $u_1^1 = \frac{1}{2}[\lambda^2 u_0^0 + 2(1-\lambda^2)u_1^0 + \lambda^2 u_2^0)]$, $u_0^0 = 0$

$i = 2$, $u_2^1 = \frac{1}{2}[\lambda^2 u_1^0 + 2(1-\lambda^2)u_2^0 + \lambda^2 u_3^0)]$

⋮

i = M−1, $u_1^1 = \frac{1}{2}[\lambda^2 u_{m-2}^0 + 2(1-\lambda^2)u_{m-1}^0 + \lambda^2 u_m^0)]$, $u_m^0 = 0$

where $u_i^0 = u(x_i, 0) = f(x_i)$, (4.45)

for each i = 0, 1, 2, …, M.

Matrix form of this set gives:

$$\begin{bmatrix} u_1^1 \\ u_2^1 \\ u_3^1 \\ . \\ . \\ . \\ u_{m-1}^1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 2(1-\lambda^2) & \lambda^2 & 0\cdots & & 0 \\ \lambda^2 & 2(1-\lambda^2) & \lambda^2 & \ddots & \vdots \\ 0 & \lambda^2 & 2(1-\lambda^2) & \lambda^2 & \\ \vdots & \ddots & \ddots & \ddots & \ddots \quad 0 \\ & & & & \lambda^2 \\ 0 & \cdots & 0 & \lambda^2 & 2(1-\lambda^2) \end{bmatrix}\begin{bmatrix} u_1^0 \\ u_2^0 \\ u_3^0 \\ . \\ . \\ . \\ u_{m-1}^0 \end{bmatrix} \quad (4.46)$$

Then from (4.38), for n = 1, 2, …, N and i = 1, 2, …, M−1 we have the following.

For *n* = 1: $u_i^2 = \mathrm{a}[\lambda^2 u_{i-1}^1 + 2(1-\lambda^2)u_i^1 + \lambda^2 u_{i+1}^1)] - \mathrm{b}u_i^0$, then

i = 1, $u_1^2 = \mathrm{a}[\lambda^2 u_0^1 + 2(1-\lambda^2)u_1^1 + \lambda^2 u_2^1)] - \mathrm{b}u_1^0$

i = 2, $u_2^2 = \mathrm{a}[\lambda^2 u_1^1 + 2(1-\lambda^2)u_2^1 + \lambda^2 u_3^1)] - \mathrm{b}u_2^0$

$\vdots$

i = M−1, $u_{m-1}^2 = \mathrm{a}[\lambda^2 u_{m-2}^1 + 2(1-\lambda^2)u_{m-1}^1 + \lambda^2 u_m^1)] - \mathrm{b}u_{m-1}^0$

For *n* = 2: $u_i^3 = \mathrm{a}[\lambda^2 u_{i-1}^2 + 2(1-\lambda^2)u_i^2 + \lambda^2 u_{i+1}^2)] - \mathrm{b}u_i^1$, then

i = 1, $u_1^3 = \mathrm{a}[\lambda^2 u_0^2 + 2(1-\lambda^2)u_1^2 + \lambda^2 u_2^2)] - \mathrm{b}u_1^1$

i = 2, $u_2^3 = \mathrm{a}[\lambda^2 u_1^2 + 2(1-\lambda^2)u_2^2 + \lambda^2 u_3^2)] - \mathrm{b}u_2^1$

$\vdots$

i = M−1, $u_{m-1}^3 = \mathrm{a}[\lambda^2 u_{m-2}^2 + 2(1-\lambda^2)u_{m-1}^2 + \lambda^2 u_m^2)] - \mathrm{b}u_{m-1}^1$

In general, for i = 1, 2, …, M−1, equation (4.38) can be expressed as follows:

i = 1, u_1^{n+1} = a$[\lambda^2 u_0^n + 2(1-\lambda^2)u_1^n + \lambda^2 u_2^n\ )] - \mathrm{b}u_1^{n-1}$, u_0^n = 0

i = 2, u_2^{n+1} = a$[\lambda^2 u_1^n + 2(1-\lambda^2)u_2^n + \lambda^2 u_3^n)] - \mathrm{b}u_2^{n-1}$

$$\vdots$$

i = M−1, u_{m-1}^{n+1} = a$[\lambda^2 u_{m-2}^n + 2(1-\lambda^2)u_{m-1}^n + \lambda^2 u_m^n)] - \mathrm{b}u_{m-1}^{n-1}$, u_m^n = 0

Writing this set of equations in matrix form gives:

$$\begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ . \\ . \\ . \\ u_{m-1}^{n+1} \end{bmatrix} = \mathrm{a} \begin{bmatrix} 2(1-\lambda^2) & \lambda^2 & 0 & \cdots & 0 \\ \lambda^2 & 2(1-\lambda^2) & \lambda^2 & \ddots & \\ 0 & \lambda^2 & 2(1-\lambda^2) & \lambda^2 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots\ 0 \\ & & & & \lambda^2 \\ 0 & \cdots & 0 & \lambda^2 & 2(1-\lambda^2) \end{bmatrix} \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ . \\ . \\ . \\ u_{m-1}^n \end{bmatrix} - \mathrm{b} \begin{bmatrix} u_1^{n-1} \\ u_2^{n-1} \\ u_3^{n-1} \\ . \\ . \\ . \\ u_{m-1}^{n-1} \end{bmatrix} \qquad (4.47)$$

where n = 1, 2, …, N.

The procedure using this formula will be referred to as the (1, 3, 1) method. This means that the molecule involves 1grid point at the (n+1) time level, 3 grid points at the (n) time level and 1grid point at the (n−1) time level. The computational molecule of this formula is shown in Figure 4.

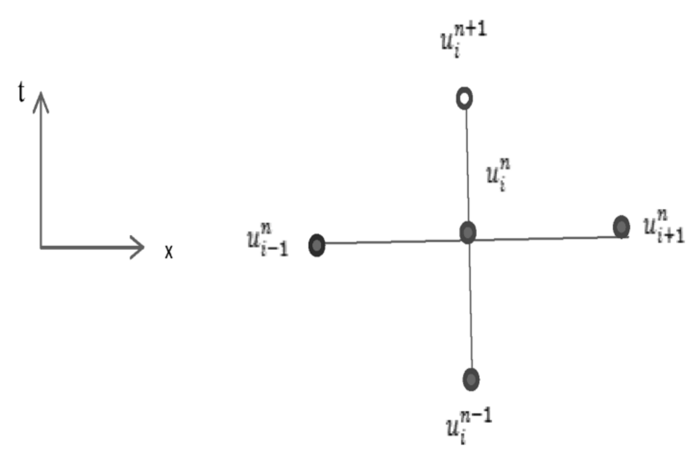


Figure 4: The computational molecule for (1, 3, 1) scheme

We can now summarize the computational algorithm:

1. Compute $u(x_i, 0) = u_i^0 = f(x_i)$ for each i = 0, 1, 2, …, M.
2. Compute u_i^1 by (4.43) for i = 1, 2, …, M−1 and set u_i^1 = 0 for the boundary points i = 0 and i = M.
3. For each time level n = 1, 2, …, N
   a) apply (4.47) to find u_i^{n+1} for i = 1, 2, …, M−1
   b) set u_i^{n+1} = 0 for the boundary points i = 0 and i = M.

The algorithm essentially consists of moving a finite difference stencil through all the mesh points.

4.3.2 Implicit Finite Difference Method

We now consider the following central difference approximations of the derivatives for solving the equation (4.16) subject to the BCs and ICs given.

$$\frac{\partial^2 u}{\partial t^2}(x_i, t_n) = \frac{u(x_i, t_{n+1}) - 2u(x_i, t_n) + u(x_i, t_{n-1})}{\Delta t^2} - \frac{(\Delta t)^2}{12}\frac{\partial^4 u}{\partial t^4}(x_i, \mu_n) + O(\Delta t^2) \tag{4.48}$$

$$\frac{\partial u}{\partial t}(x_i, t_n) = \frac{u(x_i, t_{n+1}) - u(x_i, t_{n-1})}{2\Delta t} - \frac{(\Delta t)^2}{6}\frac{\partial^3 u}{\partial t^3}(x_i, \mu_n) + O(\Delta t^2) \tag{4.49}$$

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_n) = \frac{u(x_{i+1}, t_{n+1}) - 2u(x_i, t_{n+1}) + u(x_{i-1}, t_{n+1})}{\Delta x^2} - \frac{(\Delta x)^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i, t_n) + O(\Delta x^2), \tag{4.50}$$

where $\mu_n \in (t_{n-1}, t_{n+1})$ and $\xi_i \in (x_{i-1}, x_{i+1})$ for n = 1, 2, …, N-1and i = 1, 2, …, M−1.

Substituting these difference approximations in (4.16) and neglecting the error term,

$$\mathrm{E}\{u_i^n\} = \tau_i^n = \frac{1}{12}\left[\Delta t^2 \frac{\partial^4 u}{\partial t^4}(x_i, \mu_n) + 2\mathrm{k}\Delta t^2 \frac{\partial^3 u}{\partial t^3}(x_i, \mu_n) - \mathrm{c}^2 \Delta x^2 \frac{\partial^4 u}{\partial t^4}(\xi_i, t_n) + O\{(\Delta t)^2, (\Delta x)^2\}\right]$$

leads to the difference equation

$$\frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{\Delta t^2} + 2\mathrm{k}\left[\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t}\right] = c^2\left[\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta x^2}\right]$$

$$\Rightarrow u_i^{n+1} - 2u_i^n + u_i^{n-1} + \mathrm{k}\Delta t(u_i^{n+1} - u_i^{n-1}) = \lambda^2(u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}),$$

$$\Rightarrow \frac{-1}{2}\lambda^2 u_{i-1}^{n+1} + \left(\frac{1+\text{k}\Delta t}{2} + \lambda^2\right) u_i^{n+1} - \frac{1}{2}\lambda^2 u_{i+1}^{n+1} + \left(\frac{1-\text{k}\Delta t}{2}\right) u_i^{n-1} = u_i^n$$

$$\Rightarrow \frac{-1}{2}\lambda^2 u_{i-1}^{n+1} + (p + \lambda^2)u_i^{n+1} - \frac{1}{2}\lambda^2 u_{i+1}^{n+1} + q u_i^{n-1} = u_i^n \ , \qquad (4.51)$$

where $\lambda = c\,\frac{\Delta t}{\Delta x}$, p = $\frac{1+\text{k}\Delta t}{2}$ and q = $\frac{1-\text{k}\Delta t}{2}$ (4.52)

Equation (4.51) holds for each n = 1, 2, …, N and i = 1, 2, …, M−1.

Also the total of the truncation errors in using (4.51) instead of (4.16) is,

$$\text{E}\{u_i^n\} = \tau_i^n = \frac{1}{12}\left[\Delta t^2 \frac{\partial^4 u}{\partial t^4}(x_i\,,\mu_n) + 2\text{k}\Delta t^2 \frac{\partial^3 u}{\partial t^3}(x_i\,,\mu_n) - \text{c}^2\Delta x^2 \frac{\partial^4 u}{\partial t^4}(\xi_i\,,t_n) + O\{(\Delta t)^2,(\Delta x)^2\}\right]$$

This may be written as

$$\text{E}\ \{u\} = O\{(\Delta t)^2,(\Delta x)^2\}$$

And in this method the IC, $g(x)$ = 0. Since u_i^{-1} is not given, we cannot also get u_i^1 directly from (4.51) when n = 0. So, when n = 0 we need u_i^{-1} calculated and shown in (4.42). Therefore, when n = 0, (4.51) gives

$$\frac{-1}{2}\lambda^2 u_{i-1}^1 + (p + \lambda^2)u_i^1 + \frac{-1}{2}\lambda^2 u_{i+1}^1 + (q)u_i^{-1} = u_i^0$$

$$\Rightarrow \frac{-1}{2}\lambda^2 u_{i-1}^1 + (p + q + \lambda^2)u_i^1 + \frac{-1}{2}\lambda^2 u_{i+1}^1 = u_i^0$$

$$\Rightarrow \frac{-1}{2}\lambda^2 u_{i-1}^1 + (1 + \lambda^2)u_i^1 + \frac{-1}{2}\lambda^2 u_{i+1}^1 = d_i \ , \qquad (4.53)$$

where $d_i = u_i^0$ = u (x_i , 0) = f (x_i) and p + q = 1, for i = 1, 2, …, M−1. (4.54)

We now use (4.53) together with the BCs $u_0^n = u_m^n = 0$ and ICs, for n = 0, to get set of equations of u_i^1, and its matrix form.

For *n* = 0: $\frac{-1}{2}\lambda^2 u_{i-1}^1 + (1 + \lambda^2)u_i^1 + \frac{-1}{2}\lambda^2 u_{i+1}^1 = d_i$, we have

i = 1, $\frac{-1}{2}\lambda^2 u_0^1 + (1 + \lambda^2)u_1^1 + \frac{-1}{2}\lambda^2 u_2^1 = d_1$, u_0^1 = 0

$i = 2, \frac{-1}{2}\lambda^2 u_1^1 + (1+\lambda^2)u_2^1 + \frac{-1}{2}\lambda^2 u_3^1 = d_2$

$$\vdots$$

$i = \text{M}-1, \frac{-1}{2}\lambda^2 u_{m-2}^1 + (1+\lambda^2)u_{m-1}^1 + \frac{-1}{2}\lambda^2 u_m^1 = d_{M-1}$, $u_m^1 = 0$

Writing this system in matrix form gives:

$$\begin{bmatrix} 1+\lambda^2 & \frac{-1}{2}\lambda^2 & 0 & \cdots & 0 \\ \frac{-1}{2}\lambda^2 & 1+\lambda^2 & \frac{-1}{2}\lambda^2 & \ddots & \vdots \\ 0 & \frac{-1}{2}\lambda^2 & 1+\lambda^2 & \frac{-1}{2}\lambda^2 & \\ \vdots & \ddots & \ddots & \ddots & \ddots \; 0 \\ & & & & \frac{-1}{2}\lambda^2 \\ 0 & \cdots & 0 & \frac{-1}{2}\lambda^2 & 1+\lambda^2 \end{bmatrix} \begin{bmatrix} u_1^1 \\ u_2^1 \\ u_3^1 \\ . \\ . \\ . \\ u_{m-1}^1 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ . \\ . \\ . \\ d_{m-1} \end{bmatrix} \qquad (4.55)$$

where $d_i = u_i^0 = u(x_i, 0) = f(x_i)$, for $i = 1, 2, \ldots, \text{M}-1$.

Then from (4.51), we have

$$\frac{-1}{2}\lambda^2 u_{i-1}^{n+1} + (p+\lambda^2)u_i^{n+1} + \frac{-1}{2}\lambda^2 u_{i+1}^{n+1} = u_i^n - q u_i^{n-1}$$

$$\Rightarrow \frac{-1}{2}\lambda^2 u_{i-1}^{n+1} + (p+\lambda^2)u_i^{n+1} + \frac{-1}{2}\lambda^2 u_{i+1}^{n+1} = d_i, \qquad (4.56)$$

where $d_i = u_i^n - q u_i^{n-1}$, for $n = 1, 2, \ldots, \text{N}$ & $i = 1, 2, \ldots, \text{M}-1$. (4.57)

For $n = 1$: $\frac{-1}{2}\lambda^2 u_{i-1}^2 + (p+\lambda^2)u_i^2 + \frac{-1}{2}\lambda^2 u_{i+1}^2 = d_i$, then

$i = 1, \frac{-1}{2}\lambda^2 u_0^2 + (p+\lambda^2)u_1^2 + \frac{-1}{2}\lambda^2 u_2^2 = d_1$, $u_0^2 = 0$

$i = 2, \frac{-1}{2}\lambda^2 u_1^2 + (p+\lambda^2)u_2^2 + \frac{-1}{2}\lambda^2 u_3^2 = d_2$

$$\vdots$$

$i = \text{M}-1, \frac{-1}{2}\lambda^2 u_{m-2}^2 + (p+\lambda^2)u_{m-1}^2 + \frac{-1}{2}\lambda^2 u_m^2 = d_{M-1}$, $u_m^2 = 0$

Matrix form of this system is:

$$\begin{bmatrix} p+\lambda^2 & \frac{-1}{2}\lambda^2 & 0 & \cdots & 0 \\ \frac{-1}{2}\lambda^2 & p+\lambda^2 & \frac{-1}{2}\lambda^2 & \ddots & \vdots \\ 0 & \frac{-1}{2}\lambda^2 & p+\lambda^2 & \frac{-1}{2}\lambda^2 & \\ \vdots & \ddots & \ddots & \ddots & \ddots\ 0 \\ & & & & \frac{-1}{2}\lambda^2 \\ 0 & \cdots & 0 & \frac{-1}{2}\lambda^2 & p+\lambda^2 \end{bmatrix} \begin{bmatrix} u_1^2 \\ u_2^2 \\ u_3^2 \\ . \\ . \\ . \\ u_{m-1}^2 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ . \\ . \\ . \\ d_{m-1} \end{bmatrix} \tag{4.58}$$

$$\vdots$$

In general, for i = 1, 2, …, M−1 equation (4.56) can be expressed as follows.

$i = 1,\ \frac{-1}{2}\lambda^2 u_0^{n+1} + (p+\lambda^2)u_1^{n+1} + \frac{-1}{2}\lambda^2 u_2^{n+1} = d_1,\ u_0^{n+1} = 0$

$i = 2,\ \frac{-1}{2}\lambda^2 u_1^{n+1} + (p+\lambda^2)u_2^{n+1} + \frac{-1}{2}\lambda^2 u_3^{n+1} = d_2$

$$\vdots$$

$i = M-1,\ \frac{-1}{2}\lambda^2 u_{m-2}^{n+1} + (p+\lambda^2)u_{m-1}^{n+1} + \frac{-1}{2}\lambda^2 u_m^{n+1} = d_{M-1},\ u_m^{n+1} = 0$

Matrix form of the system gives:

$$\begin{bmatrix} p+\lambda^2 & \frac{-1}{2}\lambda^2 & 0 & \cdots & 0 \\ \frac{-1}{2}\lambda^2 & p+\lambda^2 & \frac{-1}{2}\lambda^2 & \ddots & \vdots \\ 0 & \frac{-1}{2}\lambda^2 & p+\lambda^2 & \frac{-1}{2}\lambda^2 & \\ \vdots & \ddots & \ddots & \ddots & \ddots\ 0 \\ & & & & \frac{-1}{2}\lambda^2 \\ 0 & \cdots & 0 & \frac{-1}{2}\lambda^2 & p+\lambda^2 \end{bmatrix} \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ . \\ . \\ . \\ u_{m-1}^{n+1} \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ . \\ . \\ . \\ d_{m-1} \end{bmatrix} \tag{4.59}$$

where $d_i = u_i^n - qu_i^{n-1}$, for n = 1, 2, …, N & i = 1, 2, …, M−1.

Similar to the first method, this method has also (1, 3, 1) molecule scheme.

We can also summarize the computational algorithm as follows.

1. Compute $u_i^0 = u\ (x_i\ , 0) = f(x_i)$ for each i = 0, 1, 2, …, M.
2. For time level n = 0, apply (4.53) to compute u_i^1, where $d_1 = u_i^0 = u(x_i\ , 0) = f(x_i)$ for i = 1, 2, …, M−1.
3. For each time level n = 1, 2, …, N, apply (4.59) to compute u_i^{n+1}, where

   $d_i = u_i^n - qu_i^{n-1}$, for i = 1, 2, …, M−1.

Also the algorithm essentially consists of moving a finite difference stencil through all the mesh points.

4.4 Stability, consistency and convergence analysis of the methods

The most important properties that every finite difference approximation of a PDE should possess are stability, consistency and convergence. These notions cover different aspects of the relation between the partial differential equation and its finite difference equation.

❖ **Truncation Error**

Truncation error is defined as the error that results from using an approximation in place of exact mathematical procedure. Truncation error results from terminating after a finite number of terms. Therefore the error between the exact solution and the numerical solution is determined by the error that is committed by going from a differential operator to a difference operator is called the truncation error or the discretization error. The term truncation error reflects the fact that a finite part of a Taylor series is used in the approximation.

4.4.1 Stability and Von Neumann Stability Analysis (VNSA)

Stability means that the errors at any stage of computation are not amplified but are attenuated as the computation progresses, C. Chapra, p.875 (2010). That means that the solution of the difference equation is not too sensitive to small perturbations in the initial data. The choices of Δx and Δt are the most important components of a stable numerical solution. Running a simulation with unstable parameters causes the calculated solutions to diverge rapidly. When possible it is a good idea to perform basic stability analysis, the most common measures being awareness of the Courant - Friedrichs - Lewy (CFL) condition, i.e., $\frac{\Delta x}{\Delta t} \geq$ c (i.e; the wave speed of the numerical schemes be at least as large as the wave speed of the PDE) and (VNSA).

VNSA works on finite difference equations describing linear PDEs and relies on Fourier analysis to determine the constraints on the CFL condition that preserve stability, Dean. G. Duffy (1998). (The CFL condition, λ is the term that collects the time step, space size and velocity terms into one variable for a given finite difference scheme.) A solution for the error equation is now sought in the variable separable form

$$\xi_j^n = |G|^n e^{i\beta j}, \tag{4.60}$$

in which

$$\mathrm{G} = e^{\beta_m \Delta t} \text{ and } \beta = m\pi\Delta x, \tag{4.61}$$

where the time dependence of this Fourier component of the error at $x = i\Delta x$ and $t = n\Delta t$, is contained in the coefficient G^n which is the nth power of the complex number G. The term $e^{i\beta j} = e^{im\pi xj} = \cos(m\pi x) + i\sin(m\pi x)$ is bounded and, therefore, any growth in the numerical solution will arise from the term G, known as the amplification factor.

I. Stability Analysis for Explicit Scheme

Now, considering (4.38), for explicit FDM, i.e.,

$$u_i^{n+1} = \frac{1}{1+\mathrm{k}\Delta t}\left[\lambda^2 u_{i-1}^n + 2(1-\lambda^2)u_i^n + \lambda^2 u_{i+1}^n\right] - \frac{1-\mathrm{k}\Delta t}{1+\mathrm{k}\Delta t} u_i^{n-1}$$

$$\Rightarrow u_i^{n+1} = \frac{1}{1+\mathrm{k}\Delta t}\left[\lambda^2 u_{i-1}^n + 2(1-\lambda^2)u_i^n + \lambda^2 u_{i+1}^n + (1-\mathrm{k}\Delta t)u_i^{n-1}\right] \tag{4.62}$$

By replacing ξ instead of u in (4.62), we obtain

$$\xi_j^{n+1} = \frac{1}{1+\mathrm{k}\Delta t}\left[\lambda^2 \xi_{j-1}^n + 2(1-\lambda^2)\xi_j^n + \lambda^2 \xi_{j+1}^n + (1-\mathrm{k}\Delta t)\xi_j^{n-1}\right] \tag{4.63}$$

Substituting (4.60) in (4.63) gives

$|G|^{n+1}e^{i\beta j}$= $\frac{1}{1+\mathrm{k}\Delta t}[\lambda^2|G|^n e^{i\beta(j-1)} + 2(1-\lambda^2)|G|^n e^{i\beta j} + \lambda^2|G|^n e^{i\beta(j+1)} + (1-\mathrm{k}\Delta t)|G|^{n-1}e^{i\beta j}]$(4.64)

Dividing both sides of (4.64) by $|G|^{n+1}e^{i\beta j}$, we obtain

$$1 = \frac{1}{1+\mathrm{k}\Delta t}\,[\lambda^2\frac{|G|^n e^{i\beta(j-1)}}{|G|^{n+1}e^{i\beta j}} + 2(1-\lambda^2)\frac{|G|^n e^{i\beta j}}{|G|^{n+1}e^{i\beta j}} + \lambda^2\frac{|G|^n e^{i\beta(j+1)}}{|G|^{n+1}e^{i\beta j}} + (1-\mathrm{k}\Delta t)\frac{|G|^{n-1}e^{i\beta j}}{|G|^{n+1}e^{i\beta j}}]$$

$$\Rightarrow 1 + \mathrm{k}\Delta t = [\lambda^2 e^{-i\beta}G^{-1} + 2(1-\lambda^2)G^{-1} + \lambda^2 e^{i\beta}G^{-1} + (1-\mathrm{k}\Delta t)G^{-2}]$$

$$\Rightarrow 1 + \mathrm{k}\Delta t = G^{-2}\,[\lambda^2 e^{-i\beta}G + 2(1-\lambda^2)G + \lambda^2 e^{i\beta}G + (1-\mathrm{k}\Delta t)]$$

$$\Rightarrow (1 + \mathrm{k}\Delta t)\,G^2 - [2(1-\lambda^2)G + \lambda^2(e^{i\beta} + e^{-i\beta})G + (1-\mathrm{k}\Delta t)] = 0$$

$$\Rightarrow (1 + \mathrm{k}\Delta t)\,G^2 - [2(1-\lambda^2) + 2\lambda^2(cos\beta)]G + (1-\mathrm{k}\Delta t) = 0$$

$$\Rightarrow (1 + \mathrm{k}\Delta t)\,G^2 - [2(1-\lambda^2) + 2\lambda^2(1 - 2sin^2\,\beta/2)]G + (1-\mathrm{k}\Delta t) = 0$$

$$\Rightarrow (1 + \mathrm{k}\Delta t)\,G^2 + [-2 + 4\lambda^2 sin^2\,\beta/2)]G + (1-\mathrm{k}\Delta t) = 0 \qquad (4.65)$$

For all three-level methods the Von Neumann amplification factor G may have one of two values which are the roots of a quadratic equation of the form

$$\mathrm{A}G^2 + \mathrm{B}\,G + \mathrm{D} = 0 \qquad (4.66)$$

In which the coefficients A, B, D are functions of β, λ and may be complex numbers. For coefficients of quadratic equation (4.66) the Von Neumann Stability is established by using the following criteria.

(a) If $|A| > |D|$, then the given finite difference formula is *stable* if and only if

$$|A|^2 - |D|^2 \geq |B\bar{A} - D\bar{B}|,$$

(b) If $|A| = |D|$ and $|B\bar{A} - D\bar{B}|$= 0, then the given finite difference formula is *Stable* if and only if

$$2|A| \geq |B|,$$

(c) If $|A| < |D|$, then the given finite difference formula is *unstable*, Saberi Najafi & Izadi (2014).

Consider (4.65), which may be written in the form (4.66) with

A = $1 + \mathrm{k}\Delta t$, B = $-2 + 4\lambda^2 sin^2\, \beta/2$ and D = $1 - \mathrm{k}\Delta t$.

Note that $1 > \mathrm{k}\Delta t > 0$, this implies $1 + \mathrm{k}\Delta t > 0$ and $1 - \mathrm{k}\Delta t > 0$. Since $|A|$ = $1 + \mathrm{k}\Delta t$ and $|D|$ = $1 - \mathrm{k}\Delta t$, then $|A| > |D|$ (4.67)

It follows from (a), that the equation (4.38) is *stable* if and only if

$$|A|^2 - |D|^2 \geq |B\bar{A} - D\bar{B}|$$

$$\Rightarrow |1 + \mathrm{k}\Delta t|^2 - |1 - \mathrm{k}\Delta t|^2 \geq \left|\left(-2 + 4\lambda^2 sin^2 \tfrac{\beta}{2}\right)\left(-(1 + \mathrm{k}\Delta t\,)\right) - (1 - \mathrm{k}\Delta t)(2 - 4\lambda^2 sin^2 \tfrac{\beta}{2})\right| \quad (4.68)$$

$$\Rightarrow (1 + \mathrm{k}\Delta t\,)^2 - (1 - \mathrm{k}\Delta t)^2 \geq \left|\left(2 - 4\lambda^2 sin^2 \tfrac{\beta}{2}\right)(1 + \mathrm{k}\Delta t) - (2 - 4\lambda^2 sin^2 \tfrac{\beta}{2})(1 - \mathrm{k}\Delta t)\right|$$

$$\Rightarrow 1 + 2\mathrm{k}\Delta t + \mathrm{k}^2\Delta t^2 - 1 + 2\mathrm{k}\Delta t - \mathrm{k}^2\Delta t^2 \geq \left|(2\mathrm{k}\Delta t)(2 - 4\lambda^2 sin^2 \tfrac{\beta}{2})\right|$$

$$\Rightarrow 4\mathrm{k}\Delta t \geq 4\mathrm{k}\Delta t \left|1 - 2\lambda^2 sin^2 \tfrac{\beta}{2}\right|$$

$$\Rightarrow 1 \geq \left|1 - 2\lambda^2 sin^2 \tfrac{\beta}{2}\right|$$

$$\Rightarrow \left|1 - 2\lambda^2 sin^2 \tfrac{\beta}{2}\right| \leq 1$$

$$\Rightarrow -1 \leq 1 - 2\lambda^2 sin^2 \tfrac{\beta}{2} \leq 1$$

$$\Rightarrow \; 0 \leq \lambda^2 sin^2 \tfrac{\beta}{2} \leq 1$$

$$\Rightarrow \; 0 \leq \lambda sin \tfrac{\beta}{2} \leq 1 \quad (4.69)$$

So the range of stability for this procedure is $0 < \lambda \leq 1$, for λ = c$\frac{\Delta t}{\Delta x}$ since 0 and 1 represent minima and maxima of the sine term for all β. This shows the scheme is stable.

Recall (4.38) that the explicit method can be written in matrix form as

$$u^{n+1}\text{=A}u^n\text{+}bu^{n-1},$$

where the tri-diagonal matrix A has $1-\lambda^2$ on diagonal and $\frac{1}{2}\lambda^2$ on the super and sub-diagonal. The norm of the matrix dictates how fast errors are growing (the vector bu^{n-1} does not come into play), B. Neta (2003).

If we check the infinity norm or the 1 norm we get

$$\|A\|_\infty = \|A\|_1 = |1/2\,\lambda^2| + |1-\ \lambda^2\ | + |1/2\,\lambda^2|$$

For $1-\lambda^2 \geq 0$ implies $0 < \lambda \leq 1$, all the numbers inside the absolute values are non-negative and we get a norm of 1.

For $1-\lambda^2 < 0$ implies $\lambda > 1$, the norm is $2\lambda^2 - 1$ which is greater than 1.Thus we have conditional stability with the condition $0 < \lambda \leq 1$. This implies the scheme conditionally stable.

II. Stability Analysis for Implicit Scheme

Also, for implicit FDM, considering (4.51), i.e.,

$$\frac{-1}{2}\lambda^2 u_{i-1}^{n+1} + (p+\lambda^2)u_i^{n+1} + \frac{-1}{2}\lambda^2 u_{i+1}^{n+1} + q u_i^{n-1} = u_i^n, \tag{4.70}$$

where p = $\frac{1+k\Delta t}{2}$ and q = $\frac{1-k\Delta t}{2}$

By replacing ξ instead of u in (4.70), we obtain

$$\frac{-1}{2}\lambda^2 \xi_{i-1}^{n+1} + (p+\lambda^2)\xi_i^{n+1} + \frac{-1}{2}\lambda^2 \xi_{i+1}^{n+1} + q\xi_i^{n-1} = \xi_i^n, \tag{4.71}$$

Substituting (4.60) in (4.71) gives

$$\frac{-1}{2}\lambda^2 |G|^{n+1} e^{i\beta(j-1)} + (p+\lambda^2)|G|^{n+1} e^{i\beta j} + \frac{-1}{2}\lambda^2 |G|^{n+1} e^{i\beta(j+1)} + q|G|^{n-1} e^{i\beta j} = |G|^n e^{i\beta j}, \tag{4.72}$$

Dividing both sides of (4.72) by $|G|^n e^{i\beta j}$, we obtain

$$\frac{-1}{2}\lambda^2 \frac{|G|^{n+1} e^{i\beta(j-1)}}{|G|^n e^{i\beta j}} + (p+\lambda^2)\frac{|G|^{n+1} e^{i\beta j}}{|G|^n e^{i\beta j}} - \frac{1}{2}\lambda^2 \frac{|G|^{n+1} e^{i\beta(j+1)}}{|G|^n e^{i\beta j}} + q\,\frac{|G|^{n-1} e^{i\beta j}}{|G|^n e^{i\beta j}} = 1$$

$$\Rightarrow \frac{-1}{2}\lambda^2 e^{-i\beta} G + (p+\lambda^2)\mathrm{G} - \frac{1}{2}\lambda^2 e^{i\beta} G + qG^{-1} = 1$$

$$\Rightarrow (p+\lambda^2)\mathrm{G} - \frac{1}{2}\lambda^2 (e^{i\beta} + e^{-i\beta})G + qG^{-1} = 1$$

$$\Rightarrow (p+\lambda^2)\mathrm{G} - \frac{1}{2}\lambda^2 (2cos\beta)G + qG^{-1} = 1$$

$$\Rightarrow \left(p + 2\lambda^2 sin^2\frac{\beta}{2}\right) G^2 - G + q = 0$$

$$\Rightarrow \left(\frac{1+k\Delta t}{2} + 2\lambda^2 sin^2\frac{\beta}{2}\right) G^2 - G + \frac{1-k\Delta t}{2} = 0$$

$$\Rightarrow \left(1 + k\Delta t + 4\lambda^2 sin^2\frac{\beta}{2}\right) G^2 - 2G + (1 - k\Delta t) = 0 \tag{4.73}$$

$$\Rightarrow AG^2 + BG + D = 0, \tag{4.74}$$

where $A = 1 + k\Delta t + 4\lambda^2 sin^2\frac{\beta}{2}$, $B = -2$, $D = 1 - k\Delta t$ (4.75)

Since $|A| = 1 + k\Delta t + 4\lambda^2 sin^2\frac{\beta}{2} > 1 - k\Delta t = |D| \Rightarrow |A| \geq |D|$ (4.76)

Then it follows from (a), that the equation (4.51) is stable if and only if

$$|A|^2 - |D|^2 \geq |B\bar{A} - D\bar{B}|$$

Since $0 \leq sin^2\frac{\beta}{2} \leq 1$, we have $1 + k\Delta t \leq A \leq 1 + k\Delta t + 4\lambda^2$

Now taking $A = 1 + k\Delta t$ (the minimum value of A), we obtain

$$(1 + k\Delta t)^2 - (1 - k\Delta t)^2 \geq |-2[-(1 + k\Delta t)] - (1 - k\Delta t)2| \tag{4.77}$$

$$\Rightarrow 4k\Delta t \geq |2(1 + k\Delta t) - 2(1 - k\Delta t)|$$

$$\Rightarrow 4k\Delta t \geq |4k\Delta t| \Rightarrow 1 \geq 1$$

And taking $A = 1 + k\Delta t + 4\lambda^2$ (the maximum value of A), we obtain

$$[(1 + k\Delta t) + 4\lambda^2]^2 - (1 - k\Delta t)^2 \geq |-2[-(1 + k\Delta t + 4\lambda^2)] - (1 - k\Delta t)2| \tag{4.78}$$

$$\Rightarrow (1 + k\Delta t)^2 + 8\lambda^2(1 + k\Delta t) + (4\lambda^2) - (1 - 2k\Delta t + k^2\Delta t^2) \geq |4k\Delta t + 8\lambda^2|$$

$$\Rightarrow 4k\Delta t + 8\lambda^2 k\Delta t + 16\lambda^4 + 8\lambda^2 \geq 4k\Delta t + 8\lambda^2$$

$$\Rightarrow k\Delta t + 2\lambda^2 \geq 0 \Rightarrow 2\lambda^2 > 0 \Rightarrow 0 < \lambda^2 \leq 1$$

$$\Rightarrow \quad 0 < \lambda \le 1 \tag{4.79}$$

Therefore the range of stability for this procedure is also $0 < \lambda \le 1$, for $\lambda = c\frac{\Delta t}{\Delta x}$.

Recall (4.59) that the implicit method can be written in matrix vector form

$$Au^{n+1} = b,$$

where the tri-diagonal matrix A have $\frac{1+k\Delta t}{2} + \lambda^2$ on diagonal and $\frac{-1}{2}\lambda^2$ on the super and sub-diagonal. If we check the infinity norm or the 1 norm we get

$$\|A\|_\infty = \|A\|_1 = \left|\frac{-1}{2}\lambda^2\right| + \left|\frac{1+k\Delta t}{2} + \lambda^2\right| + \left|\frac{-1}{2}\lambda^2\right|$$

For $0 < \lambda \le 1$ and $\lambda > 1$, the norm is $\frac{1+k\Delta t}{2} + 2\lambda^2$ which is greater than 1. Thus we have unconditional stability. This shows scheme is unconditionally stable.

4.4.2 Consistency

Definition: The finite difference approximation is said to be consistent with the differential equation if the truncation error goes to zero when the numerical parameters are made arbitrarily small. A numerical method is said to be consistent if

$$\lim_{\Delta x, \Delta t \to 0} \|\tau_i^n\| = 0, \text{ where } \tau_i^n \text{ is the truncation error.}$$

According to the interpolation of the truncation error stated, any consistent numerical method closely matches the differential equation when the step sizes Δx , Δt are sufficiently close to zero.

A finite-difference formula of a partial differential equation is said to be consistent with a partial differential equation, if in the limit as the grid spacing tend to zero, the finite-difference formula is identical to the partial differential equation at each point in the solution domain. Consistency deals with how well the finite difference equation approximates the partial differential equation and it is the necessary condition for convergence, Dean. G. Duffy (1998).

I. Consistency Analysis for Explicit Scheme

To researching the consistency of equation (4.16) by explicit finite-difference method, we consider the equation (4.38).

$$u_i^{n+1} = \frac{1}{1+k\Delta t}\left[\lambda^2 u_{i-1}^n + 2(1-\lambda^2)u_i^n + \lambda^2 u_{i+1}^n - (1-k\Delta t)u_i^{n-1}\right], \tag{4.80}$$

written in the form

$$L_\Delta\{u_j^n\} = (1+k\Delta t)u_j^{n+1} - \lambda^2 u_{j-1}^n - 2(1-\lambda^2)u_j^n - \lambda^2 u_{j+1}^n + (1-k\Delta t)u_j^{n-1} = 0 \tag{4.81}$$

Replacing u_j^n in (4.81) by the exact solution $\hat{u}_i^n$ of equation (4.16), we obtain

$$L_\Delta\{\hat{u}_i^n\} = (1+k\Delta t)\hat{u}_i^{n+1} - \lambda^2 \hat{u}_{i-1}^n - 2(1-\lambda^2)\hat{u}_i^n - \lambda^2 \hat{u}_{i+1}^n + (1-k\Delta t)\hat{u}_i^{n-1}$$

$$\Rightarrow L_\Delta\{\hat{u}_i^n\} = (\hat{u}_i^{n+1} - 2\hat{u}_i^n + \hat{u}_i^{n-1}) + k\Delta t(\hat{u}_i^{n+1} - \hat{u}_i^{n-1}) - \lambda^2(\hat{u}_{i+1}^n - 2\hat{u}_i^n + \hat{u}_{i-1}^n), \tag{4.82}$$

which may be no longer equal to zero. Since $\hat{u}_i^n$ is continuously differentiable, then terms of (4.82) may be replaced by their Taylor expansions about the point $(i\Delta x, n\Delta t)$. This gives,

$$L_\Delta\{\hat{u}_i^n\} = \Delta t^2\left\{\left[\frac{\partial^2 u}{\partial t^2}(x_i,t_n) + \frac{\Delta t^2}{12}\frac{\partial^4 u}{\partial t^4}(x_i,t_n) - \cdots\right] + 2k\left[\frac{\partial u}{\partial t}(x_i,t_n) + \frac{\Delta t^2}{6}\frac{\partial^3 u}{\partial t^3}(x_i,t_n) + \cdots\right] - c^2\left[\frac{\partial^2 u}{\partial x^2}(x_i,t_n) + \frac{\Delta x^2}{12}\frac{\partial^4 u}{\partial x^4}(x_i,t_n) - \cdots\right]\right\} = 0$$

$$\Rightarrow L_\Delta\{\hat{u}_i^n\} = \Delta t^2\left\{\frac{\partial^2 u}{\partial t^2}(x_i,t_n) + 2k\frac{\partial u}{\partial t}(x_i,t_n) - c^2\frac{\partial^2 u}{\partial x^2}(x_i,t_n) + \frac{\Delta t^2}{12}\frac{\partial^4 u}{\partial t^4}(x_i,t_n) - \cdots + k\frac{\Delta t^2}{3}\frac{\partial^3 u}{\partial t^3}(x_i,t_n) + \cdots - c^2\frac{\Delta x^2}{12}\frac{\partial^4 u}{\partial x^4}(x_i,t_n) - \cdots\right\} \tag{4.83}$$

$$\Rightarrow L_\Delta\{\hat{u}_i^n\} = \Delta t^2\left\{\frac{\partial^2 u}{\partial t^2}(x_i,t_n) + 2k\frac{\partial u}{\partial t}(x_i,t_n) - c^2\frac{\partial^2 u}{\partial x^2}(x_i,t_n) + \frac{\Delta t^2}{12}\frac{\partial^4 u}{\partial t^4}(x_i,\mu_n) + k\frac{\Delta t^2}{3}\frac{\partial^3 u}{\partial t^3}(x_i,\mu_n) - c^2\frac{\Delta x^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i,t_n)\right\}, \tag{4.84}$$

where

$$\mu_n \epsilon (t_{n-1}, t_{n+1}),\ \xi_i \epsilon (x_{i-1}, x_{i+1}) \tag{4.85}$$

And

$$\mathrm{E}\,\{\hat{u}_i^n\} = \frac{\Delta t^2}{12}\frac{\partial^4 u}{\partial t^4}(x_i\,,\mu_n) + \ k\frac{\Delta t^2}{3}\frac{\partial^3 u}{\partial t^3}(x_i\,,\mu_n) - c^2\frac{\Delta x^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i\,,t_n), \tag{4.86}$$

is the truncation error of the second-order accurate in time and space. When the grid spacing gets smaller and smaller with the explicit finite-difference method, the truncation error gets smaller and smaller at a fixed point (ξ_i, μ_n) in the solution domain. In the limit as $\Delta x \to 0$, $\Delta t \to 0$ the finite-difference formula (4.84) is equivalent to the partial differential equation (4.16); so the explicit finite-difference method is consistent.

II. Consistency Analysis for Implicit Scheme

And to see the consistency of equation (4.16) by implicit finite-difference method, consider the equation (4.51)

$$\frac{-1}{2}\lambda^2 u_{i-1}^{n+1} + (p+\lambda^2)u_i^{n+1} - \frac{1}{2}\lambda^2 u_{i+1}^{n+1} + q u_i^{n-1} = u_i^n\ , \tag{4.87}$$

Written in the form

$$L_\Delta\{u_j^n\} = (1 + k\Delta t + 2\lambda^2)u_j^{n+1} - \lambda^2(u_{j+1}^{n+1} + u_{j-1}^{n+1}) - 2u_j^n - u_{j+1}^n + (1 - k\Delta t)u_j^{n-1} = 0 \tag{4.88}$$

Replacing u_j^n in (4.88) by the exact solution $\hat{u}_i^n$ of equation (4.16), we get

$$L_\Delta\{\hat{u}_i^n\} = (1 + k\Delta t + 2\lambda^2)\hat{u}_i^{n+1} - \lambda^2(\hat{u}_{i+1}^{n+1} + \hat{u}_{i-1}^{n+1}) - 2\hat{u}_i^n + (1 - k\Delta t)\hat{u}_i^{n-1}$$

$$\Rightarrow L_\Delta\{\hat{u}_i^n\} = (\hat{u}_i^{n+1} - 2\hat{u}_i^n + \hat{u}_i^{n-1}) + k\Delta t(\hat{u}_i^{n+1} - \hat{u}_i^{n-1}) - \lambda^2(\hat{u}_{i+1}^{n+1} - 2\hat{u}_i^{n+1} + \hat{u}_{i-1}^{n+1}), \tag{4.89}$$

which may be no longer equal to zero. Since $\hat{u}_i^n$ is continuously differentiable, then terms of (4.89) may be replaced by their Taylor expansions about the point $(i\Delta x, n\Delta t)$. This gives,

$$L_\Delta\{\hat{u}_i^n\}=\Delta t^2\left\{\left[\frac{\partial^2 u}{\partial t^2}(x_i\,,t_n) + \frac{\Delta t^2}{12}\frac{\partial^4 u}{\partial t^4}(x_i\,,t_n) - \cdots\right] + 2k\left[\frac{\partial u}{\partial t}(x_i\,,t_n) + \frac{\Delta t^2}{6}\frac{\partial^3 u}{\partial t^3}(x_i\,,t_n) + \cdots\right] - c^2\left[\frac{\partial^2 u}{\partial x^2}(x_i\,,t_n) + \frac{\Delta x^2}{12}\frac{\partial^4 u}{\partial x^4}(x_i\,,t_n) - \cdots\right]\right\} = 0$$

$$\Rightarrow L_\Delta\{\hat{u}_i^n\}= \Delta t^2 \left\{\frac{\partial^2 u}{\partial t^2}(x_i\,,t_n) + 2k\frac{\partial u}{\partial t}(x_i\,,t_n) - c^2\frac{\partial^2 u}{\partial x^2}(x_i\,,t_n) + \frac{\Delta t^2}{12}\frac{\partial^4 u}{\partial t^4}(x_i\,,t_n) - \cdots + k\frac{\Delta t^2}{3}\frac{\partial^3 u}{\partial t^3}(x_i\,,t_n) + \cdots - c^2\frac{\Delta x^2}{12}\frac{\partial^4 u}{\partial x^4}(x_i\,,t_n) - \cdots\right\} \tag{4.90}$$

$$\Rightarrow L_\Delta\{\hat{u}_i^n\}= \Delta t^2 \left\{\frac{\partial^2 u}{\partial t^2}(x_i\,,t_n) + 2k\frac{\partial u}{\partial t}(x_i\,,t_n) - c^2\frac{\partial^2 u}{\partial x^2}(x_i\,,t_n) + \frac{\Delta t^2}{12}\frac{\partial^4 u}{\partial t^4}(x_i\,,\mu_n) + k\frac{\Delta t^2}{3}\frac{\partial^3 u}{\partial t^3}(x_i\,,\mu_n) - c^2\frac{\Delta x^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i\,,t_n)\right\}, \tag{4.91}$$

where $\mu_n \epsilon (t_{n-1}\,, t_{n+1})$, $\xi_i \epsilon (x_{i-1}\,, x_{i+1})$ (4.92)

And $$E\,\{\hat{u}_i^n\} = \frac{\Delta t^2}{12}\frac{\partial^4 u}{\partial t^4}(x_i\,,\mu_n) - k\frac{\Delta t^2}{3}\frac{\partial^3 u}{\partial t^3}(x_i\,,\mu_n) - c^2\frac{\Delta x^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i\,,t_n), \tag{4.93}$$

is the truncation error of the second-order accurate in time and space. Also when the grid spacing gets smaller and smaller with the implicit finite-difference method, the truncation error gets smaller and smaller at a fixed point (ξ_i, μ_n) in the solution domain. In the limit as $\Delta x \to 0$, $\Delta t \to 0$ the finite-difference formula (4.91) is equivalent to the partial differential equation (4.16); therefore the implicit finite-difference method is consistent.

4.4.3 Convergence

Convergence means that the solution of the difference equation approaches to the solution of partial differential equation as the computational mesh grid is refined, C. Chapra (2010).

Definition

A one-step finite difference scheme approximating a partial differential equation is a convergent scheme if for any solution to the partial differential equation, $u(x\,,t)$, and solutions to the finite difference scheme, u_i^n, such that u_i^0 converges to $u_0(x)$ as $i\Delta x$ converges to x, then u_i^n converges to $u(x\,,t)$ as $(i\Delta x\,, n\Delta t)$ converges to $(x\,,t)$ as Δx, Δt converge to 0.

The Lax-Richtmyer Equivalence Theorem

A consistent finite difference scheme for a partial differential equation for which the initial value problem is well posed is convergent if and only if it is consistent and stable.

Remark: The Lax-Richtmyer Equivalence Theorem is often called the *Fundamental Theorem of Numerical Analysis*, even though it is only applicable to the small subset of linear numerical methods for well-posed, linear PDEs; expresses a relationship between stability, consistency and convergence.

Consistency + stability $\Leftrightarrow$ convergence

So, as Lax-Richtmyer Equivalence Theorem states that for a consistent FDM of a well posed, linear initial and boundary value problems, the method is convergent if and only if it is stable. Therefore, convergence is shown by Lax-Richtmyer Equivalence Theorem; so we can conclude that both the schemes are convergent.

4.5 Numerical tests

In this section we apply the numerical schemes to solve the following example. The accuracy of the proposed numerical methods is measured by computing the difference between the analytic and the numerical solutions at some mesh point. The analytic solution of the equation (4.16) subject to BCs (4.17) and ICs (4.18) with L $=\pi$, $c^2 = 1$, which is obtained by the method of separation of variables, D.G. Duffy (2003), is given by

$$u(x, t) = \sum_{n=1}^{\infty} \sin(nx)\, e^{-kt}[a_n \cos(\sqrt{n^2-k^2}\, t) + b_n \sin(\sqrt{n^2-k^2}\, t)], \qquad (4.94)$$

where $a_n = \frac{2}{\pi}\int_0^{\pi} f(x)\sin(nx)dx$

and $b_n = \frac{1}{\sqrt{n^2-k^2}}(\mathrm{k}a_n + \frac{2}{\pi}\int_0^{\pi} \mathrm{g}(x)\sin(nx)dx) \qquad (4.95)$

are Fourier half-range sine expansions over the interval $(0, \pi)$, D.G. Duffy (2003).

Example to apply the schemes

Consider $\frac{\partial^2 u}{\partial t^2} + 2\mathrm{k}\frac{\partial u}{\partial t} = c^2\frac{\partial^2 u}{\partial x^2}$, $0 < x < \pi$, $0 \le t \le 1$, with (4.96)

ICs: $u(x, 0) = f(x) = \begin{cases} \frac{2x}{\pi}, & 0 \le x \le \frac{\pi}{2} \\ \frac{-2(x-\pi)}{\pi}, & \frac{\pi}{2} \le x \le \pi \end{cases}$ and $u_t(x, 0) = g(x) = 0$ (4.97)

BCs: $u(0, t) = u(\pi, t) = 0$ with L $= \pi$, $c^2 = 1$, k = 0.5 (4.98)

Now, let's solve the given example using the methods of explicit and implicit FDM applying their respective numerical schemes that we have derived. We may consider the following three cases for each method.

Case1: $\Delta t_1 = \frac{1}{32}$ and $\Delta x_1 = \frac{\pi}{100}$

Case2: $\Delta t_2 = \frac{\Delta t_1}{3} = \frac{1}{96}$ and $\Delta x_2 = \frac{\Delta x_1}{3} = \frac{\pi}{300}$

Case3: $\Delta t_3 = \frac{\Delta t_1}{5} = \frac{1}{160}$ and $\Delta x_3 = \frac{\Delta x_1}{5} = \frac{\pi}{500}$

Case1: $\boldsymbol{\Delta t_1 = \frac{1}{32}}$ **and** $\boldsymbol{\Delta x_1 = \frac{\pi}{100}}$

In this case for c = 1, k = 0.5, T = 1 and L = π, we have

$\lambda = c\frac{\Delta t_1}{\Delta x_1} = \frac{25}{8\pi}$ a = $\frac{1}{1+k\Delta t_1} = \frac{64}{65}$, b = $\frac{-1+k\Delta t_1}{1+k\Delta t_1} = \frac{-63}{65}$, p = $\frac{1+k\Delta t_1}{2} = \frac{65}{128}$ & q = $\frac{1-k\Delta t_1}{2} = \frac{63}{128}$

The initial values of the wave at the nodal points, $u_0^0, u_1^0, u_2^0, \ldots, u_{49}^0, u_{50}^0, u_{51}^0, \ldots, u_{100}^0$, using

$$u_i^0 = u(x_i, 0) = f(x_i) = \begin{cases} \frac{2x_i}{\pi}, 0 \le x_i \le \frac{\pi}{2} \\ \frac{-2(x_i-\pi)}{\pi}, \frac{\pi}{2} \le x_i \le \pi \end{cases},$$

where $x_i = i\Delta x_1 = \frac{i\pi}{100}$, i = 0, 1, 2, …, 100, become

$$u_0^0 = u(x_0, 0) = f(x_0) = f(0) = \frac{2}{\pi}(0) = 0.0000$$

$$u_1^0 = u(x_1, 0) = f(x_1) = f(\frac{\pi}{100}) = \frac{2}{\pi}(\frac{\pi}{100}) = 0.0200$$

$$u_2^0 = u(x_2, 0) = f(x_2) = f(\frac{2\pi}{100}) = \frac{2}{\pi}(\frac{2\pi}{100}) = 0.0400$$

$$\vdots$$

$$u_9^0 = u(x_9, 0) = f(x_3) = f(\frac{9\pi}{100}) = \frac{2}{\pi}(\frac{9\pi}{100}) = 0.1800$$

$$u_{10}^0 = u(x_{10}, 0) = f(x_3) = f(\frac{10\pi}{100}) = \frac{2}{\pi}(\frac{10\pi}{100}) = 0.2000$$

$$\vdots$$

$$u_{100}^0 = u(x_{100}, 0) = f(x_{100}) = f(\frac{100\pi}{100}) = \frac{-2}{\pi}(\frac{100\pi}{100} - \pi) = 0.0000$$

u_1^0	u_2^0	u_3^0	u_4^0	u_5^0	u_6^0	u_7^0	u_8^0	u_9^0	u_{10}^0
0.0200	0.0400	0.0600	0.0800	0.1000	0.1200	0.1400	0.1600	0.1800	0.2000

Table 1: Initial values of the vibrating string of case 1 at the 1st time step

N.B: Since the string is symmetrical with respect to the point $x_{50} = \frac{\pi}{2}$, it is satisfactory to find the values of $u(x_i, t_n)$ only for $0 \le x_i \le \frac{\pi}{2}$.

4.5.1 For explicit scheme

Using (4.46) and taking i = 1, 2, 3, …, 51 successively, we get the following matrix form to solve u_i^1.

$$\begin{bmatrix} u_1^1 \\ u_2^1 \\ u_3^1 \\ \cdot \\ \cdot \\ \cdot \\ u_{50}^1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0.0211 & 0.9895 & 0 & \cdots & & 0 \\ 0.9895 & 0.0211 & 0.9895 & & \ddots & \\ 0 & 0.9895 & 0.0211 & 0.9895 & & \vdots \\ & & & & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ & & & & & 0.9895 \\ 0 & \cdots & 0 & 0.9895 & & 0.0211 \end{bmatrix}\begin{bmatrix} 0.0200 \\ 0.0400 \\ 0.0600 \\ \cdot \\ \cdot \\ \cdot \\ 1.0000 \end{bmatrix}, \tag{4.99}$$

where $2(1-\lambda^2)$ = 0.0211 and λ^2 = 0.9895 (to the nearest four decimal places). Then solving u_i^1, i = 1, 2, …, 50 from (4.100) using MATLAB code, we have

u_1^1 = 0.0200, u_2^1 = 0.0400, u_3^1 = 0.0600, u_4^1 = 0.0800, u_5^1 = 0.1000, …, u_9^1 = 0.1800, u_{10}^1 =0.2010, …, u_{48}^1 = 0.9600, u_{49}^1 = 0.9800, u_{50}^1 = 0.4954

u_1^1	u_2^1	u_3^1	u_4^1	u_5^1	u_6^1	u_7^1	u_8^1	u_9^1	u_{10}^1
0.0200	0.0400	0.0600	0.0800	0.1000	0.1200	0.1400	0.1600	0.1800	0.2010

Table 2: Values of the vibrating string in case1of explicit scheme at the 1st time step

And we consider (4.47) to solve u_i^2, u_i^3, u_i^4, …, u_i^{32} taking i = 1, 2, 3, …, 50 successively for each u_i^{n+1} for n = 1, 2, 3, …, 31 using MATLAB code,

$$\begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ . \\ . \\ . \\ u_{m-1}^{n+1} \end{bmatrix} = \frac{192}{193} \begin{bmatrix} 0.0211 & 0.9895 & 0 & \cdots & & 0 \\ 0.9895 & 0.0211 & 0.9895 & & \ddots & \\ 0 & 0.9895 & 0.0211 & 0.9895 & & \vdots \\ & & & & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ & & & & & 0.9895 \\ 0 & \cdots & 0 & 0.9895 & & 0.0211 \end{bmatrix} \begin{bmatrix} u_1^{n} \\ u_2^{n} \\ u_3^{n} \\ . \\ . \\ . \\ u_{m-1}^{n} \end{bmatrix} - \frac{191}{193} \begin{bmatrix} u_1^{n-1} \\ u_2^{n-1} \\ u_3^{n-1} \\ . \\ . \\ . \\ u_{m-1}^{n-1} \end{bmatrix} \quad (4.100)$$

4.5.2 For implicit scheme

Using (4.55) and taking i = 1, 2, 3, …, 51 successively, we get its matrix vector form AX = B as follows, where A is a tri-diagonal matrix and B is a vector.

$$\begin{bmatrix} 1.9895 & -0.4947 & 0 & \cdots & & 0 \\ -0.4947 & 1.9895 & -0.4947 & & \ddots & \\ 0 & -0.4947 & 1.9895 & -0.4947 & & \vdots \\ & & & & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ & & & & & -0.4947 \\ 0 & \cdots & 0 & -0.4947 & & 1.9895 \end{bmatrix} \begin{bmatrix} u_1^{1} \\ u_2^{1} \\ u_3^{1} \\ . \\ . \\ . \\ u_{50}^{1} \end{bmatrix} = \begin{bmatrix} 0.0200 \\ 0.0400 \\ 0.0600 \\ . \\ . \\ . \\ 1.0000 \end{bmatrix}, \quad (4.101)$$

where $1+\lambda^2$ = 1.9895 and $\frac{-1}{2}\lambda^2$ = -0.4947 (to the nearest four decimal places), and $d_i = u_i^0 = u(x_i, 0) = f(x_i)$, for i = 1, 2, 3, …, 51. Then solving this system of equation AX = B using MATLAB code or Thomas Algorithm, we get the values of the first time-step nodal points, u_i^1 as u_1^1 = 0.0200, u_2^1 = 0.0400, u_3^1 = 0.0600, u_4^1 = 0.0800, u_5^1 = 0.1000, u_6^1 = 0.1200,…, u_9^1 = 0.1800, u_{10}^1 = 0.2007,…, u_{49}^1 =0.9076, u_{50}^1 = 0.7283

u_1^1	u_2^1	u_3^1	u_4^1	u_5^1	u_6^1	u_7^1	u_8^1	u_9^1	u_{10}^1
0.0200	0.0400	0.0600	0.0800	0.1000	0.1200	0.1400	0.1600	0.1800	0.2007

Table 3: Values of the vibrating string in case 1 of implicit scheme at the 1st time step

And considering (4.59), a matrix vector form AX = B, where A is a tri-diagonal matrix and B is a vector, we can solve u_i^{n+1} taking i = 1, 2, 3, …, 51 successively for n = 1, 2, 3, …, 31 using MATLAB code or Thomas Algorithm.

$$\begin{bmatrix} 1.4973 & -0.4947 & 0 & \cdots & & 0 \\ -0.4947 & 1.4973 & -0.4947 & & \ddots & \\ 0 & -0.4947 & 1.4973 & -0.4947 & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ & & & & & -0.4947 \\ 0 & \cdots & & 0 & -0.4947 & 1.4973 \end{bmatrix} \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ . \\ . \\ . \\ u_{m-1}^{n+1} \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ . \\ . \\ . \\ d_{m-1} \end{bmatrix}, \tag{4.102}$$

Where p + λ^2 = 1.4973, $\frac{-1}{2}\lambda^2 = -0.4947$(to the nearest four decimal places), $d_i = u_i^n - qu_i^{n-1}$, for n = 1, 2, …, 31 and i = 1, 2, …, 51, i.e., we can solve u_i^2, u_i^3, …, u_i^{32}, for n = 1, n = 2, …, n = 31, $d_i = u_i^1 - qu_i^0$, $d_i = u_i^2 - qu_i^1$, ..., $d_i = u_i^{31} - qu_i^{30}$ respectively; taking i = 1, 2, …, 51 successively.

Case2: $\Delta t_2 = \frac{\Delta t_1}{3} = \frac{1}{96}$ and $\Delta x_2 = \frac{\Delta x_1}{3} = \frac{\pi}{300}$

In this case for c = 1, k = 0.5, T = 1 and L = π, we have

λ = c $\frac{\Delta t_2}{\Delta x_2} = \frac{75}{24\pi}$, a = $\frac{1}{1+k\Delta t_2} = \frac{192}{193}$, b = $\frac{-1+k\Delta t_2}{1+k\Delta t_2} = \frac{-191}{193}$, q = $\frac{1-k\Delta t_2}{2} = \frac{191}{384}$ & p = $\frac{1+k\Delta t_2}{2} = \frac{193}{384}$

The initial values of the wave at the nodal points, u_0^0, u_1^0, u_2^0, u_3^0, …, u_{150}^0, u_{151}^0, …, u_{300}^0, using

$$u_i^0 = u(x_i, 0) = f(x_i) = \begin{cases} \frac{2x_i}{\pi}, 0 \le x_i \le \frac{\pi}{2} \\ \frac{-2(x_i-\pi)}{\pi}, \frac{\pi}{2} \le x_i \le \pi \end{cases},$$

where $x_i = i\Delta x_2 = \frac{i\pi}{300}$, i = 0, 1, 2, …, 300, become

$$u_0^0 = u(x_0, 0) = f(x_0) = f(0) = \frac{2}{\pi}(0) = 0.0000$$

$$u_1^0 = u(x_1, 0) = f(x_1) = f(\frac{\pi}{300}) = \frac{2}{\pi}(\frac{\pi}{300}) = 0.0067$$

$$u_2^0 = u(x_2, 0) = f(x_2) = f(\frac{2\pi}{300}) = \frac{2}{\pi}(\frac{2\pi}{300}) = 0.0133$$

$$\vdots$$

$$u_{10}^0 = u(x_{10}, 0) = f(x_{10}) = f(\frac{10\pi}{300}) = \frac{2}{\pi}(\frac{10\pi}{300}) = 0.0667$$

$$\vdots$$

$u_{300}^0 = u(x_{300}, 0) = f(x_{300}) = f(\frac{300\pi}{300}) = \frac{-2}{\pi}(\frac{300\pi}{300} - \pi) = 0.0000$

u_1^0	u_2^0	u_3^0	u_4^0	u_5^0	u_6^0	u_7^0	u_8^0	u_9^0	u_{10}^0
0.0067	0.0133	0.02000	0.0267	0.0333	0.0400	0.0467	0.0533	0.0600	0.0667

Table 4: Initial values of the vibrating string in case 2 at the 1st time step

N.B: Since the string is symmetrical with respect to the point $x_{150} = \frac{\pi}{2}$, it is satisfactory to find the values of $u(x_i, t_n)$ only for $0 \leq x_i \leq \frac{\pi}{2}$.

4.5.3 For explicit scheme

Using (4.46) and taking i =1, 2, 3, …, 151 successively, we get the following matrix form to solve u_i^1.

$$\begin{bmatrix} u_1^1 \\ u_2^1 \\ u_3^1 \\ . \\ . \\ . \\ u_{150}^1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0.0211 & 0.9895 & 0 & \cdots & & 0 \\ 0.9895 & 0.0211 & 0.9895 & & \ddots & \\ 0 & 0.9895 & 0.0211 & 0.9895 & & \vdots \\ & & & & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ & & & & & 0.9895 \\ 0 & \cdots & 0 & 0.9895 & 0.0211 & \end{bmatrix}\begin{bmatrix} 0.0067 \\ 0.0133 \\ 0.0200 \\ . \\ . \\ . \\ 1.0000 \end{bmatrix}, \tag{4.103}$$

where $2(1-\lambda^2) = 0.0211$ and $\lambda^2 = 0.9895$(to the nearest four decimal places). Then solving u_i^1, i = 1, 2, …, 150 from (4.104) using MATLAB code, we have

$u_1^1 = 0.0067$, $u_2^1 = 0.0134$, $u_3^1 = 0.0200$, $u_4^1 = 0.0267$, $u_5^1 = 0.0334$, $u_6^1 = 0.0400$, $u_7^1 = 0.0467$, $u_8^1 = 0.0534$, $u_9^1 = 0.0600$, $u_{10}^1 = 0.0667$, …, u_{150}^1

u_1^1	u_2^1	u_3^1	u_4^1	u_5^1	u_6^1	u_7^1	u_8^1	u_9^1	u_{10}^1
0.0067	0.0134	0.0200	0.0267	0.0334	0.0400	0.0467	0.0534	0.0600	0.0667

Table 5: Values of the vibrating string in case 2 of explicit scheme at the 1st time step

And we consider (4.47) to solve u_i^2, u_i^3, u_i^4, …, u_i^{96} taking i = 1, 2, 3, …, 150 successively for each u_i^{n+1} for n = 1, 2, 3, …, 95 using MATLAB code.

$$\begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ . \\ . \\ . \\ u_{m-1}^{n+1} \end{bmatrix} = \frac{192}{193} \begin{bmatrix} 0.0211 & 0.9895 & 0 & \cdots & 0 \\ 0.9895 & 0.0211 & 0.9895 & \ddots & \\ 0 & 0.9895 & 0.0211 & 0.9895 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots \quad 0 \\ & & & & 0.9895 \\ 0 & \cdots & 0 & 0.9895 & 0.0211 \end{bmatrix} \begin{bmatrix} u_1^{n} \\ u_2^{n} \\ u_3^{n} \\ . \\ . \\ . \\ u_{m-1}^{n} \end{bmatrix} - \frac{191}{193} \begin{bmatrix} u_1^{n-1} \\ u_2^{n-1} \\ u_3^{n-1} \\ . \\ . \\ . \\ u_{m-1}^{n-1} \end{bmatrix} \qquad (4.104)$$

4.5.4 For implicit scheme

Using (4.55) and taking i = 1, 2, 3, …, 151 successively, we get its matrix vector form AX = B as follows, where A is a tri-diagonal matrix and B is a vector.

$$\begin{bmatrix} 1.9895 & -0.4947 & 0 & \cdots & 0 \\ -0.4947 & 1.9895 & -0.4947 & \ddots & \\ 0 & -0.4947 & 1.9895 & -0.4947 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots \quad 0 \\ & & & & -0.4947 \\ 0 & \cdots & 0 & -0.4947 & 1.9895 \end{bmatrix} \begin{bmatrix} u_1^{1} \\ u_2^{1} \\ u_3^{1} \\ . \\ . \\ . \\ u_{150}^{1} \end{bmatrix} = \begin{bmatrix} 0.0067 \\ 0.0133 \\ 0.0200 \\ . \\ . \\ . \\ 1.0000 \end{bmatrix}, \qquad (4.105)$$

where $1+\lambda^2$ = 1.9895 and $\frac{-1}{2}\lambda^2$ = -0.4947 (to the nearest four decimal places), and $d_i = u_i^0 = u(x_i, 0) = f(x_i)$, for i = 1, 2, 3, …, 151. Then solving this system of equation AX = B using MATLAB code or Thomas Algorithm, we get the values of the next time-step nodal points, u_i^1 as u_1^1 = 0.0067, u_2^1 = 0.0133, u_3^1 = 0.0200, u_4^1 = 0.0267, u_5^1 = 0.0333, u_6^1 = 0.0400, u_7^1 = 0.0467, u_8^1 = 0.0533, u_9^1 = 0.0600, u_{10}^1 = 0.0667, …, u_{150}^1.

u_1^1	u_2^1	u_3^1	u_4^1	u_5^1	u_6^1	u_7^1	u_8^1	u_9^1	u_{10}^1
0.0067	0.0133	0.0200	0.0267	0.0333	0.0400	0.0467	0.0533	0.0600	0.0667

Table 6: Values of the vibrating string in case 2 of implicit scheme at the 1st time step

And considering (4.60), a matrix vector form AX = B, where A is a tri-diagonal matrix and B is a vector, we can solve u_i^{n+1} taking i = 1, 2, 3, …, 151 successively for n = 1, 2, 3, …, 95 using MATLAB code or Thomas Algorithm.

$$\begin{bmatrix} 1.4921 & -0.4947 & 0 & \cdots & & 0 \\ -0.4947 & 1.4921 & -0.4947 & & \ddots & \\ 0 & -0.4947 & 1.4921 & -0.4947 & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ & & & & & -0.4947 \\ 0 & \cdots & & 0 & -0.4947 & 1.4921 \end{bmatrix} \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ . \\ . \\ . \\ u_{m-1}^{n+1} \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ . \\ . \\ . \\ d_{m-1} \end{bmatrix}, \quad (4.106)$$

Where p + λ^2 = 1.4921, $\frac{-1}{2}\lambda^2 = -0.4947$ (to the nearest four decimal places), $d_i = u_i^n - qu_i^{n-1}$, for n = 1, 2, …, 95 and i = 1, 2, …, 151, i.e., we can solve u_i^2, u_i^3, …, u_i^{96}, for n = 1, n = 2, …, n = 95, $d_i = u_i^1 - qu_i^0$, $d_i = u_i^2 - qu_i^1$, ..., $d_i = u_i^{95} - qu_i^{94}$ respectively; taking i = 1, 2, …, 151 successively.

Case3: $\boldsymbol{\Delta t_3 = \frac{\Delta t_1}{5} = \frac{1}{160}}$ and $\boldsymbol{\Delta x_3 = \frac{\Delta x_1}{5} = \frac{\pi}{500}}$

In this case, for c = 1, k = 0.5, T = 1 and L = π, we have

$\lambda = c\frac{\Delta t_3}{\Delta x_3} = \frac{25}{8\pi}$, a = $\frac{1}{1+k\Delta t_3} = \frac{320}{321}$, b = $\frac{-1+k\Delta t_3}{1+k\Delta t_3} = \frac{-319}{321}$, p = $\frac{1+k\Delta t_3}{2} = \frac{321}{640}$, q = $\frac{1-k\Delta t_3}{2} = \frac{319}{640}$

The initial values of the wave at the nodal points, u_0^0, u_1^0, u_2^0, u_3^0, …, u_{250}^0, u_{251}^0, …, u_{500}^0 using

$$u_i^0 = u(x_i, 0) = f(x_i) = \begin{cases} \frac{2x_i}{\pi}, 0 \le x_i \le \frac{\pi}{2} \\ \frac{-2(x_i-\pi)}{\pi}, \frac{\pi}{2} \le x_i \le \pi \end{cases},$$

where $x_i = i\Delta x_3 = \frac{i\pi}{500}$, i = 0, 1, 2, …, 500, become

$$u_0^0 = u(x_0, 0) = f(x_0) = f(0) = \frac{2}{\pi}(0) = 0.0000$$

$$u_1^0 = u(x_1, 0) = f(x_1) = f(\frac{\pi}{500}) = \frac{2}{\pi}(\frac{\pi}{500}) = 0.0040$$

$$u_2^0 = u(x_2, 0) = f(x_2) = f(\frac{2\pi}{500}) = \frac{2}{\pi}(\frac{2\pi}{500}) = 0.0080$$

$$\vdots$$

$$u_{10}^0 = u(x_{10}, 0) = f(x_{10}) = f(\frac{10\pi}{500}) = \frac{2}{\pi}(\frac{10\pi}{500}) = 0.0400$$

⋮

$u_{500}^0 = u(x_{500}, 0) = f(x_{500}) = f(\frac{500\pi}{500}) = \frac{-2}{\pi}(\frac{500\pi}{500} - \pi) = 0.0000$

u_1^0	u_2^0	u_3^0	u_4^0	u_5^0	u_6^0	u_7^0	u_8^0	u_9^0	u_{10}^0
0.0040	0.0080	0.0120	0.0160	0.0200	0.0240	0.0280	0.0320	0.0360	0.0400

Table 7: Initial values of the vibrating string in case 3 at the 1st time step

N.B: Since the string is symmetrical with respect to the point $x_{250} = \frac{\pi}{2}$, it is satisfactory to find the values of $u(x_i, t_n)$ only for $0 \leq x_i \leq \frac{\pi}{2}$.

4.5.5 For explicit scheme

Using (4.46) and taking i = 1, 2, 3, …, 250 successively, we get the following matrix form to solve u_i^1.

$$\begin{bmatrix} u_1^1 \\ u_2^1 \\ u_3^1 \\ . \\ . \\ . \\ u_{250}^1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0.0211 & 0.9895 & 0 & \cdots & & 0 \\ 0.9895 & 0.0211 & 0.9895 & & \ddots & \\ 0 & 0.9895 & 0.0211 & 0.9895 & & \vdots \\ & & & & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ & & & & & 0.9895 \\ 0 & \cdots & 0 & 0.9895 & & 0.0211 \end{bmatrix}\begin{bmatrix} 0.004 \\ 0.008 \\ 0.012 \\ . \\ . \\ . \\ 1.000 \end{bmatrix}, \quad (4.107)$$

where $2(1 - \lambda^2) = 0.0211$ and $\lambda^2 = 0.9895$ (to the nearest four decimal places). Then solving u_i^1, i = 1, 2, …, 250 from (4.108) using MATLAB code, we have

$u_1^1 = 0.0040$, $u_2^1 = 0.0080$, $u_3^1 = 0.0120$, $u_4^1 = 0.0160$, $u_5^1 = 0.0200$, $u_6^1 = 0.0240$, $u_7^1 = 0.0280$, $u_8^1 = 0.0320$, $u_9^1 = 0.0360$, $u_{10}^1 = 0.0400$, …, u_{250}^1

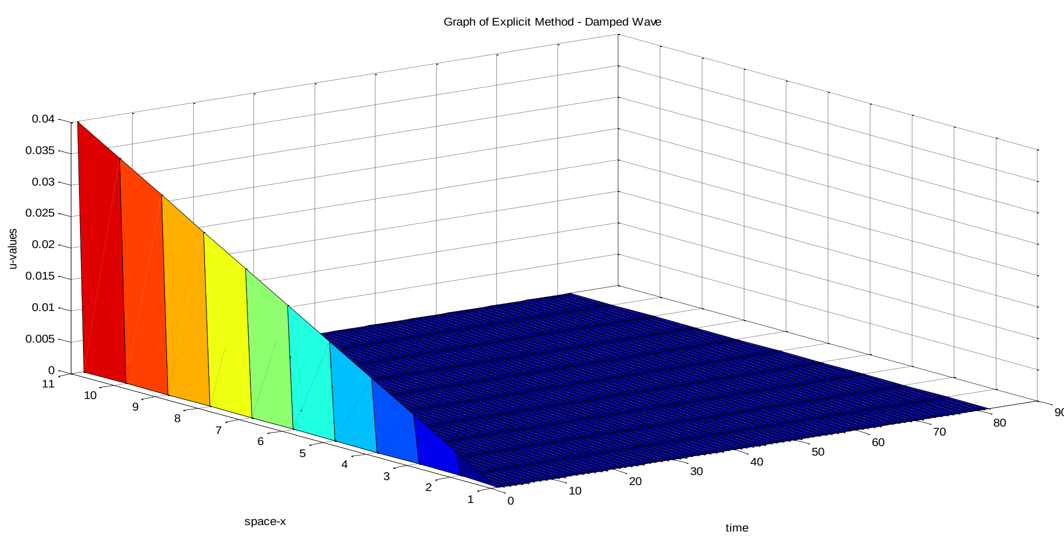
u_1^1	u_2^1	u_3^1	u_4^1	u_5^1	u_6^1	u_7^1	u_8^1	u_9^1	u_{10}^1
0.0040	0.0080	0.0120	0.0160	0.0200	0.0240	0.0280	0.0320	0.0360	0.0400

Table 8: Values of the vibrating string in case 3 of explicit scheme at the 1st time step

And we consider (4.47) to solve u_i^2, u_i^3, u_i^4, …, u_i^{160} taking i = 1, 2, 3, …, 250 successively for each u_i^{n+1} for n = 1, 2, 3, …, 159 using MATLAB code.

$$\begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ . \\ . \\ . \\ u_{m-1}^{n+1} \end{bmatrix} = \frac{192}{193} \begin{bmatrix} 0.0211 & 0.9895 & 0 & \cdots & & 0 \\ 0.9895 & 0.0211 & 0.9895 & & \ddots & \\ 0 & 0.9895 & 0.0211 & 0.9895 & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ & & & & & 0.9895 \\ 0 & \cdots & 0 & & 0.9895 & 0.0211 \end{bmatrix} \begin{bmatrix} u_1^{n} \\ u_2^{n} \\ u_3^{n} \\ . \\ . \\ . \\ u_{m-1}^{n} \end{bmatrix} - \frac{191}{193} \begin{bmatrix} u_1^{n-1} \\ u_2^{n-1} \\ u_3^{n-1} \\ . \\ . \\ . \\ u_{m-1}^{n-1} \end{bmatrix} \quad (4.108)$$

4.5.6 Graph of explicit scheme



4.5.7 For implicit scheme

Using (4.55) and taking i = 1, 2, …, 251 successively we get its matrix vector form AX = B as follows, where A is a tri-diagonal matrix and B is a vector.

$$\begin{bmatrix} 1.9895 & -0.4947 & 0 & \cdots & & 0 \\ -0.4947 & 1.9895 & -0.4947 & & \ddots & \\ 0 & -0.4947 & 1.9895 & -0.4947 & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ & & & & & -0.4947 \\ 0 & \cdots & & 0 & -0.4947 & 1.9895 \end{bmatrix} \begin{bmatrix} u_1^1 \\ u_2^1 \\ u_3^1 \\ . \\ . \\ . \\ u_{250}^1 \end{bmatrix} = \begin{bmatrix} 0.02 \\ 0.04 \\ 0.06 \\ . \\ . \\ . \\ 1.00 \end{bmatrix}, \tag{4.109}$$

where $1+\lambda^2$ = 1.9895 and $\frac{-1}{2}\lambda^2$ = -0.4947 (to the nearest four decimal places), and $d_i = u_i^0 = u(x_i, 0) = f(x_i)$, for i = 1, 2, 3, …, 251. Then solving this system of equation AX = B using MATLAB or Thomas Algorithm, we get the values of the next time-step nodal points, u_i^1 as u_1^1 = 0.0040, u_2^1 = 0.0040, u_3^1 = 0.0040, u_4^1 = 0.0040, u_5^1 = 0.0040, u_6^1 = 0.0040, u_7^1 = 0.0040, u_8^1 = 0.0040, u_9^1 = 0.0040, u_{10}^1 = 0.0040, …, u_{250}^1

u_1^1	u_2^1	u_3^1	u_4^1	u_5^1	u_6^1	u_7^1	u_8^1	u_9^1	u_{10}^1
0.0040	0.0080	0.0120	0.0160	0.0200	0.0240	0.0280	0.0320	0.0360	0.0400

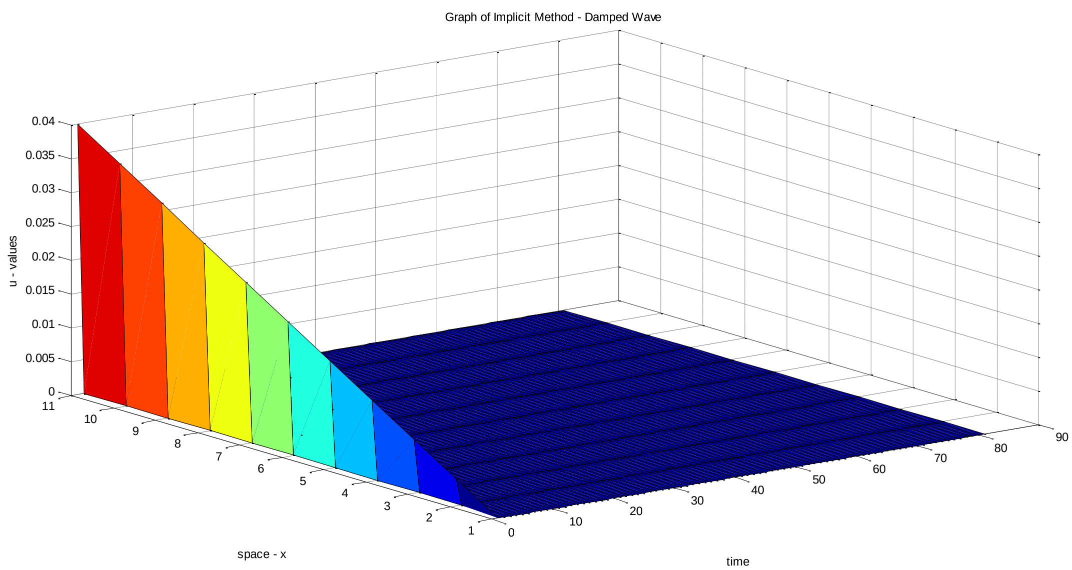
Table 9: Values of the vibrating string in case 3 of implicit scheme at the 1st time step

And considering (4.60), a matrix vector form AX = B, where A is a tri-diagonal matrix and B is a vector, we can solve u_i^{n+1} taking i = 1, 2, …, 251 successively for n = 1, 2, …, 159 using MATLAB or Thomas Algorithm.

$$\begin{bmatrix} 1.4910 & -0.4947 & 0 & \cdots & & 0 \\ -0.4947 & 1.4910 & -0.4947 & & \ddots & \\ 0 & -0.4947 & 1.4910 & -0.4947 & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ & & & & & -0.4947 \\ 0 & \cdots & & 0 & -0.4947 & 1.4910 \end{bmatrix} \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ . \\ . \\ . \\ u_{m-1}^{n+1} \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ . \\ . \\ . \\ d_{m-1} \end{bmatrix}, \tag{4.110}$$

Where p + λ^2 = 1.4910, $\frac{-1}{2}\lambda^2 = -0.4947$ (to the nearest four decimal places), $d_i = u_i^n - qu_i^{n-1}$. We can solve u_i^2, u_i^3, …, u_i^{160}, $d_i = u_i^1 - qu_i^0$, $d_i = u_i^2 - qu_i^1$, ..., $d_i = u_i^{159} - qu_i^{158}$ respectively for n = 1, 2, …, 159; taking i = 1, 2, …, 251 successively.

4.5.8 Graph of implicit scheme



4.6 Analytic (exact) solution

By (4.94) and (4.95) the exact solution of the given example in (4.96) with the initial and boundary conditions (4.97) and (4.98) respectively, with c = 1and k = 0.5 is given by

$$u(x\,,t) = \sum_{n=1}^{\infty} \sin(nx)\, e^{-t/2}\left[a_n \cos\left(\frac{\sqrt{4n^2-1}\,t}{2}\right) + b_n \sin\left(\frac{\sqrt{4n^2-1}\,t}{2}\right)\right], \quad (4.111)$$

where $a_n = \frac{8}{n^2\pi^2}\sin\left(\frac{n\pi}{2}\right)$

and $b_n = \frac{ka_n}{\sqrt{n^2-1/4}} = \frac{2ka_n}{\sqrt{n^2-1}} = \frac{8}{n^2\pi^2\sqrt{4n^2-1}}\sin\left(\frac{n\pi}{2}\right)$, for n = 1, 2, … (4.112)

$$u(x\,,t) = \sum_{n=1}^{\infty} \sin(nx)\, e^{\frac{-t}{2}}\left[\frac{8}{n^2\pi^2}\sin\left(\frac{n\pi}{2}\right)\cos\left(\frac{\sqrt{4n^2-1}\,t}{2}\right) + \frac{8}{n^2\pi^2\sqrt{4n^2-1}}\sin\left(\frac{n\pi}{2}\right)\sin\left(\frac{\sqrt{4n^2-1}\,t}{2}\right)\right]$$

$$\Rightarrow u(x\,,t) = \frac{8}{\pi^2}e^{\frac{-t}{2}}\sum_{n=1}^{\infty}\frac{1}{n^2}\sin(nx)\left[\sin\left(\frac{n\pi}{2}\right)\cos\left(\frac{\sqrt{4n^2-1}\,t}{2}\right) + \frac{1}{\sqrt{4n^2-1}}\sin\left(\frac{n\pi}{2}\right)\sin\left(\frac{\sqrt{4n^2-1}\,t}{2}\right)\right]$$

$$\Rightarrow u(x\,,t) = \frac{8}{\pi^2}e^{\frac{-t}{2}}\sin(x)\left[\cos\left(\frac{\sqrt{3}t}{2}\right) + \frac{1}{\sqrt{3}}\sin\left(\frac{\sqrt{3}t}{2}\right)\right], \text{ for n = 1.} \quad (4.113)$$

Solving the example given from its analytic solution using Mat Lab code analytically, particularly at the first time step, we get the values of the vibrations of the string (wave) at each nodal point. So, computing the solutions of (4.16) of the first time step using Mat lab code considering the above three cases, we get the following tabular results.

4.6.1 Case1: $\Delta t_1 = \frac{1}{32}$ **and** $\Delta x_1 = \frac{\pi}{100}$

u_1^1	u_2^1	u_3^1	u_4^1	u_5^1	u_6^1	u_7^1	u_8^1	u_9^1	u_{10}^1
0.0200	0.0400	0.0600	0.0800	0.1000	0.1200	0.1400	0.1600	0.1800	0.2000

Table10: Exact values of the vibrating string (wave) in case1 at the 1st time step

4.6.2 Case2: $\Delta t_2 = \frac{\Delta t_1}{3} = \frac{1}{96}$ **and** $\Delta x_2 = \frac{\Delta x_1}{3} = \frac{\pi}{300}$

u_1^1	u_2^1	u_3^1	u_4^1	u_5^1	u_6^1	u_7^1	u_8^1	u_9^1	u_{10}^1
0.0067	0.0133	0.0200	0.0267	0.0333	0.0400	0.0467	0.0530	0.0600	0.0667

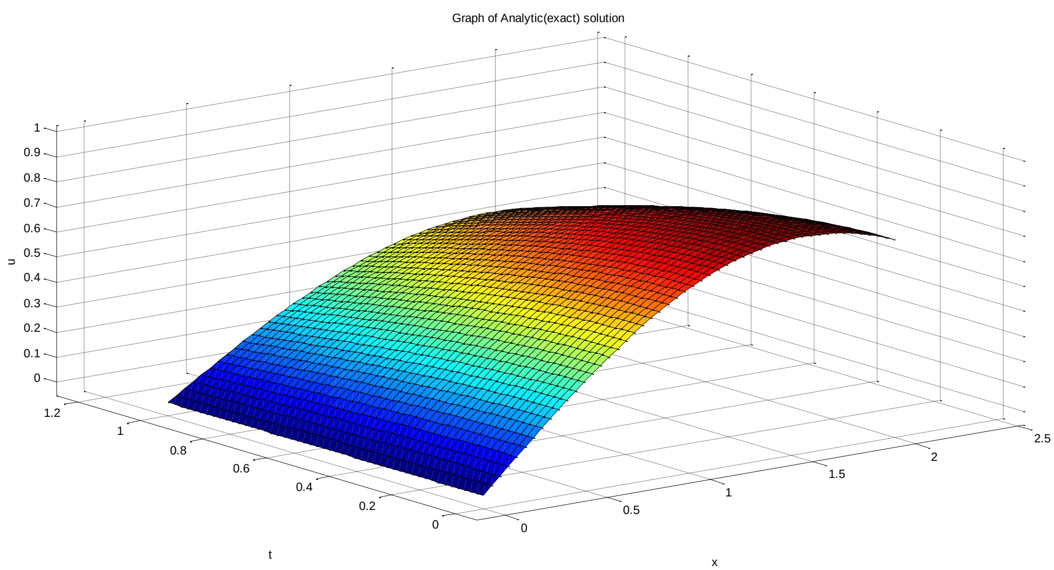
Table11: Exact values of the vibrating string (wave) in case2 at the 1st time step

4.6.3 Case3: $\Delta t_3 = \frac{\Delta t_1}{5} = \frac{1}{160}$ and $\Delta x_3 = \frac{\Delta x_1}{5} = \frac{\pi}{500}$

u_1^1	u_2^1	u_3^1	u_4^1	u_5^1	u_6^1	u_7^1	u_8^1	u_9^1	u_{10}^1
0.0040	0.0080	0.0120	0.0160	0.0200	0.0240	0.0280	0.0320	0.0360	0.0400

Table12: Exact values of the vibrating string (wave) in case3 at the 1st time step

4.6.4 Graph of analytic (exact) solution



4.7 Comparison of solutions of FDM and Analytic (Exact) of the cases

4.7.1 Comparison of solutions of FDM and Analytic for case 1

Method	Wave at some grid points/ Values of vibrating string at some nodal points									Remark
	u_1^1	u_2^1	u_3^1	u_4^1	u_5^1	u_6^1	u_7^1	u_8^1	u_9^1	u_{10}^1	
Exact	0.0200	0.0400	0.0600	0.0800	0.1000	0.1200	0.1400	0.1600	0.1800	0.2000	
Explicit	0.0200	0.0400	0.0600	0.0800	0.1000	0.1200	0.1400	0.1600	0.1800	0.2010	
Error	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0010	
Implicit	0.0200	0.0400	0.0600	0.0800	0.1000	0.1200	0.1400	0.1600	0.1800	0.2007	
Error	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0007	

Table 13: Comparison of FDM and exact solutions for case 1

4.7.2 Comparison of solutions of FDM and Analytic for case 2

Method	Wave at some grid points/ Values of vibrating string at some nodal points									Remark
	u_1^1	u_2^1	u_3^1	u_4^1	u_5^1	u_6^1	u_7^1	u_8^1	u_9^1	u_{10}^1	
Exact	0.0067	0.0133	0.0200	0.0267	0.0333	0.0400	0.0467	0.0530	0.0600	0.0667	
Explicit	0.0067	0.0134	0.0200	0.0267	0.0334	0.0400	0.0467	0.0534	0.0600	0.0667	
Error	0.0000	0.0001	0.0000	0.0000	0.0001	0.0000	0.0000	0.0004	0.0000	0.0000	
Implicit	0.0067	0.0133	0.0200	0.0267	0.0333	0.0400	0.0467	0.0533	0.0600	0.0667	
Error	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0003	0.0000	0.0000	

Table 14: Comparison of FDM and exact solutions for case 2

4.7.3 Comparison of solutions of FDM and Analytic for case 3

Method	Wave at some grid points/ Values of vibrating string at some nodal points									Remark
	u_1^1	u_2^1	u_3^1	u_4^1	u_5^1	u_6^1	u_7^1	u_8^1	u_9^1	u_{10}^1	
Exact	0.0040	0.0080	0.0120	0.0160	0.0200	0.0240	0.0280	0.0320	0.0360	0.0400	
Explicit	0.0040	0.0080	0.0120	0.0160	0.0200	0.0240	0.0280	0.0320	0.0360	0.0400	
Error	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
Implicit	0.0040	0.0080	0.0120	0.0160	0.0200	0.0240	0.0280	0.0320	0.0360	0.0400	
Error	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	

Table 15: Comparison of FDM and exact solutions for case 3

We see that the error of the solutions produced by the methods becomes reducing, when Δx and Δt replaced by $\frac{\Delta x}{3}$ and $\frac{\Delta t}{3}$ and $\frac{\Delta x}{5}$ and $\frac{\Delta t}{5}$ respectively.

$$\text{Error} = max_{i,n}|\hat{u}_i^n - u_i^n|,$$

where $\hat{u}_i^n$ is exact and u_i^n is numerical solution.

Comparison of the error of explicit and implicit methods in some selected grid points is shown in table 16. Second-order accurate (k = 0.5).

Method	Case	Δt	Δx	Error
Explicit	*i.*	$1/32$	$\pi/100$	0.0010
	ii.	$1/96$	$\pi/300$	0.0004
	iii.	$1/160$	$\pi/500$	0.0000
Implicit	*i.*	$1/32$	$\pi/100$	0.0007
	ii.	$1/96$	$\pi/300$	0.0003
	iii.	$1/160$	$\pi/500$	0.0000

Table 16: Comparison of the error of explicit and implicit methods

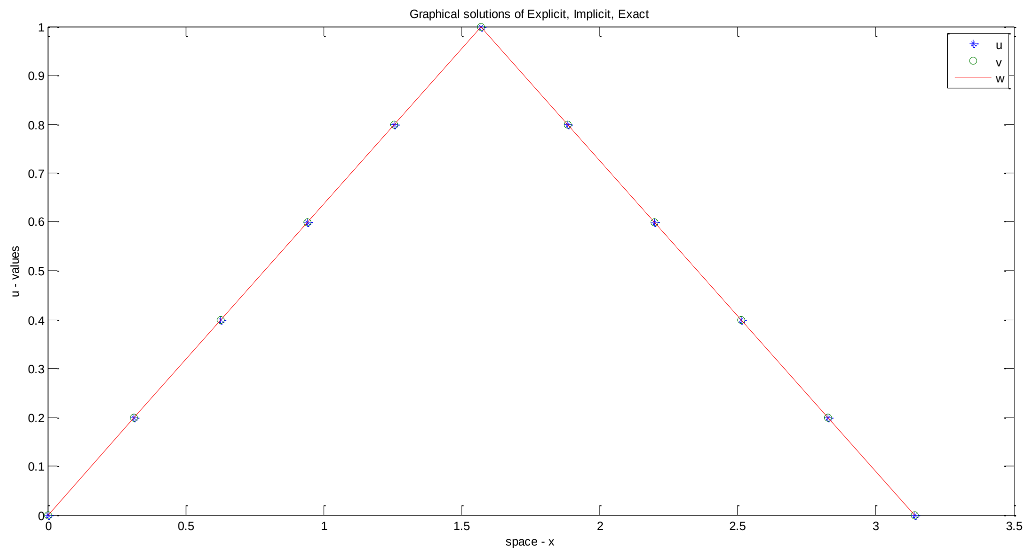


Figure5: Graphical solutions of Explicit(u), Implicit(v) & Exact(w) at $\Delta t = \frac{1}{160}$, $\Delta x = \frac{\pi}{10}$, k = 0.5

4.8 Discussion

Using the central difference approximation for the first-order and second-order time and space derivatives, we derived difference equations equivalent to the one dimensional damped wave equation up to an error of order $O\{(\Delta t)^2\ ,\ (\Delta x)^2\}$, which leads to an explicit and implicit numerical schemes.

Analyzing the stability of the schemes using VNSA, we arrived at the range of stability $0 < \lambda \leq 1$ for λ = c$\frac{\Delta t}{\Delta x}$, i.e., at the condition c $\leq \frac{\Delta x}{\Delta t}$, which is CFL-condition for explicit and implicit FDM. These show both methods are stable.

But, from matrix form u^{n+1} = Au^{n} + $bu^{n-1}$of equation (4.38), we have seen that, $\|A\|_{\infty} = \|A\|_1 = |1/2\,\lambda^2| + |1-\ \lambda^2\ | + |1/2\,\lambda^2|$.

- ✓ For 1$-\ \lambda^2 \geq 0$ implies $0 < \lambda \leq 1$, all the numbers inside the absolute values are non-negative and we get a norm of 1.
- ✓ For 1$-\ \lambda^2 < 0$ implies $\lambda > 1$, the norm is $2\lambda^2 - 1$ which is greater than 1.Thus we have conditional stability with the condition $0 < \ \lambda \ \leq\ 1$. This implies the explicit scheme conditionally stable.

And from matrix form Au^{n+1} = b of equation (4.51), we have seen that $\|A\|_{\infty} = \|A\|_1 = \left|\frac{-1}{2}\lambda^2\right| + \left|\frac{1+\mathrm{k}\Delta\mathrm{t}}{\mathbf{2}}\ +\ \lambda^2\ \right| + \left|\frac{-1}{2}\lambda^2\right|$.

- ✓ For $0 < \lambda \leq 1$ and $\lambda > 1$, the norm is $\frac{1+\mathrm{k}\Delta\mathrm{t}}{\mathbf{2}}$ + $2\lambda^2$ which is greater than 1. Thus we have unconditional stability. This shows implicit scheme is unconditionally stable.

We have seen also the consistency and convergence of the methods. The accuracy of the methods based on error analysis using test example was clearly indicated in the tables. The solutions using explicit and implicit FDMs in comparison with the exact solution are graphically presented.

Considering numerical example and applying the schemes using three different cases with damping factor k = 0.5, we arrived at table 16. From this table in case 1, we see that the absolute maximum error of explicit scheme is 0.0010 and that of implicit is 0.0007. In case 2, the absolute maximum error of explicit scheme is 0.0004 while the implicit one is 0.0003. These indicate that the implicit finite difference method is more stable, accurate, and converges faster than the explicit finite difference method. But in case 3, the absolute error of the schemes is nil. This shows as the step sizes Δt, Δx sufficiently close to zero, the approximate numerical solutions refined and converge to the exact solution, and the methods are consistent & convergent as well.

CHAPTER FIVE

5. Conclusion and Recommendation

5.1 Conclusion

Wave equation is one of the most important equations in science and engineering. In this work, we studied explicit and implicit finite-difference schemes to approximate the numerical solution of damped wave equation on finite string. The discretization schemes of damped wave equation are almost similar to that of un damped one with regard to the steps. But the complexity in the PDE due to the damped term causes algorithm complexity in the FDMs. So, solving damped wave equation using the two FDMs is more difficult than solving the un damped one. Using the centered difference (which yield more accurate approximation) and Taylor's series expansion for the first and second order space and time derivatives, we derived a difference equation of each schemes equivalent to the damped wave equation up to an error of order O $\{(\Delta x)^2, (\Delta t)^2\}$. We could also see the truncation error of each scheme.

A sufficient discussion is presented in this study to solve the damped wave equation using explicit and implicit FDMs. Stability, consistency and convergence of the methods were treated using VNSA, CFL condition, L_∞, L_1 norm of a tri-diagonal matrices of the schemes, definition and theorem. Numerical solutions of the schemes were shown considering illustrative example in each scheme taking different cases, i.e.,

i. $\Delta t_1 = \frac{1}{32}$ and $\Delta x_1 = \frac{\pi}{100}$

ii. $\Delta t_2 = \frac{\Delta t_1}{3} = \frac{1}{96}$ and $\Delta x_2 = \frac{\Delta x_1}{3} = \frac{\pi}{300}$

iii. $\Delta t_3 = \frac{\Delta t_1}{5} = \frac{1}{96}$ and $\Delta x_3 = \frac{\Delta x_1}{5} = \frac{\pi}{500}$

Finally from this study we could conclude which scheme is more efficient in solving the damped wave equation.

When we consider each schemes one by one explicit scheme is somewhat easy because the unknown can easily evaluated operating the matrices known numerically. But as the implicit scheme is expressed by tri-diagonal matrix, it can be solved using mat lab code or Thomas Algorithm.

The errors of the solutions produced by the respective methods become reducing, when Δx and Δt are decreasing. This shows that we could minimize error by increasing the grid (mesh) points of space and time step dimensions. Also even if the result obtained in the implicit method is better than the explicit method in case 1 and case 2, the solutions of the two methods are very close to each other and very close to the exact solution with minimized error. But in case 3, the results obtained by the two methods are the same to the exact solution. This may be due to the use central difference approximation, the effect of damping and using decreased uniform time step and the space size. These show almost the two methods work with good performance. The main difference between the two methods is that explicit FDM is conditionally stable satisfying the CFL condition. And the main similarity is that both methods use central difference approximation for the PDE.

5.2 Recommendation

In this study we see that how stability, consistency and convergence of each schemes occurs focusing on the nature of the solutions as Δx and Δt change their values taking constant damping coefficient, k. As we see in these cases it is clear that if both Δx and Δt approach to zero, the solution of the difference equation will converge to the exact solution; that means the errors can be minimized. So, it can be concluded that FDM is a reliable method which is capable to solve the damped wave equation.

In this work, we have not tested the effect of different values of non-negative damping coefficient, k, the effect of step size, Δx with fixed values of time step, Δt and the effect of time step, Δt with fixed values of step size, Δx on the numerical solution of damped wave equation. Thus, this thesis can be extended further testing the effects of these cases. It is suggested that the methods could also be expanded to higher dimensional damped wave equation; and the other numerical methods can be treated to solve the problem.

References

- B. Neta (2003), Numerical solution of partial differential equations MA 3243 lecture notes, department of mathematics, Naval postgraduate school, code MA/Nd, Montery, California.
- C. J. Zarowski (2004), An introduction to Numerical Analysis for Electrical and Computer Engineers.
- CF Chan Man Fong D De Kee (2003), Advanced Mathematics for Engineering and Science, Tulane University, USA.
- Chaigne (1992), "On the use of finite difference for musical synthesis, application to plucked stringed instruments," J. d'Acoust. 5(2), 181- 211.
- Christoph Kirsch (2011), Math529-Mathematical Methods for Physical Sciences II, April 27.
- D. G. Duffy (1998), "Advanced Engineering Mathematics".
- D. G. Duffy (2003), "Advanced engineering mathematics with MALAB, Chapman," Hall/CRC Press.
- Donald C. Armstead (July 2004), Modeling a vibrating string, an Honors Thesis (HONRS 499), Ball State University, Municie, Indiana.
- Erwin Kreyszing (2006), "Advanced Engineering Mathematics,'' Wiley International Edition, 9thEdition, Ohio state university, PP. 535-540.
- Gilberto E. Urroz, (2004), Convergence, Stability, and Consistency of Finite Difference Schemes in the Solution of Partial Differential Equations.
- G. Kirchhoff (1857), "On the motion of electricity in wires," Philo. Mag, vol.13, pp. 393-412.
- H. Saberi Najafi and F. Izadi (2014), Comparison of finite difference methods for solving the damped wave equation, Islamic Azad University, International Journal of Mathematical Engineering and Science.

- Hans Petter Langtangen (Dec. 12, 2013), center for Biomedical computing, Simula Research Laboratory, Department of Informatics, University of Oslo.
- J. E. Mcias-Diaz. and A. Puri (2010), “A boundedness-preserving finite-difference scheme for a damped nonlinear wave equation,’’ Applied Numerical Mathematics., vol. 60, no.1 pp.934-948.
- Jain, M.K; (2003), Numerical methods for Scientific and Engineering computation.
- Jake Wood, Modeling Numerical Solutions to Partial Differential Equations: A study in waves; Department of Physics Middlebury College, Middlebury, Vermont 05753, project report for Phys 0704.
- Jradeh M. (2008), “On the damped wave equation”, 12^{th} International conference on hyperbolic problems, Maryland, USA.
- K. S. Yee (1966), Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. IEEE Trans. Antennas Propagat., 14:302.307
- Kermit, S; (1992), Mat Lab Premier 2^{nd} edn. Prime Park Way, Natrick.
- Lutz, T. and R. Rudolf (2000), Multidimensional String Models.
- Mango, J.M et al. (2014), Journal of mathematics and statistics, on finite differences on a string problem, Department of mathematics, Makerere University, Kampala, Uganda.
- Nathaniel Page Sites (2003), M.A/M.S ‘‘Waves and Wave Motion’’ Vision-learning Vol. PHY-1 (1), supported by The National Science Foundation and the U.S Department of Education.
- N. Yener (2011), A simple solution for the damped wave equation using the Laplace transform.
- Peterson, M.R. (2004), Musical analysis and synthesis in Mat lab college Math Journal 35(5): 396 – 401.
- R. Mitchell and D. F. Grifliths (1980), “The Finite Difference Method in Partial Differential Equations,” Willey, New York.
- Rao V. Dukkipati (2010), Numerical Methods.
- Ravi P. Aarwal Donal O’Regan (2009), Ordinary and Partial Differential Equations; with special functions, Fourier series and Boundary value problems.

- Richard L. Burden & J. Douglas Fairs (2010), Youngstown state university, Numerical Analysis, Ninth Edition.
- Ron, K., (2002), General wave motion-school for champions.
- S. Abarbanel and D. Gottlieb (1976), A note on the leap-frog scheme in two and three dimensions, J. Comput. Phys., 21:351.355.
- S.S. Sastry (2006), "Introduction to Methods of Numerical Analysis, "Fourth Edition, P.333, New Delhi.
- Shipra Agarwal Chemical Engineering IIT Bombay "Fundamentals of finite difference methods", 9th Indo-German Winter Academy, PP. 9 – 11.
- Siwiak, Kazimierz, Debra McKeon (2004), Ultra-Wideband radio technology, USA: John Wiley and Sons, p.15.
- Steven C. Chapra (2010), Numerical Methods for Engineers, 6th Edition.
- X.Lu (2010), "the applications of micro local analysis in σ- evolution equation," Ph. D Thesis.

Appendices for codes

Appendix(A): Mat Lab code for Analytic solution: Case1

```
%Analytic(Exact) solution – Wave

>> x=0:0.031415926:pi/10;

>> t=0:0.03125:0.5;

>> m=length(x);

>> n=length(t);

>> %[x,t]=meshgrid(x,t);

>> %v=(8/pi.^2)*exp(-0.5*t)*sin(x)*(cos(sqrt(3)/2*t)+(1/sqrt(3))*sin(sqrt(3)/2*t));

>> %BCS

>> u(1,:)=0;

>> u(m,:)=0;

>> l=0.03125;h=0.031415926;%Mesh sizes

>> %ICS

>> u(:,1)=(2/pi)*x,0<=x<=pi/2;u_t(:,1)=0
```

Appendix(B): Mat Lab code for Analytic solution: Case2

```
>> %Analytic(Exact) solution - Wave

>> x=0:0.010471975:pi/30;

>> t=0:0.010416666:0.5;

>> m=length(x);
```

```
>> n=length(t);

>> %[x,t]=meshgrid(x,t);

>> %v=(8/pi.^2)*exp(-0.5*t)*sin(x)*(cos(sqrt(3)/2*t)+(1/sqrt(3))*sin(sqrt(3)/2*t));

>> %BCS

>> u(1,:)=0;

>> u(m,:)=0;

>> l=0.010416666;h=0.010471975;%Mesh sizes

>> %ICS

>> u(:,1)=(2/pi)*x,0<=x<=pi/2;u_t(:,1)=0
```

Appendix(C): Mat Lab code for Analytic solution: Case3

```
>> %Analytic(Exact) solution - Wave

>> x=0:0.006283185:pi/50;

>> t=0:0.00625:0.5;

>> m=length(x);

>> n=length(t);

>> %[x,t]=meshgrid(x,t);

>> %v=(8/pi.^2)*exp(-0.5*t)*sin(x)*(cos(sqrt(3)/2*t)+(1/sqrt(3))*sin(sqrt(3)/2*t));

>> %BCS

>> u(1,:)=0;

>> u(m,:)=0;

>> l=0.00625;h=0.006283185;%Mesh sizes

>> %ICS
```

```
>> u(:,1)=(2/pi)*x,0<=x<=pi/2;u_t(:,1)=0
```

Appendix (D): Mat Lab code for Explicit Scheme – Wave using Case3

```
>> %Explicit Method - Wave

>> x=0:0.006283185:pi/50;

>> t=0:0.00625:0.5;

>> m=length(x);

>> n=length(t);

>> %[x,t]=meshgrid(x,t);

>> %v=(8/pi.^2)*exp(-0.5*t)*sin(x)*(cos(sqrt(3)/2*t)+(1/sqrt(3))*sin(sqrt(3)/2*t));

>> %BCS

>> u(1,:)=0;

>> l=0.00625;h=0.006283185;%Mesh sizes

>> %ICS

>> u(:,1)=(2/pi)*x,0<=x<=pi/2;u_t(:,1)=0;

>> %Exact solution

>> %v=(8/pi.^2)*exp(-0.5*t)*sin(x)*(cos(sqrt(3)/2*t)+(1/sqrt(3))*sin(sqrt(3)/2*t));

>> %Damping factor

>> k=0.5;

>> for j=2:n;

for i=2:m-1;

a=1/(1+k*j);b=(-1+k*j)/(1+k*j);d=j/h;

u(i,j+1)=a*((d.^2)*u(i-1,j)+2*(1-d.^2)*u(i,j)+(d.^2)*u(i+1,j))+b*u(i,j-1);
```

```
end

end

>> %v=(8/pi.^2)*exp(-0.5*t)*sin(x)*(cos(sqrt(3)/2*t)+(1/sqrt(3))*sin(sqrt(3)/2*t));

>> surf(u)

>> title('Graph of Explicit Method - Wave'),xlabel('time'),ylabel('space-x'),zlabel('u-values')
```

Appendix (E): Mat Lab code for Implicit Scheme – Wave using Case3

```
>> %Implicit Method - Wave

>> x=0:0.006283185:pi/50;

>> t=0:0.00625:0.5;

>> m=length(x);

>> n=length(t);

>> %[x,t]=meshgrid(x,t);

>> %v=(8/pi.^2)*exp(-0.5*t)*sin(x)*(cos(sqrt(3)/2*t)+(1/sqrt(3))*sin(sqrt(3)/2*t));

>> %BCS

>> u(1,:)=0;

>> l=0.00625;h=0.006283185;%Mesh sizes

>> %ICS

>> u(:,1)=(2/pi)*x,0<=x<=pi/2;u_t(:,1)=0;

>> %Exact solution

>> %v=(8/pi.^2)*exp(-0.5*t)*sin(x)*(cos(sqrt(3)/2*t)+(1/sqrt(3))*sin(sqrt(3)/2*t));

>> %Damping factor

>> k=0.5;
```

```
>> for j=2:n-1;

for i=2:m-1;

p=(1+k*j)/2;q=(1-k*j)/2;d=j/h;

u(i,j)=(-0.5)*(d.^2)*u(i-1,j+1)+(p+d.^2)*u(i,j+1)+(-0.5)*(d.^2)*u(i+1,j+1)+q*u(i,j-1);

end

end

>> %v=(8/pi.^2)*exp(-0.5*t)*sin(x)*(cos(sqrt(3)/2*t)+(1/sqrt(3))*sin(sqrt(3)/2*t));

>> surf(u)

>> title('Graph of Implicit Method - Damped Wave'),xlabel('time'),ylabel('space - x'),zlabel('u - values')
```

Appendix (F): Mat Lab code for Analytic(exact) solution

```
>> syms x t u

>> u=(8/pi.^2)*exp(-0.5*t)*sin(x)*(cos(sqrt(3)/2*t)+(1/sqrt(3))*sin(sqrt(3)/2*t));

>> ezsurf(x,t,u,[0 1 0 2])

>> title('Graph of Analytic(exact) solution'),xlabel('x'),ylabel('t'),zlabel('u')
```