ERROR ANALYSIS FOR FINITE DIFFERENCE SCHEMES

Bokang Kotoane

201501277

supervised by: Mr Nchejane J. Ngaka

National University Of Lesotho

A project submitted in partial fulfilment of the requirements for the degree of Bachelor of Science Honours in Applied Mathematics.

Department of Mathematics and Computer Science

Faculty of Science and Technology
National University of Lesotho

Declaration

I, Bokang Kotoane, student number 201501277, declare that the project entitled, Error analysis for finite difference schemes for a degree of Bachelor of science Honours in Applied Mathematics at National University of Lesotho has not been previously submitted by me at this or any other University. Further, I declare that this is my original work and work done by others has been acknowledged in accordance.

Signed: B.C. Kotoane

Date: 23 July 2024

Abstract

Numerical methods are widely used in modern computations to approximate solutions to differential equations. There are many types of numerical methods that can be considered, but in this project, finite difference methods are considered. The underlying concept required for development of numerical schemes is taken into consideration. Numerical schemes are developed and error analysis is always carried out for each scheme and solutions are developed using those schemes. in finding numerical solutions $\theta = 0$, $\theta = 1$, and $\theta = \frac{1}{2}$ are considered and they represent different schemes. It is found that the Crank-Nicholson scheme is the best performing finite difference scheme.

Dedication

This project is dedicated to my fellow friend, Nchakha T Rateele. This dedication is for him due to his language when discussing Mathematical concepts. He always gave a clear picture of how numerical methods are translated and made it easy for me to interpret and make sense out of them. He really made me enjoy and even more importantly made me feel quite interested in the topic itself.

Acknowledgements

I would like to express my deepest appreciation to my supervisor Mr Nchejane J Ngaka for everything he guided me with. Most succinctly, I would like to thank him for not spoon feeding me and letting me find things on my own; that's the most enjoyable and interesting part you played in my life. I would like to thank the national University of Lesotho more specifically the department of Mathematics and Computer science for this golden opportunity to further my studies. Prof. M. Molati has been with us throughout this journey helping us with almost everything, I am very proud of your presence in the department. Finally, Mrs Ellen Mabatho Thakedi, you are highly acknowledged for your dedication in assisting, empowering and supporting us throughout the entire journey. Thank you very much.

Contents

Chapter 1

Background to the Study

1.1 Project Outline

The project is organized as follows: background to the study is covered in chapter 1 where we commence by looking at a brief introduction to finite difference methods. The project's aim which is broken down into objectives, the scope and limitations of the study are also outlined in Chapter 1. A concise literature review on derivations of the finite difference methods considered together with error analysis are all provided in Chapter 2. Chapter 3 outlines the results of the study. In addition, it covers the graphs representing the solutions and the errors for each scheme. This chapter also covers the conclusion.

1.2 Introduction

Finite difference methods (FDM) are numerical techniques used for solving differential equations by approximating the derivatives with finite differences [13][17][6]. Let's break down what this means. First of all, a differential equation describes how a function changes in relation to its independent variables. They appear in various scientific and engineering fields, such as Physics, Chemistry, Biology and many others [23] [27] [25] [35][14] [2]. These equations include derivatives (like first order or second order) of an unknown function. In finite difference methods, we discretize both the spatial domain (if applicable) and the time domain (if the problem is time dependent). The spatial domain is broken into a finite number of intervals (grid points), and the values of the solution at these intervals are approximated. Instead of directly solving the differential equation, finite difference schemes convert it into a system of linear or non-linear equations that can be solved using matrix algebra techniques. As mentioned earlier, here we approximate the derivatives. Thus, it is clear that our approximations are never 100% efficient. There are errors associated with our approximation and these errors are our main concern [30] [40]. How accurate is each scheme in approximating the exact solution? To address this matter, we will be developing or deriving these finite difference schemes and then concentrate on the error analysis for each scheme. The function which we would be working on must be differentiable [28].

This project focuses on error analysis of numerical methods, specifically, some finite difference schemes. It begins with the development of the basic finite difference schemes which are less accurate to discretize the PDEs and advances to more accurate ones which will be developed later. In this project, we will take a simple PDE which also has an analytic solution. We then apply different finite schemes to solve the PDE numerically. In the process, we will also analyse errors under L_2 norm contributed on each scheme towards the solution of the PDE.

1.3 Aim

To analyse errors for specific numerical schemes.

1.3.1 Objectives

At the end of this report, we must;

- understand the underlying concepts required for development of numerical schemes,
- develop numerical schemes and perform error analysis for each numerical scheme,
- find numerical solutions for a given problem using those developed numerical schemes,
- compare the numerical solutions with the exact solution,
- investigate the error of numerical schemes subject to L_2 norm.

1.4 Scope and Limitations

The scope of this research only included investigating the errors of the theta scheme, mainly focusing on three values of theta, $\theta = 0$, $\theta = \frac{1}{2}$ and $\theta = 1$. This study is only limited to those three cases as finite difference schemes is a very broad field of study. Moreover, here we are not interested in solving the PDE analytically, rather we are just concentrating on comparing the error contributed by different theta schemes and finding the best scheme among those.

1.5 Finite Differences

The core idea is to approximate derivatives using finite differences. A finite difference is a mathematical expression of the form:

$$f(x+b,t) - f(x,t),$$

if we fix time and then focus on change in the spatial domain where b is an arbitrary constant that define the mesh size (to be considered in the following subsection).
If we divide a finite difference by b, we get a difference quotient:

$$\frac{f(x+b,t) - f(x,t)}{b}.$$

Finite dofference methods use these difference quotients to approximate derivatives. For a differentiable function, we derive approximations for the first and second derivatives using finite differences. In the context of finite difference methods, the term "mesh size" refers to the discretization of the spatial domain [28][31]. When solving partial differential equations(PDEs) using FDM, we divide the continuous spatial domain into a grid of discrete points or nodes [28][20] . Each node represent a location in space where we approximate the solution.

1.5.1 Mesh Size

The mesh size (often denoted as h) determines the spacing between adjacent nodes in the grid. It represents the distance between neighbouring points along each dimension (e.g, x, y, z) of the spatial domain provided a uniform mesh is used. Otherwise, different mesh sizes ought to be found for each dimension. Smaller mesh size results in denser grids with more nodes, allowing higher accuracy but potentially increasing computational cost[28][29]. In contrast, larger mesh sizes lead to coarser grids, reducing accuracy but improving computation efficiency [28][5].

1.5.2 Choice of Mesh Size

Selecting an appropriate mesh size is crucial:

- Too small: very fine meshes may lead to excessive computational overhead while resulting to accurate solution of our problem. [28].
- Too large: coarse meshes may introduce significant errors in the numerical solutions while it makes computations easier [5].

Engineers and scientists often perform mesh refinement studies to find an optimal balance between accuracy and efficiency [5] [12][41]. But in this study we will focus on one mesh size and compare the errors resulting from that mesh since we are comparing different schemes instead of focusing on the mesh for a given problem. Let us succinctly emphasize that in this study we will focus on one mesh size and compare the errors resulting from that mesh since the focus is on different schemes instead of best mesh for given problem. Earlier we defined h as the step size. Now let us make it clear that the step size can be chosen depending on the parameter we are trying to discretise i.e, time or space in our context. It does not have to be the same for time and space for a particular problem. It is defined in spatial and time data as Δx and Δt, respectively, for a function of two variables (x and t)[15] [10]. How do we make approximations of derivatives of functions without taking the limit as Δx or Δt goes to zero? This is where the idea of discretization comes into picture[11][22].

So let us focus on some function $f(x,t)$. We will recall that we can essentially approximate the derivative of this function at some point $(x, f(x,t))$ at fixed t by the slope of the line connecting $(x, f(x,t))$ and $(x + \Delta x, f(x + \Delta x, t))$ [36]. The line connecting these two points is going to be used to approximate the slope of the tangent line at x which is $\frac{\partial f(x,t)}{\partial x}|_{tfixed}$ [8].
So given a function $f(x,t)$, the derivative of f with respect to x is given by:

$$\frac{\partial f(x,t)}{\partial x}|_{tfixed} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x, t) - f(x,t)}{\Delta x}.$$

Essentially, what we are doing is, we are taking the limit as the value of the function at $x + \Delta x$ becomes infinitesimally close to the the value of the function evaluated at x. But numerically when we want to do an approximation on data, it is often a tuned approximation not to take the limit as $\Delta x \to 0$. An approximation of this derivative is required therefore, and it is given by:

$$\frac{\mathrm{d}f(x,t)}{\mathrm{d}x}|_{tfixed} \approx \frac{f(x + \Delta x, t) - f(x,t)}{\Delta x}.$$

So, we can approximate the derivatives of a function $\frac{\partial f}{\partial x}|_{tfixed}$ by taking a simple finite difference over a finite Δx. This is the basis of numerical differentiation.

Recall we said we approximate the derivatives of a function using a certain scheme. This should tell us that we have an error associated with that approximation [1]. The way we are going to identify this error is by looking at the Taylor series expansion of $f(x + \Delta x, t)$ about a base point (x, t). The Taylor series expansion gives the approximation to the derivative together with the truncation error [19][34].

The theta scheme has been extensively studied and applied in various fields, including physics, engineering, and finance[4][13][37]. A significant body of research has focused on optimizing the theta scheme for different types of PDEs, such as wave equations [9] and fractional differential equations[18]. Studies have introduced high-order theta schemes, which aim to improve the accuracy without compromising the stability properties[9] . Recent advancements have combined the theta scheme with other numerical methods to enhance its performanc [24]. For instance, the Legendre spectral method has been used in conjunction with the theta scheme to solve fractional Klein–Gordon equations, demonstrating improved convergence rates and computational efficiency [18].

Furthermore, reviews of finite-difference schemes for the 1D heat/diffusion equation have compared the theta scheme with other finite-approximation schemes, providing insights into their accuracy and stability [7]. These reviews are crucial for understanding the practical applications and limitations of the theta scheme in solving real-world problems [21].

Chapter 2

Methods

Here we present numerical schemes which are the main part of this project. It is always important to know and understand the tools used before they can be put into use. For that matter, numerical schemes are derived to have a full picture of how they came about. For each scheme, error analysis shall also be taken into consideration after its derivation. It is crucial to understand the concept of Taylor series expansion before attempting this chapter [39].

2.0.1 Types of Finite Difference Schemes to Focus on

In this section we will develop and analyse the error for the following finite difference schemes.

- Forward difference
- Backward difference
- Central difference
- Second order difference
- Theta scheme
- Crank-Nicholson

2.1 Derivation of Numerical Schemes

If we want to simulate partial differential equation of the form;

$$\frac{\partial U(x,t)}{\partial x} = f(U(x,t)).$$

Then we can approximate $\frac{\partial U(x,t)}{\partial x}|_{tfixed}$ using this formula:

$$\frac{\partial U(x,t)}{\partial x} \approx \frac{U(x+\Delta x,t) - U(x,t)}{\Delta x} \approx f(U(x,t)),$$

or, we can have:

$$U(x+\Delta x,t) \approx U(x,t) + \Delta x f(U(x,t)).$$

This is actually an iteration scheme which can be implemented up in a computer in Python, in Matlab or any other programming language of preference. We notice to get the the value of the function U at the next step $x+\Delta x$ we require its value at x. The basic idea behind it is that if we have the state of the system $U(x,t)$ and we can evaluate $f(U(x,t))$ then everything on the right will be known at the current time t and the current position x and we can approximate U in the next iteration $U(x+\Delta x,t)$. So, if we have this information, we can plug that in for U(x,t) and $f(U(x,t))$ and we continue to approximate for $U(x+\Delta x,t)$ and the system can step forward to the next iteration. This is a numerical integrator. So we can integrate our differential equation using this kind of finite difference approximation to the derivative. This is one of the schemes we shall be considering known as the forward difference.

2.1.1 Forward Difference Scheme

Let us consider the Taylor series expansion of $f(x+\Delta x,t)$

$$f(x+\Delta x,t) = f(x,t) + \Delta x\frac{\partial f(x,t)}{\partial x} + \frac{(\Delta x)^2}{2!}\frac{\partial^2 f(x,t)}{\partial x^2} + \frac{(\Delta x)^3}{3!}\frac{\partial^3 f(x,t)}{\partial x^2} + \dots \tag{2.1}$$

subtracting $f(x,t)$ both sides we get

$$f(x+\Delta t,t) - f(x,t) = \frac{\partial f(x,t)}{\partial x}\Delta x + \frac{(\Delta x)^2}{2!}\frac{\partial^2 f(x,t)}{\partial x^2} + \frac{(\Delta x)^3}{3!}\frac{\partial^3 f(x,t)}{\partial x^3} + \dots$$

. Dividing both sides by Δx we get

$$\frac{f(x+\Delta x,t) - f(x,t)}{\Delta x} = \frac{\partial f(x,t)}{\partial x} + \frac{(\Delta x)}{2!}\frac{\partial^2 f}{\partial x^2} + \frac{(\Delta x)^2}{3!}\frac{\partial^3 f}{\partial x^3} + \dots$$

The step size (Δx in this case) is very small so the terms containing it including everything that follows it can be neglected to get;

$$\frac{\partial f(x,t)}{\partial x} \approx \frac{f(x+\Delta x,t) - f(x,t)}{\Delta x}. \tag{2.2}$$

Since we stated very clearly that we neglected some term in order to approximate the derivative in equation (2.2), let us do the analysis for the error associated with our approximation.

ERROR ANALYSIS

We will do the analysis by substituting (2.1) into (2.2) to get;

$$\frac{\partial f(x,t)}{\partial x} = \frac{f(x,t) + \Delta x \frac{\partial f(x,t)}{\partial x} + \frac{(\Delta x)^2}{2!}\frac{\partial^2 f(x,t)}{\partial x^2} + \frac{(\Delta x)^3}{3!}\frac{\partial^3 f(x,t)}{\partial x^3} + \ldots - f(x,t)}{\Delta x},$$

$f(x,t)$ cancel in the numerator and we have;

$$\frac{\partial f(x,t)}{\partial x} = \frac{\frac{\partial f(x,t)}{\partial x}\Delta x + \frac{(\Delta x)^2}{2!}\frac{\partial^2 f(x,t)}{\partial x^2} + \frac{(\Delta x)^3}{3!}\frac{\partial^3 f(x,t)}{\partial x^3} + \ldots}{\Delta x}.$$

Divide throughout by Δx we get;

$$\frac{\partial f(x,t)}{\partial x} = \frac{\partial f(x,t)}{\partial x} + \frac{\Delta x}{2!}\frac{\partial^2 f(x,t)}{\partial x^2} + \frac{(\Delta x)^2}{3!}\frac{\partial^3 f(x,t)}{\partial x^3} + \ldots. \tag{2.3}$$

We notice that the right hand side and the left hand side both have the $\frac{\partial f(x,t)}{\partial x}$ term . This means that on the right hand side the rest of the terms are error terms. We say that this error is of the first order of Δx because this error is controlled by Δx. We now get;

$$\frac{\partial f(x,t)}{\partial x} = \frac{f(x+\Delta x,t) - f(x,t)}{\Delta x} + \frac{\Delta x}{2!}\frac{\partial^2 f(x,t)}{\partial x^2} + \frac{(\Delta x)^2}{3!}\frac{\partial^3 f(x,t)}{\partial x^3} + \ldots.$$

The error term is denoted as $O(\Delta x)$ so that we have [28]

$$\frac{\partial f(x,t)}{\partial x} = \frac{f(x+\Delta,t) - f(x,t)}{\Delta x} + O(\Delta x). \tag{2.4}$$

If we make Δx smaller, the first error term is the one which dominates, so that means for smaller Δx, $(\Delta x)^2$ is much less than Δx so we basically say all of the higher order Δx terms are negligible and $\frac{\Delta x}{2!}\frac{\partial^2 f(x,t)}{\partial x^2}$ is the leading order error term. So, the error is of order Δx.
Briefly what this means is that we can control the error of the approximation to be smaller by making Δx smaller. The reason behind that is the error is just directly proportional to Δx. That says if we want the error to be 10 times less, then we make Δx 10 times smaller [28]. We should be aware that higher order Δx terms are negligible.

2.1.2 Backward Difference

This backward difference is exactly the same as the forward difference except now we are going to use $f(x - \Delta x, t)$.

So, we will consider the Taylor Series for $f(x-\Delta x,t)$ given by;

$$f(x-\Delta x,t)=f(x,t)-\frac{\partial f(x,t)}{\partial x}\Delta x+\frac{(\Delta x)^2}{2!}\frac{\partial^2 f(x,t)}{\partial x^2}-\frac{(\Delta x)^3}{3!}\frac{\partial^3 f(x,t)}{\partial x^3}+\dots \tag{2.5}$$

subtracting $f(x,t)$ both sides we get;

$$f(x-\Delta x,t)-f(x,t)=-\frac{\partial f(x,t)}{\partial x}\Delta x+\frac{(\Delta x)^2}{2!}\frac{\partial^2 f(x,t)}{\partial x^2}-\frac{(\Delta x)^3}{3!}\frac{\partial^3 f(x,t)}{\partial x^3}+\dots$$

multiplying by -1 both sides we get;

$$f(x,t)-f(x-\Delta x,t)=\Delta x\frac{\partial f(x,t)}{\partial x}-\frac{(\Delta x)^2}{2!}\frac{\partial^2 f(x,t)}{\partial x^2}+\frac{(\Delta x)^3}{3!}\frac{\partial^3 f(x,t)}{\partial x^3}+\dots$$

dividing both sides by Δx we get;

$$\frac{f(x,t)-f(x-\Delta x,t)}{\Delta x}=\frac{\partial f(x,t)}{\partial x}-\frac{\Delta x}{2!}\frac{\partial^2 f(x,t)}{\partial x^2}+\frac{(\Delta x)^2}{3!}\frac{\partial^3 f(x,t)}{\partial x^3}+\dots$$

rearranging the above equation we get;

$$\frac{\partial f(x,t)}{\partial x}=\frac{f(x,t)-f(x-\Delta x,t)}{\Delta x}+\frac{\Delta x}{2!}\frac{\partial^2 f(x,t)}{\partial x^2}-\frac{(\Delta x)^2}{3!}\frac{\partial^3 f(x,t)}{\partial x^3}+\dots.$$

For the same reason as stated earlier we have the approximation to the derivative as;

$$\frac{\partial f(x,t)}{\partial x}\approx\frac{f(x,t)-f(x-\Delta x,t)}{\Delta x}. \tag{2.6}$$

ERROR ANALYSIS

We can now discuss the error for this by first substituting (2.5) into (2.6) to get;

$$\frac{\partial f(x,t)}{\partial x}=\frac{f(x,t)-(f(x,t)+\Delta x\frac{\partial f(x,t)}{\partial x}+\frac{(\Delta x)^2}{2!}\frac{\partial^2 f(x,t)}{\partial x^2}+\frac{(\Delta x)^3}{3!}\frac{\partial^3 f(x,t)}{\partial x^3}+\dots)}{\Delta x}$$

$$\frac{\partial f(x,t)}{\partial x}=\frac{f(x,t)-f(x,t)+\Delta x\frac{\partial f(x,t)}{\partial x}-\frac{(\Delta x)^2}{2!}+\frac{(\Delta x)^3}{3!}+\dots}{\Delta x}$$

$$\frac{\partial f(x,t)}{\partial x}=\frac{\Delta x\frac{\partial f(x,t)}{\partial x}-\frac{(\Delta x)^2}{2!}\frac{\partial^2 f(x,t)}{\partial x^2}+\frac{(\Delta x)^3}{3!}\frac{\partial^3 f(x,t)}{\partial x^3}+\dots}{\Delta x}$$

$$\frac{\partial f(x,t)}{\partial x}=\frac{\partial f(x,t)}{\partial x}-\frac{\Delta x}{2!}\frac{\partial^2 f(x,t)}{\partial x^2}+\frac{(\Delta x)^2}{3!}\frac{\partial^3 f(x,t)}{\partial x^3}+\dots.$$

Likewise, we can also see that;

$$\frac{\partial f(x,t)}{\partial x}=\frac{f(x,t)-f(x-\Delta x,t)}{\Delta x}-\frac{\Delta x}{2!}\frac{\partial^2 f(x,t)}{\partial x^2}+\frac{(\Delta x)^2}{3!}\frac{\partial^3 f(x,t)}{\partial x^3}+\dots.$$

And the error is of order Δx, so we have;

$$\frac{\partial f(x,t)}{\partial x}|_{tfixed}=\frac{f(x,t)-f(x-\Delta x,t)}{\Delta x}+O(\Delta x). \tag{2.7}$$

2.1.3 Central Difference Scheme

We noticed that the error in the previous schemes comes from the third term of the Taylor series. Now we develop a scheme that performs better. Let's constructively build the central difference. So let us subtract equation (2.5) from equation (2.1) to get;

$$f(x+\Delta x,t) - f(x-\Delta x,t) = 2\Delta x\frac{\partial f(x,t)}{\partial x} + \frac{2(\Delta x)^3}{3!}\frac{\partial^3 f(x,t)}{\partial x^3} + \ldots.$$

Divide both sides by $2\Delta x$ to get;

$$\frac{f(x+\Delta x,t) - f(x-\Delta x,t)}{2\Delta x} = \frac{\partial f(x,t)}{\partial x} + \frac{(\Delta x)^2}{3!}\frac{\partial^3 f(x,t)}{\partial x^3} + \ldots.$$

Rearranging we get;

$$\frac{\partial f(x,t)}{\partial x} = \frac{f(x+\Delta x,t) - f(x-\Delta x,t)}{2\Delta x} - \frac{(\Delta x)^2}{3!}\frac{\partial^3 f(x,t)}{\partial x^3} + \ldots.$$

We now see that the approximation for the derivative is given by;

$$\frac{\partial f(x,t)}{\partial x} \approx \frac{f(x+\Delta x,t) - f(x-\Delta x,t)}{2\Delta x}. \tag{2.8}$$

ERROR ANALYSIS

Using (2.1) and (2.8), we have that;

$$\begin{aligned}\frac{\partial f(x,t)}{\partial x} = &\frac{f(x,t) + \Delta x\frac{\partial f(x,t)}{\partial x} + \frac{(\Delta x)^2}{2!}\frac{\partial^2 f(x,t)}{\partial x^2} + \frac{(\Delta x)^3}{3!}\frac{\partial^3 f(x,t)}{\partial x^3} + \ldots}{2\Delta x} \\ &- \left(\frac{f(x,t) - \Delta x\frac{\partial f(x,t)}{\partial x} + \frac{(\Delta x)^2}{2!}\frac{\partial^2 f(x,t)}{\partial x^2} - \frac{(\Delta x)^3}{3!}\frac{\partial^3 f(x,t)}{\partial x^3} + \ldots}{2\Delta t}\right)\end{aligned} \tag{2.9}$$

$$\begin{aligned}\frac{\partial f(x,t)}{\partial x} = &\frac{f(x,t) + \Delta x\frac{\partial f(x,t)}{\partial x} + \frac{(\Delta x)^2}{2!}\frac{\partial^2 f(x,t)}{\partial x^2} + \frac{(\Delta x)^3}{3!}\frac{\partial^3 f(x,t)}{\partial x^3} + \ldots}{2\Delta x} \\ &+ \left(\frac{-f(x,t) + \Delta x\frac{\partial f(x,t)}{\partial x} - \frac{(\Delta x)^2}{2!}\frac{\partial^2 f(x,t)}{\partial x^2} + \frac{(\Delta x)^3}{3!}\frac{\partial^3 f(x,t)}{\partial x^3} + \ldots}{2\Delta x}\right).\end{aligned} \tag{2.10}$$

Finally we have;

$$\frac{\partial f(x,t)}{\partial x} = \frac{\partial f(x,t)}{\partial x} + \frac{\Delta x^2}{3!}\frac{\partial^3 f(x,t)}{\partial x^3} + \ldots. \tag{2.11}$$

Introducing the finite difference approximation from (2.8) we have;

$$\frac{\partial f(x,t)}{\partial x} = \frac{f(x+\Delta x,t) - f(x-\Delta x,t)}{2\Delta x} + \frac{\Delta x^2}{3!}\frac{\partial^3 f(x,t)}{\partial x^3} + \ldots.$$

From the above equation, we can notice that the error is of order $(\Delta x)^2$, we can write:

$$\frac{\partial f(x,t)}{\partial x}|_{tfixed} = \frac{f(x+\Delta x,t) - f(x-\Delta x,t)}{2\Delta x} + O(\Delta x)^2. \tag{2.12}$$

The error for the central scheme is much less than that of the forward scheme and the backward scheme. If we want the approximation to be a hundred times better, we only have to make Δx 10 times smaller. This means if we wanted the same accuracy for the above schemes, we need to collect 10 times less data for central difference than we would with Forward and Backward difference.

2.1.4 Central Difference for Second Derivatives

Adding equation (2.5) from (2.1), we get;

$$\begin{aligned} f(x+\Delta x,t)+f(x-\Delta x,t) = f(x,t) + \Delta x\frac{\partial f(x,t)}{\partial x} + \frac{(\Delta x)^2}{2!}\frac{\partial^2 f(x,t)}{\partial x^2} \\ + \frac{(\Delta x)^3}{3!}\frac{\partial^3 f(x,t)}{\partial x^2} + \dots \\ + f(x,t) - \Delta x\frac{\partial f(x,t)}{\partial x} \\ + \frac{(\Delta x)^2}{2!}\frac{\partial^2 f(x,t)}{\partial x^2} - \frac{(\Delta x)^3}{3!}\frac{\partial^3 f(x,t)}{\partial x^3} + \dots \end{aligned} \tag{2.13}$$

$$f(x+\Delta x,t)+f(x-\Delta x,t) = 2f(x,t) + 2\frac{(\Delta x)^2}{2!}\frac{\partial^2 f(x,t)}{\partial x^2} + 2\frac{(\Delta x)^4}{4!}\frac{\partial^4 f(x,t)}{\partial x^4} + \dots$$

$$f(x+\Delta x,t) - 2f(x,t) + f(x-\Delta x,t) = 2\frac{(\Delta x)^2}{2!}\frac{\partial^2 f(x,t)}{\partial x^2} + 2\frac{(\Delta x)^4}{4!}\frac{\partial^4 f(x,t)}{\partial x^4} + \dots.$$

Dividing throughout by $(\Delta x)^2$, we get;

$$\frac{f(x+\Delta x,t) - 2f(x,t) + f(x-\Delta x,t)}{(\Delta x)^2} = \frac{\partial^2 f(x,t)}{\partial x^2} + \frac{(\Delta x)^2}{12}\frac{\partial^4 f(x,t)}{\partial x^4} + \dots.$$

Hence we can notice that;

$$\frac{\partial^2 f(x,t)}{\partial x^2} = \frac{f(x+\Delta x,t) - 2f(x,t) + f(x-\Delta x,t)}{(\Delta x)^2} - \frac{(\Delta x)^2}{12}\frac{\partial^4 f(x,t)}{\partial x^4} + \dots.$$

According to our previous analysis, we notice now that this finite difference has order $(\Delta x)^2$, therefore we write;

$$\frac{\partial^2 f(x,t)}{\partial x^2}\Big|_{tfixed} = \frac{f(x+\Delta x,t) - 2f(x,t) + f(x-\Delta x,t)}{(\Delta x)^2} + O(\Delta x)^2. \tag{2.14}$$

2.2 The Heat Equation

Since the theta scheme is specific for a problem to be solved, it is intuitive to introduce it before introducing the theta scheme. The problem we are going to focus on in this report is the heat equation adapted from Jiří Lebl Oklahoma State University [16]. Heat equation is a partial differential equation that describes the distribution of heat (or temperature) in a given region over time. It is commonly written as;

$$\frac{\partial U(x,t)}{\partial t} = D\nabla^2 u \tag{2.15}$$

where,

- $U(x,t)$ is the temperature at position x and time t,
- D is the thermal conductivity of the material,
- ∇^2 is the Laplace operator.

In this project, we concentrate on a simple one dimensional problem, where we look at the temperature variation along the length of the rod with time.
The heat equation for a one dimensional rod is given by;

$$\frac{\partial U(x,t)}{\partial t} = D\frac{\partial^2 U(x,t)}{\partial x^2} \tag{2.16}$$

- Thermal diffusivity/conductivity $D = 0.003$.
- Boundary condition $U(0,t) = 0$, $U(1,t) = 0$.
- Initial condition $U(x,0) = 50x(1-x)$.

The figure below shows one dimensional rod which is insulated except at the ends but kept at temperature 0 at the boundaries adapted from Jiří Lebl Oklahoma State University [16].

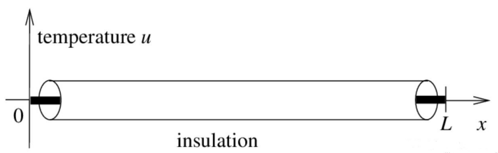

Figure 2.1: Insulated wire of length L

The theta scheme for (2.16) is introduced in the next section.

2.3 Theta Scheme

The theta scheme is one of the most important schemes in numerical methods. It gives rise to the most popular scheme known as the Crank-Nicolson scheme [23][32].

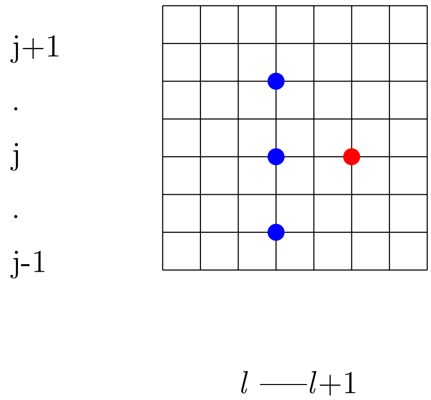

Figure 2.2: Explicit finite-difference discretisation: The red circle represents the solution U at the next time step and the blue circles represent the solution U at the current time step.

A more general form of this scheme include a parameter $\theta \in [0,1]$ and it is given by;

$$U_j^{l+1} = (1-\theta)U_j^l + \frac{D\Delta t}{(\Delta x)^2}\left[\theta\left(U_{j+1}^{l+1} - 2U_j^{l+1} + U_{j-1}^{l+1}\right)\right] + \frac{D\Delta t}{(\Delta x)^2}\left[(1-\theta)\left(U_{j+1}^{l} - 2U_j^{l} + U_{j-1}^{l}\right)\right]. \quad (2.17)$$

Theta here represents a normalized or dimensionless temperature, which can simplify the analysis and solution of the problem.
Let $\frac{D\Delta t}{(\Delta x)^2} = \alpha$, which is known as the Courant number.
So that we have;

$$U_j^{l+1} = (1-\theta)U_j^l + \alpha\left[\theta\left(U_{j+1}^{l+1} - 2U_j^{l+1} + U_{j-1}^{l+1}\right)\right] + \alpha\left[(1-\theta)\left(U_{j+1}^{l} - 2U_j^{l} + U_{j-1}^{l}\right)\right]. \quad (2.18)$$

For $\theta = 0$, (2.18) becomes ;

$$U_j^{l+1} = U_j^l + \alpha\left(U_{j+1}^l - 2U_j^l + U_{j-1}^l\right). \quad (2.19)$$

This is an explicit scheme. Upon discretizing it we have;

$$U_j^{l+1} = \alpha U_{j+1}^l + (1-2\alpha)\,U_j^l + \alpha U_{j-1}^l. \quad (2.20)$$

Equation (2.20) basically tells us that to get the solution for our equation at the next time step, we just need to know the values of that solution at the current time step at different nodes. Below is a system for the explicit method.

$$
\begin{bmatrix} U_1^{l+1} \\ \\ \vdots \\ \\ U_N^{l+1} \end{bmatrix} = \begin{bmatrix} 1-2\alpha & \alpha & 0 & \dots & 0 \\ \alpha & 1-2\alpha & \alpha & & \vdots \\ 0 & \alpha & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \alpha \\ 0 & 0 & & \alpha & 1-2\alpha \end{bmatrix} \begin{bmatrix} U_1^{l} \\ \\ \vdots \\ \\ U_N^{l} \end{bmatrix} + \alpha \begin{bmatrix} U_0^{l} \\ 0 \\ \vdots \\ 0 \\ U_{N+1}^{l} \end{bmatrix}
$$

For $\theta = 1$ we get;

$$U_j^{l+1} = U_j^l + \frac{D\Delta t}{(\Delta x)^2}\left[\left(U_{j+1}^{l+1} - 2U_j^{l+1} + U_{j-1}^{l+1}\right) + U_{j-1}^{l+1} + (1-1)\left(U_{j+1}^{l} - 2U_j^{l} + U_{j-1}^{l}\right)\right]$$

which simplify to;

$$U_j^{l+1} = U_j^l + \frac{D\Delta t}{(\Delta x)^2}\left(U_{j+1}^{l+1} - 2U_j^{l+1} + U_{j-1}^{l+1}\right).$$

This is an implicit scheme.

$$U_j^l = -\alpha U_{j+1}^{l+1} + (1+2\alpha)\, U_j^{l+1} - \alpha U_{j-1}^{l+1}.$$

As noted by Wilmott et.al, implicit finite difference scheme is also known as fully-implicit finite difference scheme. It uses the backward-difference approximation for $\frac{\partial u}{\partial t}$ term and symmetric central-difference approximation for the $\frac{\partial^2 u}{\partial x^2}$ [38][3]. Figure 2.3 gives a stencil for backward difference method.

$$U_j^{l-1} = -\alpha U_{j+1}^{l} + (1+2\alpha) U_j^{l} - \alpha U_{j-1}^{l} \tag{2.21}$$

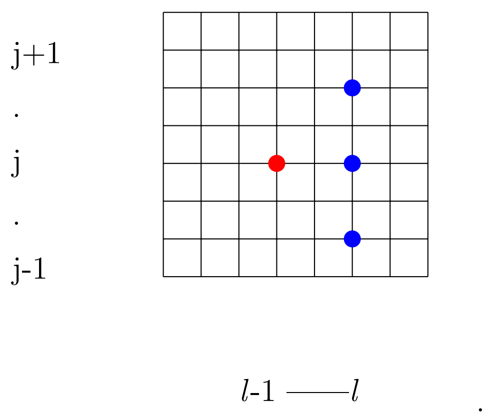

Figure 2.3: Implicit finite-difference discretisation: The red circle represents the solution U at the previous time step and the blue circles represent the solution U at the current time step.

Equations (2.21) are implicit because the solution at the previous time step depends on the solution at the current time step[26]. The matrix system that is formed is;

$$\begin{bmatrix} 1+2\alpha & -\alpha & 0 & \dots & 0 \\ -\alpha & 1+2\alpha & -\alpha & & \vdots \\ 0 & -\alpha & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & -\alpha \\ 0 & 0 & & -\alpha & 1+2\alpha \end{bmatrix} \begin{bmatrix} U_1^l \\ \vdots \\ \\ \vdots \\ U_N^l \end{bmatrix} = \begin{bmatrix} U_1^{l-1} \\ \\ \vdots \\ \\ U_N^{l-1} \end{bmatrix} + \alpha \begin{bmatrix} U_0^l \\ 0 \\ \vdots \\ 0 \\ U_{N+1}^l \end{bmatrix}$$

For $\theta = \frac{1}{2}$ (2.18) becomes;

$$-\alpha U_{j+1}^{l+1} + 2\left(1+\alpha\right) U_j^{l+1} - \alpha U_{j-1}^{l+1} = \alpha U_{j+1}^l + 2\left(1-\alpha\right) U_j^l + \alpha U_{j-1}^l \tag{2.22}$$

which is called the Crank-Nicholson scheme.

According to Wilmott et.al [38], the Crank-Nicolson finite-difference method is used to overcome the stability limitations imposed by the stability and convergence restrictions of the explicit finite-difference method, and have $O((\Delta t)^2)$ rates of convergence to the solution $f(x,t)$ the partial differential equation[1].

To be succinct, Crank-Nicholson scheme is one specific case of a more general numerical analysis method known as the theta method. If we try to find $\frac{\partial U}{\partial t}|_j^{l+\frac{1}{2}}$ using the central difference we find that;

$$\frac{\partial U}{\partial x}|_j^{l+\frac{1}{2}} \approx \frac{U_j^{l+1} - U_j^l}{\Delta x}$$

of which it is noticed that the $\frac{\partial U}{\partial t}|_j^{l+\frac{1}{2}}$ is approximated without explicitly evaluating any term at $l+\frac{l}{2}$ in the scheme. The above observation was achieved by Crank and Nicholson approximation in 1947 as shown in (2.23). We can notice then that;

$$\frac{\partial^2 U}{x^2}|_j^{l+\frac{1}{2}} \approx \frac{1}{2}\left[\frac{U_{j+1}^{l+1} - 2U_j^{l+1} + U_{j-1}^{l+1}}{(\Delta x)^2} + \frac{U_{j+1}^{l+1} - 2U_j^l + U_{j-1}^{l-1}}{(\Delta x)^2}\right]$$

Using these above approximations, the Crank-Nicolson scheme does not require any evalution of the solution at each time level $l+\frac{1}{2}$. This in turn implies that the algebraic equations resulting from the discretization of the original differential problem are coupled, making the scheme to be implicit as seen previously.

Let's set it on the grid. Time stepping is done by Euler method at the j-th spatial point.

[1] The rate of convergence of the implicit and explicit methods is O(Δt)

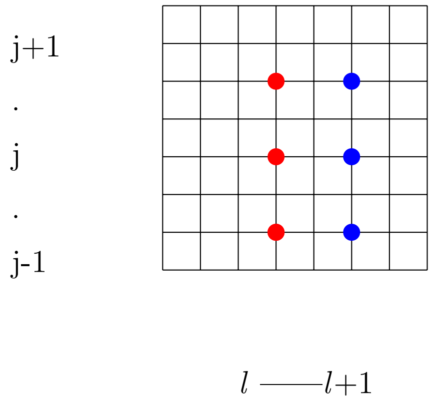

Figure 2.4: Crank-Nicholson finite-difference discretization: At each time step, the red circles are the unknowns solutions and the blue circles are the known solutions from the previous step.

The second derivative is discretized at the l time and also at the l+1 time, and the difference in time becomes second order at average time. This method can essentially be viewed as calculating U at the midpoint between the l and l+1 time step. Let's derive the Crank-Nicholson method.

The Crank-Nicholson scheme can also be derived directly following Sauer [33] as ;

$$2\frac{U_j^{l+1} - U_j^l}{\Delta t} = D\frac{U_{j+1}^l - 2U_j^l + U_{j-1}^l}{(\Delta x)^2} + D\frac{U_{j+1}^{l+1} - 2U_j^{l+1} + U_{j-1}^{l+1}}{(\Delta x)^2}$$

$$2\frac{U_j^{l+1} - U_j^l}{\Delta t} = \frac{D}{(\Delta x)^2}\left(U_{j+1}^l - 2U_j^l + U_{j-1}^l + U_{j+1}^{l+1} - 2U_j^{l+1} + U_{j-1}^{l+1}\right)$$

$$U_j^{l+1} - U_j^l = \frac{D\Delta t}{(2\Delta x)^2}\left(U_{j+1}^l - 2U_j^l + U_{j-1}^l + U_{j+1}^{l+1} - 2U_j^{l+1} + U_{j-1}^{l+1}\right)$$

But we know $\alpha = \frac{D\Delta t}{(\Delta x)^2}$ hence we get:

$$U_j^{l+1} - U_j^l = \frac{\alpha}{2}\left(U_{j+1}^l - 2U_j^l + U_{j-1}^l + U_{j+1}^{l+1} - 2U_j^{l+1} + U_{j-1}^{l+1}\right)$$

We can rearrange to get;

$$U_j^{l+1} - \frac{\alpha}{2}\left(U_{j+1}^{l+1} - 2U_j^{l+1} + U_{j-1}^{l+1}\right) = U_j^l + \frac{\alpha}{2}\left(U_{j+1}^l - 2U_j^l + U_{j-1}^l\right)$$

$$-\frac{\alpha}{2}U_{j+1}^{l+1} + U_j^{l+1} + 2\frac{\alpha}{2}U_j^{l+1} - \frac{\alpha}{2}U_{j-1}^{l+1} = \frac{\alpha}{2}U_{j+1}^l + U_j^l - 2\frac{\alpha}{2}U_j^l + \frac{\alpha}{2}U_{j-1}^l$$

or

$$-\frac{\alpha}{2}U_{j+1}^{l+1} + (1+\alpha)\,U_j^{l+1} - \frac{\alpha}{2}U_{j-1}^{l+1} = \frac{\alpha}{2}U_{j+1}^l + (1-\alpha)\,U_j^l + \frac{\alpha}{2}U_{j-1}^l$$

Multiplying throughout by 2 we get the Crank-Nicolson scheme to be;

$$-\alpha U_{j+1}^{l+1} + 2\,(1+\alpha)\,U_j^{l+1} - \alpha U_{j-1}^{l+1} = \alpha U_{j+1}^l + 2\,(1-\alpha)\,U_j^l + \alpha U_{j-1}^l \qquad (2.23)$$

Equation (2.23) is written in the form that suggests a matrix equation $\mathbf{Ax} = \mathbf{By}$. To see how, let's analyse this equation.

On the left hand side of (2.23), we target U_k^{l+1} for $k = 0, 1, 2...N$. This means that $\mathbf{x}$ contains $U_0, U_1, U_2...U_N$ at the l+1 time step and these are all unknown. $\mathbf{A}$ is the coefficient matrix depending on the α as we will see when the form of a system is constructed. The right hand side of the equation constitute another matrix $\mathbf{B}$ which also depends on α multiplied by a column vector $\mathbf{y}$. This column vector $\mathbf{y}$ contains $U_0, U_1, U_2...U_N$, now at l time step where U_0 and U_N are the boundary conditions. It is important to note that $\mathbf{y}$ is time dependent, meaning it will be updated at each and every iteration or time step. Let's assume the boundary conditions are given as $U_0^l = \beta$ and $U_N^l = \kappa$.
To commence with, it is important to write the problem as;

$$\mathbf{A}\mathbf{U}^{l+1} = \mathbf{B}\mathbf{y}^l$$

The system extracted from (2.23) to be;

$$\begin{bmatrix} 2(1+\alpha) & -\alpha & 0 & \dots & 0 \\ -\alpha & 2(1+\alpha) & -\alpha & & \vdots \\ 0 & -\alpha & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & -\alpha \\ 0 & 0 & & \alpha & 2(1+\alpha) \end{bmatrix} \begin{bmatrix} U_1^{l+1} \\ \vdots \\ U_k^{l+1} \\ \vdots \\ U_{N-1}^{l+1} \end{bmatrix} =$$

$$\begin{bmatrix} 2(1-\alpha) & \alpha & 0 & \dots & 0 \\ \alpha & 2(1-\alpha) & \alpha & & \vdots \\ 0 & \alpha & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \alpha \\ 0 & 0 & & \alpha & 2(1-\alpha) \end{bmatrix} \begin{bmatrix} \beta \\ \vdots \\ U_k^l \\ \vdots \\ \kappa \end{bmatrix} \quad (2.24)$$

for $k = 1, ..., N-1$.

The above system can be solved using any scientific tool such as Matlap, Python and Octave to mention a few. That will be handled in the problem that shall be consider in later sections on error analysis.

According to Sauer [33], Crank-Nicholson method is of order $O(\Delta x)^2$. He further emphasizes that in addition to its unconditional stability, this makes the method in general superior to the Forward and Backward difference methods in terms of convergence.

To further analyse the error of this magnificent methods, Sauer further considers four equations in developing the error focusing on the diffusion equation. The assumption that the higher derivatives for the solution U exist to satisfy the need for them is also taken into account [33].

The backward difference formula is;

$$\frac{\partial U(x,t)}{\partial t} = \frac{U(x,t) - U(x,t-\Delta t)}{\Delta t} + \frac{\Delta t}{2}\frac{\partial^2 U(x,t)}{\partial t^2} - \frac{(\Delta t)^2}{6}\frac{\partial^3 U(x,t_1)}{\partial t^3}, \tag{2.25}$$

where $t - \Delta t < t_1 < t$, assuming the partial derivatives exist. We, proceeds by expanding the Taylor series of $\frac{\partial^2 U(x,t)}{\partial t^2}$ in the variable t which is given by;

$$\frac{\partial^2 U(x,t-\Delta t)}{\partial x^2} = \frac{\partial^2 U(x,t)}{\partial x^2} - \Delta t \frac{\partial^3 U(x,t)}{\partial x^2 \partial t} - \frac{(\Delta t)^2}{2}\frac{\partial^4 U(x,t_2)}{\partial x^2 \partial t^2},$$

where $t - \Delta t < t_2 < t$.

This can also be written as;

$$\frac{\partial^2 U(x,t)}{\partial x^2} = \frac{\partial^2 U(x,t-\Delta t)}{\partial x^2} + \Delta t \frac{\partial^3 U(x,t)}{\partial x^2 \partial t} + \frac{(\Delta t)^2}{2}\frac{\partial^4 U(x,t_2)}{\partial x^2 \partial t^2}. \tag{2.26}$$

Equation (2.26) gives us the second equation in his consideration following (2.25). Moving on, the centered difference for second derivatives is given as follows;

$$\frac{\partial^2 U(x,t)}{\partial x^2} = \frac{U(x+\Delta x,t) - 2U(x,t) + U(x-\Delta x,t)}{(\Delta x)^2} + \frac{(\Delta x)^2}{12}\frac{\partial^4 U(x_1,t)}{\partial x^4}, \tag{2.27}$$

and

$$\frac{\partial^2 U(x,t-\Delta t)}{\partial x^2} = \frac{U(x+\Delta x,t-\Delta t) - 2U(x,t-\Delta t) + U(x-\Delta x,t-\Delta t)}{(\Delta x)^2} + \frac{(\Delta x)^2}{12}\frac{\partial^4 U(x_2,t-\Delta t)}{\partial x^4}, \tag{2.28}$$

where x_1 and x_2 lie between x and $x + \Delta x$.

Substituting (2.28), (2.27), (2.26)and (2.25) into the diffusion equation which is rewritten in the form;

$$\frac{\partial U}{\partial t} = D\left[\frac{1}{2}\frac{\partial^2 U(x_2,t_2)}{\partial x^2} + \frac{1}{2}\frac{\partial^2 U(x_2,t_2)}{\partial x^2}\right].$$

Strategically, the aim is to replace the left hand side with (2.25), the first half of the right hand side with (2.27) and the second part of the right side with the combination of (2.26) and (2.28). Up on doing that we get;

$$\begin{aligned}\frac{U(x,t) - U(x,t-\Delta t)}{\Delta t} + \frac{\Delta t}{2}\frac{\partial^2 U(x,t)}{\partial t^2} - \frac{(\Delta t)^2}{6}\frac{\partial^3 U(x,t_1)}{\partial t^3} = \\ \frac{D}{2}\left[\frac{U(x+\Delta x,t) - 2U(x,t) + U(x-\Delta x,t)}{(\Delta x)^2}\right] + \\ \frac{D}{2}\left[\frac{(\Delta x)^2}{12}\frac{\partial^4 U(x_1,t)}{\partial x^4}\right] + \frac{D}{2}\left[\frac{U(x+\Delta x,t-\Delta t) - 2U(x,t-\Delta t) + U(x-\Delta x,t-\Delta t)}{(\Delta x)^2}\right] + \\ \frac{D}{2}\left[\frac{(\Delta x)^2}{12}\frac{\partial^4 U(x_2,t-\Delta t)}{\partial x^4}\right] + \\ \frac{D}{2}\left[\Delta t \frac{\partial^3 U(x,t)}{\partial x^2 \partial t} + \frac{(\Delta t)^2}{2}\frac{\partial^4 U(x,t_2)}{\partial x^2 \partial t^2}\right].\end{aligned} \tag{2.29}$$

Then upon rearranging we get;

$$
\begin{aligned}
&\frac{U(x,t)-U(x,t-\Delta t)}{\Delta t} = \\
&\frac{D}{2}\left[\frac{U(x+\Delta x,t)-2U(x,t)+U(x-\Delta x,t)}{(\Delta x)^2} + \frac{U(x+\Delta x,t-\Delta t)-2U(x,t-\Delta t)+U(x-\Delta x,t-\Delta t)}{(\Delta x)^2}\right] \\
&\quad + \frac{D(\Delta x)^2}{24}\frac{\partial^4 U(x_1,t)}{\partial x^4} + \frac{D(\Delta t)^2}{2}\frac{\partial^4 U(x,t_2)}{\partial x^2 \partial t^2} + \frac{D(\Delta x)^2}{24}\frac{\partial^4 U(x_2,t-\Delta t)}{\partial x^4} + \frac{D\Delta t}{2}\frac{\partial^3 U(x,t)}{\partial x^2 \partial t} - \\
&\qquad \frac{\Delta t}{2}\frac{\partial^2 U(x,t)}{\partial t^2} + \frac{(\Delta t)^2}{6}\frac{\partial^3 U(x,t_1)}{\partial t^3}. \qquad (2.30)
\end{aligned}
$$

Hence the truncation error is;

$$
\begin{aligned}
\epsilon_{tr} = &\frac{D(\Delta x)^2}{24}\frac{\partial^4 U(x_1,t)}{\partial x^4} + \frac{D(\Delta t)^2}{2}\frac{\partial^4 U(x,t_2)}{\partial x^2 \partial t^2} + \frac{D(\Delta x)^2}{24}\frac{\partial^4 U(x_2,t-\Delta t)}{\partial x^4} + \\
&\frac{D\Delta t}{2}\frac{\partial^3 U(x,t)}{\partial x^2 \partial t} - \frac{\Delta t}{2}\frac{\partial^2 U(x,t)}{\partial t^2} + \frac{(\Delta t)^2}{6}\frac{\partial^3 U(x,t_1)}{\partial t^3}, \qquad (2.31)
\end{aligned}
$$

rearranging the terms we have;

$$
\begin{aligned}
\epsilon_{tr} = &-\frac{\Delta t}{2}\frac{\partial^2 U(x,t)}{\partial t^2} + \frac{(\Delta t)^2}{6}\frac{\partial^3 U(x,t_1)}{\partial t^3} + \frac{D(\Delta x)^2}{24}\left[\frac{\partial^4 U(x_1,t)}{\partial x^4} + \frac{\partial^4 U(x_2,t-\Delta t)}{\partial x^4}\right] + \\
&\frac{D(\Delta t)^2}{2}\frac{\partial^4 U(x,t_2)}{\partial x^2 \partial t^2} + \frac{D\Delta t}{2}\frac{\partial^3 U(x,t)}{\partial x^2 \partial t}. \qquad (2.32)
\end{aligned}
$$

This expression can be simplified by the fact that $\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2}$, For example, note that $D\frac{\partial^3 U(x,t)}{\partial t \partial x^2} = \frac{\partial}{\partial t}\left(D\frac{\partial^2 U(x,t)}{\partial x^2}\right)$ causes the first and the fourth terms to cancel [33].
Thus the truncation error is;

$$
\begin{aligned}
&\frac{(\Delta t)^2}{6}\frac{\partial^3 U(x,t_1)}{\partial t^3} + \frac{D(\Delta t)^2}{4}\frac{\partial^4 U(x,t_2)}{\partial x^2 \partial t^2} + \\
&\quad \frac{D(\Delta x)^2}{24}\left[\frac{\partial^4 U(x_1,t)}{\partial x^4} + \frac{\partial^4 U(x_2,t-\Delta t)}{\partial x^4}\right] = \frac{(\Delta t)^2}{6}\frac{\partial^3 U(x,t_1)}{\partial t^3} + \frac{D(\Delta t)^2}{4}\frac{\partial^4 U(x,t_2)}{\partial x^2 \partial t^2} + \\
&\qquad \frac{(\Delta x)^2}{24D}\left[\frac{\partial^2 U(x_1,t)}{\partial t^2} + \frac{\partial^2 U(x_2,t-\Delta t)}{\partial t^2}\right]. \qquad (2.33)
\end{aligned}
$$

To further manipulate the error, a Taylor expansion of $\frac{\partial^2 U(x,t-\Delta t)}{\partial t^2}$ in the variable t is gives;

$$
\frac{\partial^2 U(x_2,t-\Delta t)}{\partial t^2} = \frac{\partial^2 U(x_2,t)}{\partial t^2} - \Delta t\frac{\partial^3 U(x_2,t_4)}{\partial t^3}.
$$

Upon substituting into the (2.33), gives the final truncation error to be;

$$
\begin{aligned}
&\frac{5(\Delta t)^2}{12}\frac{\partial^3 U(x,t_3)}{\partial t^3} + \frac{(\Delta x)^2}{12D}\frac{\partial^2 U(x_3,t)}{\partial t^2} - \frac{(\Delta x)^2 \Delta t}{24D}\frac{\partial^3 U(x_2,t_4)}{\partial t^3} = \\
&\qquad O((\Delta x)^2) + O((\Delta t)^2) + \cdots. \qquad (2.34)
\end{aligned}
$$

In conclusion, Sauer states that the Crank-Nicolson scheme is of second order, unconditionally stable method for the heat equation [33].

Chapter 3

Results

In this chapter we look into the numerical solutions to the heat equation that was considered in the previous chapter. Before diving into numerical solutions, we can now look into the exact solution for the heat equation, which was adapted from Jiří Lebl Oklahoma State University [16] . According to Jiří Lebl, the exact solution is found by separation of variables.

3.1 Analytical Solutions

The analytical solution is given by;

$$U(x,t) = \sum_{\substack{n=1 \\ n-\text{odd}}}^{\infty} \frac{400}{n^3\pi^3} sin(n\pi x) e^{(-n^2\pi^2 0.003t)} \tag{3.1}$$

- where;
  - t is time
  - x is position
- In this case;
  - $0 \leq t \leq 100$
  - $0 \leq x \leq 1$

Up on plotting the analytical solution for different values of n we have;

- the temperature profile begins to smooth out and spread from the initial peak; 12.5 in this case,

– no temperature is lost to the environment,

– temperature diffuse from higher temperature points to points with lower temperature,

- the diffusion effect becomes more pronounced, with the profile becoming even smoother,
- the temperature continues to distribute more evenly as time progresses,
- the temperature distribution appears more uniform, reflecting extensive diffusion.

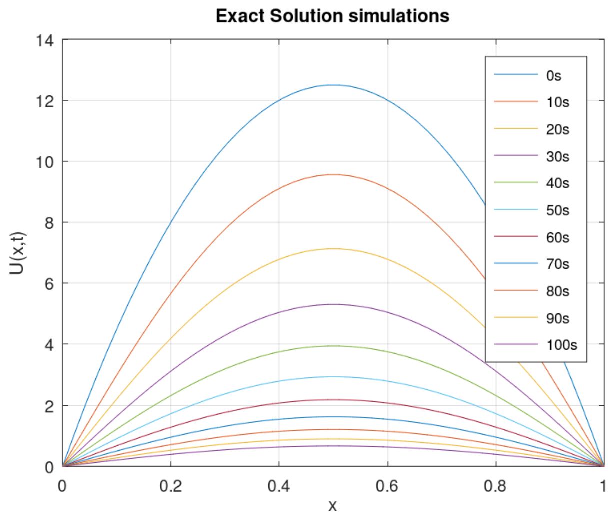

Figure 3.1: Exact solutions to the diffusion equation for the first 100 seconds produced in Octave programming language.

3.2 Numerical Solutions

With all that set and done, we will look at the numerical solutions to the heat equation, beginning with the explicit, the implicit scheme then Crank-Nicholson as discussed in the previous chapter.
We begin with the explicit scheme.

We do notice that it seemly follows the same trends as the exact solution as seen in figure (3.1) with relatively invincible error as expected.

Let us progress to the next scheme; the implicit scheme. Figure (3.3) below shows the implicit scheme solutions to the heat equation.

We also look at the Crank Nicholson scheme solutions in figure (3.4).

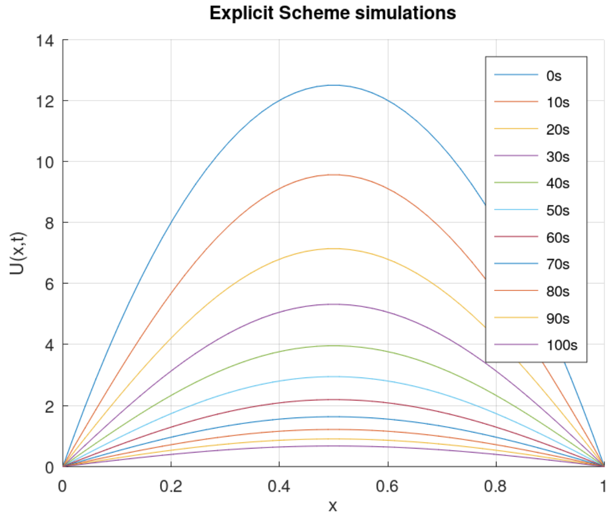

Figure 3.2: $\theta = 0$ scheme solutions to diffusion equation for the first 100 seconds produced in Octave programming language.

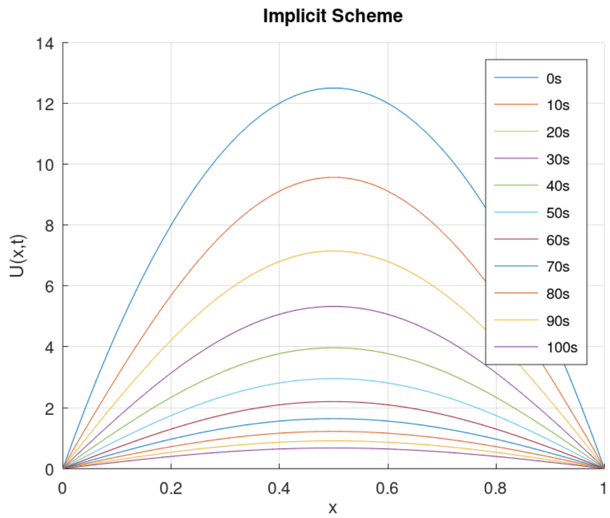

Figure 3.3: $\theta = 1$ scheme solutions to diffusion equation for the first 100 seconds produced in Octave programming language.

To get a clear picture of what is going on in the previous figures, especially in trying to avoid confusion we simulate these schemes at a fixed time and see what insight we could get from that. Below is a figure that addresses this issue.

Figure 3.5 clearly shows that there are errors associated with each scheme. This now motivates us to be interested in the how much error could each scheme posses. To lounge the error issue, we plot the errors associated with each scheme.

L_2-Norm Error

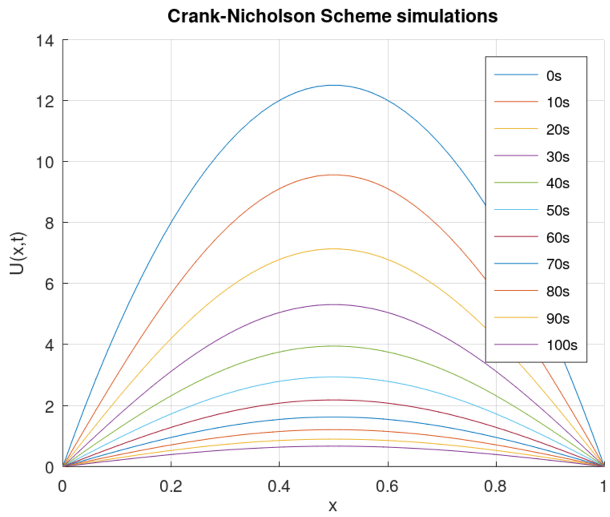

Figure 3.4: Crank Nicholson ($\theta = \frac{1}{2}$) scheme solution to diffusion equation for the first 100 seconds produced in Octave programming language.

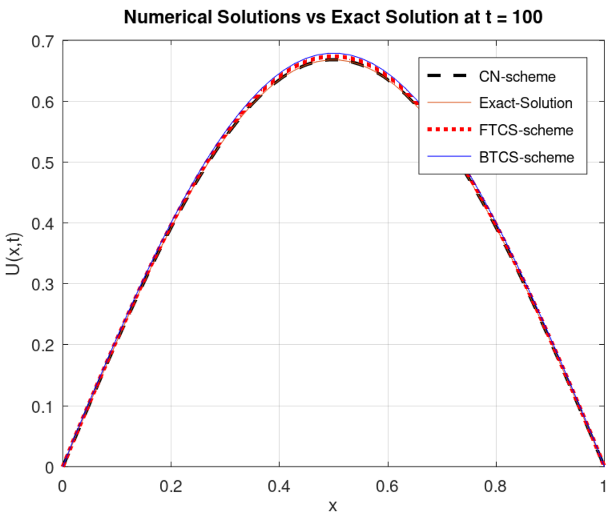

Figure 3.5: Comparison of numerical schemes against the exact/analytical solution at $t = 100s$ produced in Octave programming language.

The errors that we just looked at in figure 3.6 are not intuitive. They just give us a sense of comparing the schemes but they basically gives us no insight to the solutions we obtained. We are not going to get deep into that in this report. To refine the issue of errors, we are looking at the L_2-norm errors associated with each scheme. This L_2-norm gives the measure of the overall error. This error can be useful for assessing the accuracy and convergence of numerical methods.

The L_2-norm errors associated with the schemes are given below.

- Crank-Nicholson error $= 1.9045e^{-03}$,

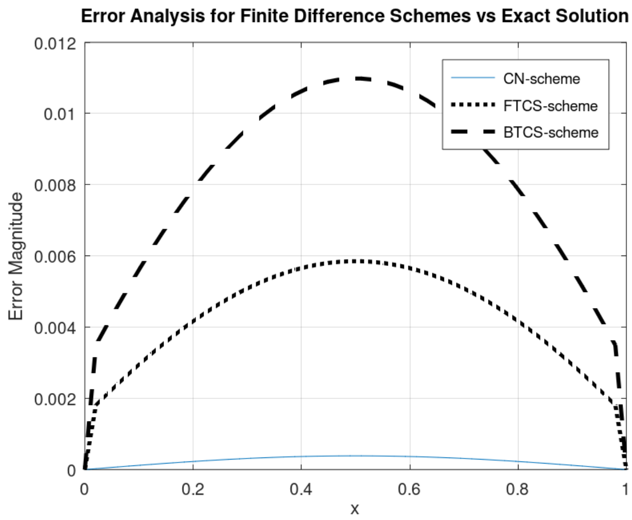

Figure 3.6: Errors of theta scheme for $\theta = 0, 1$ and $\theta = \frac{1}{2}$ at $t = 100$s produced in Octave programming language

- $\theta = 0$ scheme error $= 0.031669$,
- $\theta = 1$ scheme error $= 0.059653$.

Clearly we do realize that the Crank-Nicholson scheme is performing much better hence why it is the most famous and regularly used scheme for approximations.

Chapter 4

Conclusion

The error analysis for numerical schemes to the heat equation were successfully carried out and absolute errors found. It was found that the approximated solutions are close to the actual solution for the heat equation with some error related to each scheme. The Crank-Nicholson scheme($\theta = \frac{1}{2}$) has less error as compared to the $\theta = 0$ and $\theta = 1$ schemes. The L_2-Norm errors were found to be 0.031669 for the $\theta = 0$, 0.059653 for the $\theta = 1$ and $1.9045e^{-03}$ for the Crank- Nicholson scheme. In conclusion Crank Nicholson scheme is the scheme that has less error and is likely the most preferred for approximating the solution. It would be much clearer if we looked at the root mean square error (RMSE) which is a common metric for quantifying the error magnitude in a more intuitive sense. That is to be considered when this project is continued.

Bibliography

[1] William F Ames. *Numerical methods for partial differential equations.* Academic press, 2014.

[2] Koji Asami. Dielectric dispersion in biological cells of complex geometry simulated by the three-dimensional finite difference method. *Journal of Physics D: Applied Physics*, 39(3), 2006.

[3] Uri M Ascher, Steven J Ruuth, and Raymond J Spiteri. Implicit-explicit runge-kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, 25(2-3), 1997.

[4] Yuexing Bai, Temuer Chaolu, and Sudao Bilige. The application of improved physics-informed neural network (ipinn) method in finance. *Nonlinear Dynamics*, 107(4), 2022.

[5] Timothy J Baker. Mesh generation: Art or science? *Progress in aerospace sciences*, 41(1), 2005.

[6] Gerardine G Botte, James A Ritter, and Ralph E White. Comparison of finite difference and control volume methods for solving differential equations. *Computers & Chemical Engineering*, 24(12), 2000.

[7] M Bristeau, M Mallet, J Periaux, and G Roge. Development of finite element methods for compressible navier-stokesflow simulations in aerospace design. In *28th Aerospace Sciences Meeting*, 1990.

[8] J Carter. Some effects of spatial resolution in the calculation of slope using the spatial derivative. In *ASPRS-ACSM Annual Convention–Technical Papers*, volume 1, 1990.

[9] Juliette Chabassier and Sébastien Imperiale. Introduction and study of fourth order theta schemes for linear wave equations. *Journal of Computational and Applied Mathematics*, 245, 2013.

[10] G Chesshire and William D Henshaw. Composite overlapping meshes for the solution of partial differential equations. *Journal of Computational Physics*, 90(1), 1990.

[11] Byakatonda Denis. An overview of numerical and analytical methods for solving ordinary differential equations. *arXiv preprint arXiv:2012.07558*, 2020.

[12] Anshu Dubey, Ann Almgren, John Bell, Martin Berzins, Steve Brandt, Greg Bryan, Phillip Colella, Daniel Graves, Michael Lijewski, Frank Löffler, et al. A survey of high level frameworks in block-structured adaptive mesh refinement packages. *Journal of Parallel and Distributed Computing*, 74(12), 2014.

[13] Daniel J Duffy. *Finite difference methods in financial engineering: a partial differential equation approach*. John Wiley & Sons, 2013.

[14] Marcus R Garvie. Finite-difference schemes for reaction–diffusion equations modeling predator–prey interactions in m atlab. *Bulletin of mathematical biology*, 69, 2007.

[15] André Jaun, J Hedin, and T Johnson. Numerical methods for partial differential equations. *Swedish Netuniversity*, 1999.

[16] Jiří Lebl. Pdes, separation of variables, and the heat equation. `https://math.libretexts.org/Bookshelves/Differential_Equations/Differential_Equations_for_Engineers_(Lebl)/4%3A_Fourier_series_and_PDEs/4.06%3A_PDEs_separation_of_variables_and_the_heat_equation`, 2024 (accessed June 22, 2024).

[17] Randall J LeVeque. Finite difference methods for differential equations. *Draft version for use in AMath*, 585(6), 1998.

[18] Yanan Li, Yibin Xu, Yanqin Liu, and Yanfeng Shen. A fast θ scheme combined with the legendre spectral method for solving a fractional klein–gordon equation. *Fractal and Fractional*, 7(8), 2023.

[19] Robert J MacKinnon and Richard W Johnson. Differential-equation-based representation of truncation errors for accurate numerical simulation. *International Journal for Numerical Methods in Fluids*, 13(6), 1991.

[20] Pijus Makauskas. *Comparison of FDM, FVM with NN for solving the forward problem.* PhD thesis, Kauno technologijos universitetas, 2022.

[21] James Wayne Mercer Jr. *Finite-element approach to the modeling of hydrothermal systems.* University of Illinois at Urbana-Champaign, 1973.

[22] A Munjal and J Kaur. Numerical methods and some of their applications. In *Advanced Applications of Computational Mathematics*. River Publishers, 2022.

[23] Mitsumasa Nakajima, Kenji Tanaka, and Toshikazu Hashimoto. Neural schrödinger equation: Physical law as deep neural network. *IEEE transactions on neural networks and learning systems*, 33(6), 2021.

[24] Dai Hai Nguyen, Canh Hao Nguyen, and Hiroshi Mamitsuka. Recent advances and prospects of computational methods for metabolite identification: a review with emphasis on machine learning approaches. *Briefings in bioinformatics*, 20(6), 2019.

[25] M Necati Özişik, Helcio RB Orlande, Marcelo J Colaço, and Renato M Cotta. *Finite difference methods in heat transfer*. CRC press, 2017.

[26] Lorenzo Pareschi and Giovanni Russo. Implicit-explicit runge-kutta schemes for stiff systems of differential equations. *Recent trends in numerical analysis*, 3, 2000.

[27] RJ Pearson and DE Winterbone. The simulation of gas dynamics in engine manifolds using non-linear symmetric difference schemes. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 211(8), 1997.

[28] Nicholas Perrone and Robert Kao. A general finite difference method for arbitrary meshes. *Computers & Structures*, 5(1), 1975.

[29] Per-Olof Persson. *Mesh generation for implicit geometries*. PhD thesis, Massachusetts Institute of Technology, 2005.

[30] Christopher J Roy. Grid convergence error analysis for mixed-order numerical schemes. *AIAA journal*, 41(4), 2003.

[31] Thomas F Russell and Mary Fanett Wheeler. Finite element and finite difference methods for continuous flows in porous media. In *The mathematics of reservoir simulation*. SIAM, 1983.

[32] Mridul Sannyal, Abul Mukid Mohammad Mukaddes, Md Matiar Rahman, and MAH Mithu. Analysis of the effect of external heating in the human tissue: A finite element approach. *Polish Journal of Medical Physics and Engineering*, 26(4), 2020.

[33] Timothy Sauer. Numerical analysis pearson addison wesley, 2006.

[34] James R Scott. Solving ode initial value problems with implicit taylor series methods. Technical report, 2000.

[35] Gary A Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of computational physics*, 27(1), 1978.

[36] Arthur N Strahler. Quantitative slope analysis. *Geological Society of America Bulletin*, 67(5), 1956.

[37] Joseph F Traub. Information-based complexity. In *Encyclopedia of Computer Science*. 2003.

[38] Paul Wilmott, Sam Howison, and Jeff Dewynne. *The mathematics of financial derivatives: a student introduction*. Cambridge university press, 1995.

[39] Ernest Marshall Wright. The asymptotic expansion of integral functions defined by taylor series. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 238(795), 1940.

[40] Xingqi Zhang and Costas D Sarris. Error analysis and comparative study of numerical methods for the parabolic equation applied to tunnel propagation modeling. *IEEE Transactions on Antennas and Propagation*, 63(7), 2015.

[41] OC Zienkiewicz and JZ Zhu. Adaptivity and mesh generation. *International Journal for Numerical Methods in Engineering*, 32(4), 1991.