Numerical Solution of Diffusion Equation in One Dimension

Zana Salahaldeen Rashid Zangana

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of

Master of Science
in
Applied Mathematics and Computer Science

Eastern Mediterranean University
July 2014
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Elvan Yılmaz
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Applied Mathematics and Computer Science

Prof. Dr. Nazim Mahmudov
Chair, Department of Mathematics

We certify that we have read this thesis and that in our opinion; it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science in Applied Mathematics and Computer Science.

Assoc. Prof. Dr. Derviş Subaşı
Supervisor

Examining Committee

1. Prof. Dr. Adıguzel Dosiyev

2. Assoc. Prof. Dr. Derviş Subaşı

3. Asst. Prof. Dr. Suzan Cival Buranay

ABSTRACT

In this thesis we studied the numerical techniques for the solution of one dimensional diffusion equations. The discrete approximation of the model problem is based on different finite difference schemes. These schemes are the Explicit, Implicit, Crank Nicolson and the Weighted Average schemes. For each finite difference method we studied the local truncation error, consistency and numerical results from the solution of two model problems are considered to evaluate the performance of each scheme according to the accuracy and programming efforts.

Kay word: Diffusion equation, Finite difference method, Truncation error, Stability, Consistency, Convergence.

ÖZ

Yapılan bu çalışma tek boyutlu difüzyon differansiyel denklem problemlerinin sayısal analiz teknikleri kullanılarak çözülmesi ile ilgilidir. Bu yapılan çalışmada dört farklı sonlu farklar yöntemi problemin çözümü için kullanılmıştır. Dört farklı sonlu farklar yönteminin detaylı olarak nasıl elde edildiği, kesme hataları, stabilite şartları , yoğunluğu ve yakınsamaları detaylı olarak anlatılmıştır. Sonlu farklar metodları iki değişik problem üzerine uygulanmış ve bu metodların karşılaştırılması yapılmıştır.

Anahtar kelimeler: Difüzyon differansiyel denklem, sonlu farklar yöntemleri, kesme hatası, stabilite, yoğunluk ve yakınsama.

ACKNOWLEDGEMENT

I would like to express my sincere thanks to my supervisor Assoc. Prof. Dr. Derviş Subaşi for his patience, support and guidance which helped me during the preparation of this thesis.

My sincere thanks also goes to Mathematics Department at EMU for their support and assistance.

I would also like to thank my parent and all my family for their unconditional support throughout my study.

Finally, I would like to thank all my friends for their support and encouragement me with their best wishes.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

The diffusion equation (or heat equation) is of fundamental importance in scientific fields and engineering problem. The one dimensional diffusion equation is

$$u_t = \alpha u_{xx} \qquad\qquad 0 < x < L \ \ , 0 < t < T, \qquad\qquad (1.1)$$

where, $u = u(x,t)$ is the dependent variable, and $\propto$ is a constant coefficient. To solve equation (1.1), it is required a specific initial condition at $t = 0$, given

$$u(x,0) = f(x) \qquad\qquad 0 \le x \le L, \qquad\qquad (1.2)$$

and boundary conditions at $x = 0$ and $x = L$. The general form of boundary conditions is

$$\gamma_1 u(0,t) + \beta_1 u_x(0,t) = g_1(t)$$

$$\gamma_2 u(L,t) + \beta_2 u_x(L,t) = g_2(t) \qquad\qquad (1.3)$$

The solution of (1.1) with (1.2) and (1.3) is to find $u(x,t)$, satisfying the boundary conditions as follows [1].

1) If $\gamma_i \neq 0$ and $\beta_i = 0$, then equation (1.3) gives Dirichlet boundary condition
2) If $\gamma_i = 0$ and $\beta_i \neq 0$, then equation (1.3) gives Neumann boundary condition
3) If $\gamma_1 \neq 0$ and $\beta_1 = 0$ and $\gamma_2 = 0$ and $\beta_2 \neq 0$ or If $\gamma_1 = 0$ and $\beta_1 \neq 0$ and $\gamma_2 \neq 0$ and $\beta_2 = 0$ equation (1.3) gives mixed boundary conditions.

The solution of the one dimensional diffusion equation using several finite difference methods with Dirichlet and Neumann type boundary conditions is the core of study in this thesis.

The derivation of each finite difference scheme for Dirichlet and Neumann type boundary conditions are discussed in chapter 2.

In Chapter 3, we presented local truncation error, consistency, stability and convergence of finite difference scheme.

In Chapter 4, we presented the numerical result from solving two module problems. Concluding remarks are given each module problem.

In Chapter 5, general conclusion from work are presented

Chapter 2

THE FINITE DIFFERENCE METHOD

In this Chapter we focus on finite difference methods (FDMs), which are widely used and are the most straight forward numerical approach for solving PDE's. These methods are derived from the truncated Taylor's series where a given PDE and boundary and initial conditions are replaced by a set of algebraic equations that are then solved by several well-known numerical techniques. We analyzed different schemes for first and second order derivatives then applied them to discretize diffusion equation with initial and boundary conditions.

2.1 Taylor Series and Difference Approximations for Derivative terms in PDE's

Let us consider in case of the function $u(x,t)$ of two independent variables x and t. We first partition the spatial interval $[0,L]$ and temporal interval $[0,T]$ into respective finite grids as follows.

$$x_i = i\Delta x \qquad i = 0,1,\ldots\ldots N \qquad \text{where} \qquad \frac{L}{N} = \Delta x. \tag{2.1.1}$$

$$t_j = j\Delta t \qquad j = 0,1,\ldots\ldots,M \qquad \text{where} \qquad \frac{T}{M} = \Delta t. \tag{2.1.2}$$

The numerical solution to the PDE's is an approximation to the exact solution that is obtained using a discrete representation to the PDE at the grid point x_i in the discrete spatial mesh at every time level t_j (see Fig 2.1) [7].

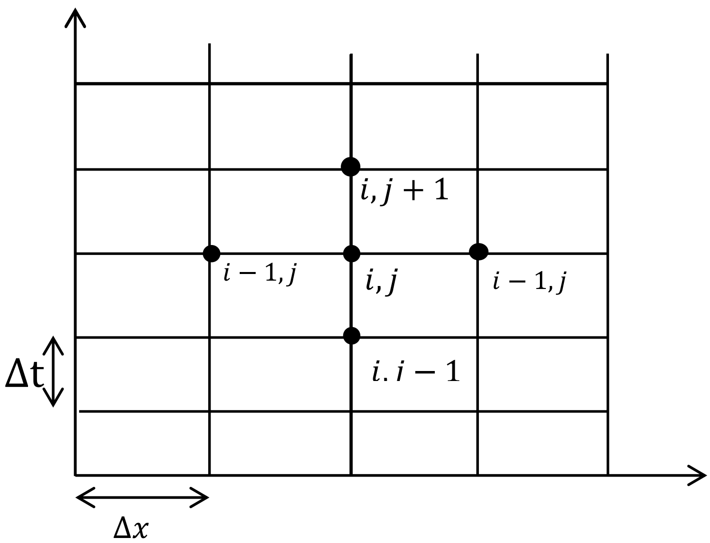


Figure 2.1: The finite difference grid in the solution region

Let us denote the numerical solution of $u(x, t)$ such that

$$u_{i,j} = u(x_i, t_i) \tag{2.1.3}$$

Consider the Taylor series for $u_{i+1,j}$, $u_{i-1,j}$ and $u_{i,j+1}$ respectively [2].

$$u_{i+1,j} = u_{i,j} + \Delta x \frac{\partial u_{i,j}}{\partial x} + \frac{(\Delta x)^2}{2} \frac{\partial^2 u_{i,j}}{\partial x^2} + O((\Delta \mathrm{x})^3) \tag{2.1.4}$$

$$u_{i-1,j} = u_{i,j} - \Delta x \frac{\partial u_{i,j}}{\partial x} + \frac{(\Delta x)^2}{2} \frac{\partial^2 u_{i,j}}{\partial x^2} + O((\Delta \mathrm{x})^3) \tag{2.1.5}$$

$$u_{i,j+1} = u_{i,j} + \Delta t \frac{\partial u_{i,j}}{\partial t} + \frac{(\Delta t)^2}{2} \frac{\partial^2 u_{i,j}}{\partial t^2} + O((\Delta \mathrm{x})^3) \tag{2.1.6}$$

If we only consider $O(\Delta x)$ terms in equation (2.1.4) and (2.1.5) then we arrive at the forward and backward difference approximation for u_x respectively.

$$\left(\frac{\partial u}{\partial x}\right)_{ij} = \frac{u_{i+1,j} - u_{i,j}}{\Delta x} + O(\Delta x) \tag{2.1.7}$$

$$\left(\frac{\partial u}{\partial x}\right)_{ij} = \frac{u_{i,j} - u_{i-1,j}}{\Delta x} + O(\Delta x) \tag{2.1.8}$$

If we only consider $O(\Delta t)$ terms in equation (2.1.6) then we arrive at the forward difference approximation for u_t.

$$\left(\frac{\partial u}{\partial t}\right)_{ij} = \frac{u_{i,j+1} - u_{i,j}}{\Delta t} + O(\Delta t) \tag{2.1.9}$$

We can also derive a higher order approximation for u_x by subtracting (2.1.5) from (2.1.4), then we obtain at the central difference in space approximation for u_x.

$$\left(\frac{\partial u}{\partial x}\right)_{ij} = \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + O((\Delta \mathrm{x})^2). \tag{2.110}$$

We can also perform similar approach to obtain an approximation for the second derivative u_{xx}. To achieve the central difference for the second derivative in space, add Eq. (2.1.4) and Eq. (2.1.5), solve expansion for $\frac{\partial^2 u}{\partial x^2}$ and the result is written by

$$\left(\frac{\partial^2 u}{\partial x^2}\right) = \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{(\Delta x)^2} + O((\Delta \mathrm{x})^2) \tag{2.1.11}$$

2.1.1 Explicit Method (FTCS)

The explicit finite difference method based on forward difference approximation of first order derivative.

$$\left(\frac{\partial u}{\partial t}\right)_{ij} = \frac{u_{i,j+1} - u_{i,j}}{\Delta t}, \tag{2.1.12}$$

also based on the central difference approximation to second order derivative.

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} \quad ,$$

and substituting these in Equation (1.1) results

$$\frac{u_{i,j+1} - u_{i,j}}{\Delta t} = \alpha \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} \quad . \tag{2.1.13}$$

In explicit finite difference method, the temperature at time $j+1$ depends on the temperature at time j, shown as in Figure (2.2). Solving $u_{i,j+1}$ in Eq. (2.1.13we get.

$$u_{i,j+1} - u_{i,j} = \frac{\alpha \Delta t}{\Delta x^2}\left(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}\right) \quad , \tag{2.1.14}$$

where $r = \frac{\alpha \Delta t}{(\Delta x)^2}$

$$u_{i,j+1} = u_{i,j} + r\left(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}\right) \tag{2.1.15}$$

Therefore

$$u_{i,j+1} = r u_{i+1,j} + (1-2r) u_{i,j} + u_{i-1,j} \quad . \tag{2.1.16}$$

Eq. (2.1.16) is called explicit finite difference or Forward Time Center Space (FTCS) approximation to the heat equation given in (1.1) [5].

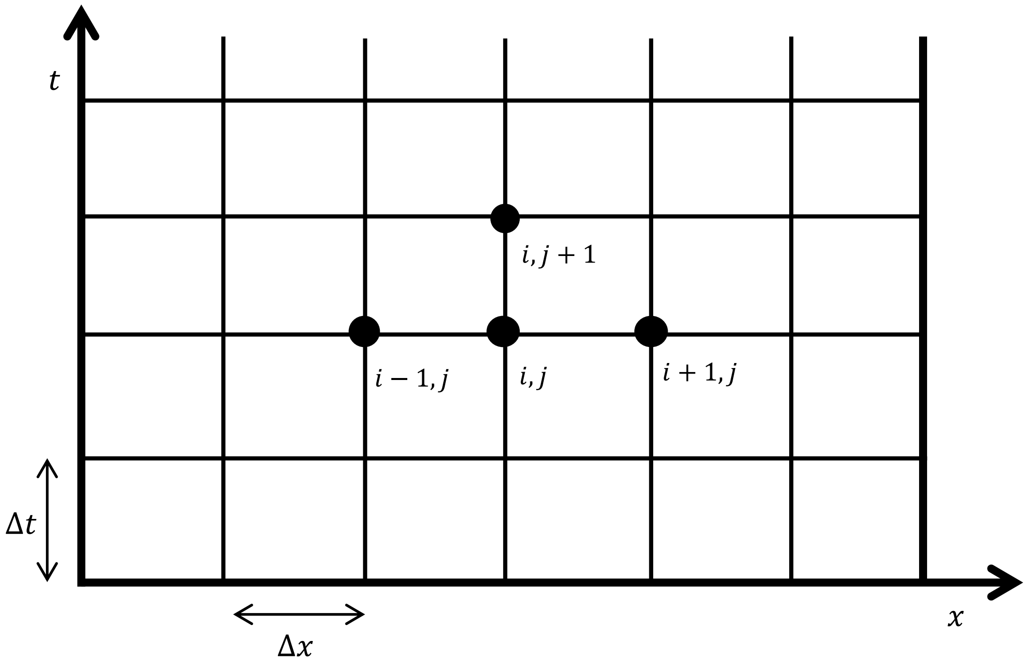


Figure 2.2: Represent point scheme for FTCS

Furthermore we can rewrite Eq. (2.1.16) in matrix vector form as;

$$\begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ . \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j+1} \end{bmatrix} = \begin{bmatrix} 1-2r & r & 0 & & . \\ r & 1-2r & r & & . \\ & . & & & . \\ & & . & & . \\ & & & & . \\ & & & & \\ & & & . & . \\ & & r & 1-2r & r \\ . & . & . & r & 1-2r \end{bmatrix} \begin{bmatrix} u_{1,j} + r u_{0,j} \\ u_{2,j} \\ . \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,} + r u_{N,j} \end{bmatrix} \quad (2.1.17)$$

2.1.2 Implicit Method (BTCS)

We can derive the implicit method by substituting forward difference approximation (2.1.9) in left hand side of (1.1) and central difference approximation at time $(j+1)$ in the right hand side of (1.1) [5].

$$\frac{u_{i,j+1} - u_{i,j}}{\Delta t} = \alpha \frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{(\Delta x)^2} \quad (2.1.18)$$

Now arrange Eq. (2.1.18) to get

$$u_{i,j+1} - u_{i,j} = \frac{\alpha\Delta t}{(\Delta x)^2}\left(u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}\right), \qquad (2.1.19)$$

where $r = \frac{\alpha\Delta t}{(\Delta x)^2}$ then (2.1.19) is given by;

$$u_{i,j+1} - u_{i,j} = r\left(u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}\right). \qquad (2.1.20)$$

Therefore

$$-ru_{i-1,j+1} + (1+2r)u_{i,j+1} - ru_{i+1,j+1} = u_{i,j} \qquad i = 1,2,3, \ldots\ldots, N-1 \quad (2.1.21)$$

The equation (2.1.21) is known as Backward Time, Centered Space (BTCS) or Implicit Method. In implicit method there are more terms in level above than those in level below is shown in Figure (2.3). Consequently, the equation cannot be reorganized to gain easy algebraic formula similar to the explicit method to determine $u_{i,j+1}$ [6]. Although this is a disadvantage of implicit method, it has the advantage of being unconditionally stable [3].

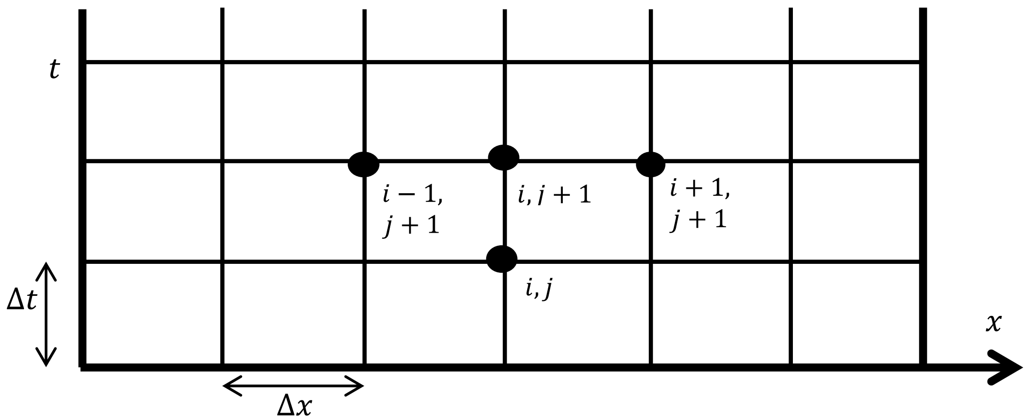


Figure 2.3: Represents point scheme for BTCS

Equation (2.1.21) gives us a set of linear equations at every spatial point $u_{i,j}$, and they will be solved correctly through the use of matrix method [5],

where $1 \leq i \leq N-1$ and $u_{0,j}$, $u_{N,j}$ are fixed because they are boundary conditions;

If $i = 1$

$$(1+2r)u_{1,j+1} - ru_{2,j+1} = u_{1,j} + ru_{0,j}\ , \tag{2.1.22a}$$

$1 < i < N-1$

$$-ru_{N-1,j+1} + (1+2r)u_{N,j+1} - ru_{N+1,j+1} = u_{N,j} \tag{2.1.22b}$$

$i = N-1$

$$-ru_{N-2,j+1} + (1-2r)u_{N-1,j+1} = u_{N-1,j} + ru_{N,j} \tag{2.1.22c}$$

We have a set of linear equations. The unknowns are on the left hand side of the equation and they give us a tri-diagonal matrix to solve equation $(2.1.22a - 2.1.22c)$. The tridiagonal matrix will be in this form [3].

$$
\begin{bmatrix}
(1+2r) & -r & 0 & 0 & . & . \\
-r & (1+2r & -r & & & . \\
0 & . & & & & . \\
0 & & . & & & . \\
. & & & & & . \\
. & & & . & & . \\
. & & & & & . \\
. & & & & & . \\
. & & & & . & . \\
. & & & -r & (1+2r) & -r \\
. & . & . & . & -r & (1+2r)
\end{bmatrix}
\begin{bmatrix}
u_{1,j+1} \\ u_{2,j+1} \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j+1}
\end{bmatrix}
=
\begin{bmatrix}
u_{1,j} + ru_{0,j} \\ u_{2,j} \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j} + ru_{N,j}
\end{bmatrix}
\tag{2.1.23}
$$

2.1.3 Crank Nicolson Method

Crank Nicolson method is a popular method to use for parabolic equations since it is second order accurate and unconditionally stable. This method is implicit, but different from simple implicit (BTCS) method explained in the former Section, as in this method the right hand is chosen at time j and at time $(j + 1)$ is shown in Figure (2.4) [7].

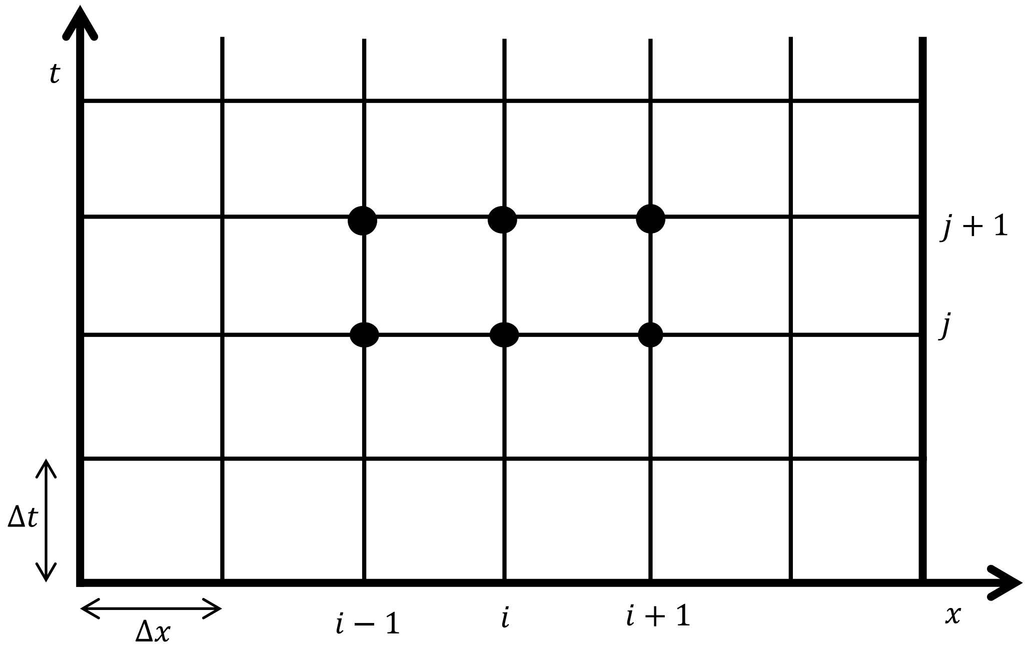


Figure 2.4: Represent point scheme for Crank Nicolson Method

$$\left(\frac{\partial \mathrm{u}}{\partial t}\right)_{i,j} = \frac{u_{i,j+1} - u_{i,j}}{\Delta t} \tag{2.1.24}$$

$$\left(\frac{\partial^2 \mathrm{u}}{\partial x^2}\right)_{i,j} = \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{(\Delta x)^2} \tag{2.1.25}$$

$$\left(\frac{\partial^2 \mathrm{u}}{\partial x^2}\right)_{i,j+1} = \frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{(\Delta x)^2} \tag{2.1.26}$$

Consider the heat equation (1.1) at midpoint $(x_i, t_{j+\frac{1}{2}})$ and instead of $(\frac{\partial^2 u}{\partial x^2})$ put average of central difference $(i, j+\frac{1}{2})$ [3].

$$\left(\frac{\partial u}{\partial t}\right)_{i,j+\frac{1}{2}} = \left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j+\frac{1}{2}} \tag{2.1.27}$$

$$\frac{u_{i,j+1} - u_{i,j}}{\Delta t} = \alpha \frac{1}{2}\left[\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + \frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{(\Delta x)^2}\right] \tag{2.1.28}$$

Therefore

$$u_{i,j+1} - u_{i,j} = \frac{\alpha\Delta t}{2(\Delta x)^2}\left[u_{i+1,j} - 2u_{i,j} + u_{i-1,j} + u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}\right] \quad (2.1.29)$$

where $r = \frac{\alpha\Delta t}{(\Delta x)^2}$ then (2.1.19) it will be;

$$\left(u_{i,j+1} - u_{i,j}\right) = r\left[u_{i+1,j} - 2u_{i,j} + u_{i-1,j} + u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}\right] \quad (2.1.30)$$

Separate j on one side and $(j + 1)$ on the another side of equation (2.1.30) giving

$$-ru_{i-1,j+1} + (2 + 2r)u_{i,j+1} - ru_{i+1,j+1} = ru_{i-1,j} + (2 - 2r)u_{i,j} + ru_{i+1,j} \ , \qquad (2.1.31)$$

where $i = 1,2,3, \dots \dots .., N - 1$.

Generally the right hand side of equation $(2.1.31)$ contains three known values and left hand contains three unknowns, Implicit method generate a set of (N-1) linear equation, which should be solved at each time level. The set of equations generate a tridiagonal matrix and can be solved by Thomas algorithm [3].

The Crank-Nicholson method can be written in a matrix vector form is as follows.

$$\begin{bmatrix} 2+2r & -r & . & . & . & . & . \\ -r & 2+2r & -r & & & & . \\ 0 & & . & & & & . \\ . & & & . & & & . \\ . & & & & . & & . \\ . & & & & -r & 2+2r & -r \\ . & . & . & . & . & -r & 2+2r \end{bmatrix} \begin{bmatrix} u_{1,j+1} \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j+1} \end{bmatrix} + \begin{bmatrix} -ru_{0,j+1} \\ 0 \\ . \\ . \\ . \\ . \\ . \\ . \\ -ru_{N,j+1} \end{bmatrix} =$$

$$\begin{bmatrix} 2-2r & r & . & . & . & . & . \\ r & 2-2r & r & & & & . \\ 0 & & . & & & & . \\ . & & & . & & & . \\ . & & & & . & & . \\ . & & & & r & 2-2r & r \\ . & . & . & . & 0 & r & 2-2r \end{bmatrix} \begin{bmatrix} u_{1,j} \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j} \end{bmatrix} + \begin{bmatrix} ru_{0,j} \\ o \\ . \\ . \\ . \\ . \\ . \\ . \\ ru_{N,j} \end{bmatrix} \qquad (2.1.32)$$

2.1.4 The Method of Weighted Averages

In this method we use two finite difference approximation to $\frac{\partial^2 u}{\partial^2 x}$ in Eq, (1.1), first one by three points in level below t_j, the other one uses three pointe on level above t_{j+1}. The left hand use forward difference approximation is used for the first derivative $\frac{\partial u}{\partial t}$ [5].

$$\frac{u_{i,j+1} - u_{i.j}}{\Delta t} = \alpha \left[\theta \left(\frac{\partial^2 u}{\partial x^2} \right)_{i,j+1} + (1-\theta) \left(\frac{\partial^2 u}{\partial x^2} \right)_{i,j} \right] . \qquad (2.1.33)$$

Substitute $\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j+1}$, $\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j}$ and rearranging, the equation (2.1.33), gives

$$\frac{u_{i,j+1} - u_{i.j}}{\Delta t} = \frac{\alpha}{(\Delta x)^2} [\theta \left(u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1} \right)$$

$$+ (1-\theta)(u_{i+1,j} - 2u_{i,j} + u_{i-1,j})] \qquad (2.1.34)$$

$$u_{i,j+1} - u_{i.j} = \frac{\alpha \Delta t}{(\Delta x)^2}[\theta\big(u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}\big)$$
$$+(1-\theta)(u_{i+1,j} - 2u_{i,j} + u_{i-1,j})], \qquad (2.1.35)$$

taking $r = \frac{\alpha \Delta t}{(\Delta x)^2}$ and then (2.1.35) takes the form

$$u_{i,j+1} - u_{i.j} = r[\theta\big(u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}\big)$$
$$+(1-\theta)(u_{i+1,j} - 2u_{i,j} + u_{i-1,j})] \qquad (2.1.36)$$

The formula (2.1.36) as known as weighted average or θ −method is shown in Figure (2.5). Where θ is non-negative weights $0 \leq \theta \leq 1$. If $\theta = 0,1,\frac{1}{2}$ from equation (2.1.36) we obtain Explicit, Implicit and Crank Nicolson method respectively. The equation (2.1.36) is stable for any $\frac{1}{2} \leq \theta \leq 1$, but for $0 \leq \theta < \frac{1}{2}$ to be stable $r \leq \frac{1}{2}\ (1-2\theta)^{-1}$ [3].

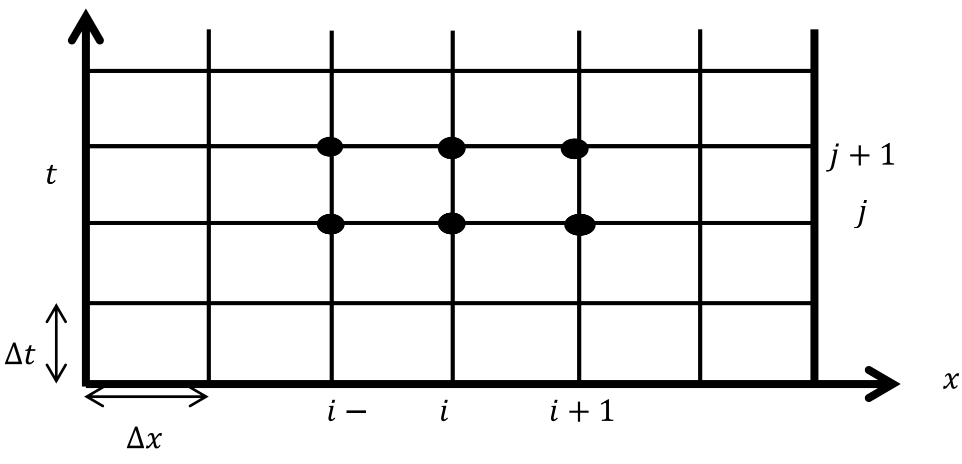


Figure 2.5: Represent point scheme for weighted average approximation

To system of equation in $(2.1.36)$ where u at time level j is known and we want to find u at time level $j+1$ is

$$-r\theta u_{i-1,j+1} + (1+2r\theta)u_{i,j+1} - r\theta u_{i+1,j+1} = r(1-\theta)u_{i-1,j} +$$

$$[1-2r(1-\theta)]u_{i,j} + r(1-\theta)u_{i+1,j} \qquad i = 1,2,3,\ldots\ldots,N-1 \quad . \qquad (2.1.37)$$

Here $u_{0,j+1}$ and $u_{N,j+1}$ as being known the Eq, (2.1.37) generate a set of $(n-1)$ linear equations which the coefficient matrix is tridiagonal [5]. Which can be solved by Thomas Algorithm [3]. It is suitable to write (2.1.37) in vector form, so let

$$u^j = \left[u_{1,j}, u_{2,j}, \ldots\ldots\ldots\ldots.., u_{N-1,j}\right]^T \ .$$

Than we can write Eq. (2.1.37) as;

$$[\mathrm{I} - r\theta C]u^{j+1} = [\mathrm{I} + r(1-\theta)C]u^j + rf^n \quad , \qquad (2.1.38)$$

where

$$C = \begin{bmatrix} -2 & 1 & . & . & . & . & . \\ 1 & -2 & 1 & & & & . \\ . & & . & & & & . \\ . & & & . & & & . \ . \\ . & & & & . & & . \\ . & & & & 1 & -2 & 1 \\ . & . & . & . & . & 1 & -2 \end{bmatrix}, f^n = \begin{bmatrix} \theta u_{0,j+1} + (1-\theta)u_{0,j} \\ 0 \\ . \\ . \\ . \\ . \\ . \\ 0 \\ \theta u_{N,j+1} + (1-\theta)u_{N,j+1} \end{bmatrix} \qquad (2.1.39)$$

2.2 Neumann Boundary Condition

In previous section we have considered the problems with Dirichlet boundary conditions. Now we consider problems with Neumann boundary condition. From Eq. (1.3), if $\gamma = 0$ and $\beta \neq 0$ we have.

$$u_x(0,t) = g_1(t)\,, \qquad u_x(L,t) = g_2(t) \tag{2.2.1}$$

Which has Neumann condition at $x = 0\,,\ x = L$.

It is possible to use forward or backward difference to represent Neumann boundary condition at left and right end of the domain, but it is generally preferable to use central difference formula by introducing the fictitious temperature $u_{i-1,j}$ at the external grid point $x = (i-1)\Delta x$ and as shown in Figure (2.5). The boundary condition at $i - 1$ is represented by Figure (2.6) [3].

$$(u_x)_{0,j} = \frac{u_{1,j} - u_{-1,j}}{2\Delta x} \tag{2.2.2}$$

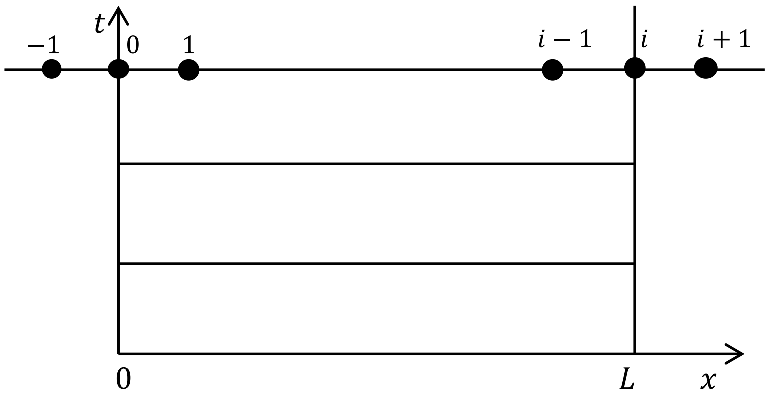


Figure 2.5: Introduce fictitious temperature

Also introduce u_{i+1} at the end of the rod at the external grid point $x = (i+1)\Delta x$. The boundary condition at $i+1$ can be represent by [3].

$$(u_x)_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} \tag{2.2.3}$$

The temperatures $u_{-1,j}$ and $u_{i+1,j}$ are unknown and this leads to more equations. It is possible to eliminated $u_{-1,j}$ and $u_{i+1,j}$ between these equations. These methods are applied to find boundary condition in following schemes [3].

2.3.1 Explicit Method with Neumann Boundary

Consider explicit method representation of Eq.(2.1.16)

$$u_{i,j+1} = ru_{i+1,j} + (1-2r)u_{i,j} + u_{i-1,j}\ ,$$

at $x = 0$ gives us

$$u_{0,j+1} = u_{0,j} + r\big(u_{-1,j} - 2u_{0,j} + u_{1,j}\big)\ . \tag{2.2.4}$$

Applying central difference for the boundary at $x = 0$ than we obtain,

$$(u_x)_{0,j} = \frac{u_{1,j} - u_{-1,j}}{2\Delta x} \tag{2.2.5}$$

Substitute into (2.2.1) we get an approximation of the Neumann condition at $(0, j\Delta t)$ as

$$u_{-1,j} = u_{1,j} - 2\Delta x g_1(j\Delta t)\ . \tag{2.2.6}$$

Use Eq, (2.2.6) to discretize explicit method (2.2.4) resulting

$$u_{0,j+1} = (1-2r)u_{0,j} + 2ru_{1,j} - 2r\Delta x g_1(j\Delta t) \tag{2.2.7}$$

Now, consider explicit method at $x = L = N\Delta x$

$$u_{N,j+1} = u_{N,j} + r\big(u_{N-1,j} - 2u_{N,j} + u_{N+1,j}\big)\,. \tag{2.2.8}$$

We apply the following central difference formula for right boundary condition at $x = L$,

$$(u_x)_{N,j} = \frac{u_{N+1,j} - u_{N-1,j}}{2\Delta x}\ . \tag{2.2.9}$$

Substitute into (2.2.1) we obtain an approximation of the Neumann condition at $\big((N+1)\Delta x, j\Delta t\big)$ as

$$u_{N+1,j} = u_{N-1,j} + 2\Delta x g_2(j\Delta t)\ . \tag{2.2.10}$$

Use Eq(2.2.10) to discretize explicit method (2.2.8) than gives as

$$u_{N,j+1} = 2ru_{N-1} + (1-2r)u_{N,j} + 2r\Delta x g_2(j\Delta t)\quad . \tag{2.2.11}$$

Matrix form as (2.2.11) can be written in

$$\begin{bmatrix} u_{0,j+1} \\ u_{1,j+1} \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j+1} \\ u_{N,j+1} \end{bmatrix} = \begin{bmatrix} 1-2r & 2r & 0 & . & . & . & . \\ r & 1-2r & r & & & & . \\ 0 & & . & & & & . \\ . & & & . & & & . \\ . & & & & . & & . \\ . & & & & r & 1-2r & r \\ . & . & . & . & . & 2r & 1-2r \end{bmatrix} \begin{bmatrix} u_{0,j} \\ u_{1,j} \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j} \\ u_{N,j} \end{bmatrix}$$

$$+ \begin{bmatrix} -2r\Delta x g_1(j\Delta t) \\ 0 \\ . \\ . \\ . \\ . \\ . \\ . \\ 2r\Delta x g_2(j\Delta t) \end{bmatrix} \tag{2.2.12}$$

2.3.2 Implicit Method with Neumann Boundary

Consider implicit method represented as the following;

$$-ru_{i-1,j+1} + (1+2r)u_{i,j+1} - ru_{i+1,j+1} = u_{i,j} \quad , \tag{2.2.13}$$

at $x = 0$ gives us

$$-ru_{-1,j+1} + (1+2r)u_{0,J+1} - ru_{1,j+1} = u_{0,j} \quad . \tag{2.2.14}$$

Applying central difference for the boundary at $x = 0$ than we obtain

$$(u_x)_{0,j+1} = \frac{u_{1,j+1} - u_{-1,j+1}}{2\Delta x} \quad , \tag{2.2.15}$$

substitute into (2.2.1) we obtain an approximation of the Neumann condition at $(0, j\Delta t)$ as

$$u_{-1,j+1} = u_{1,j+1} - 2\Delta x g_1(j\Delta t) \quad . \tag{2.2.16}$$

Use Eq(2.2.16) to discretize explicit method (2.2.14) than gives as

$$(1+2r)u_{0,j+1} - 2ru_{1,j+1} + 2r\Delta x g_1(j\Delta t) = u_{0,j} \quad . \tag{2.2.17}$$

Now, consider implicit method at $x = L$

$$u_{N,j+1} - r\left(u_{N-1,j+1} - 2u_{N,j} + u_{N+1,j+1}\right) = u_{N,j} \quad . \tag{2.2.18}$$

We apply central difference for right boundary condition at $x = L$ we get

$$u_{N,j+1} = \frac{u_{N+1,j+1} - u_{N-1,j+1}}{2\Delta x} \tag{2.2.19}$$

Substitute into (2.2.1) we obtain an approximation of the Neumann condition at

$\big((N+1)\Delta x,(j+1)\Delta t\big)$ as

$$u_{N+1,j+1} = u_{N-1,j+1} + 2\Delta x g_2((j+1)\Delta t) \quad . \tag{2.2.20}$$

Use Eq(2.2.20) to discretize implicit method (2.2.18) than gives as

$$-2ru_{N-1,j+1} + (1+2r)u_{N,j+1} - 2r\Delta x g_2\big((j+1)\Delta t\big) = u_{N,j} \ . \tag{2.2.21}$$

We can write in matrix form

$$\begin{bmatrix} 1+2r & -2r & .. & . & . & . & . \\ -r & 1+2r & -r & & & & . \\ . & & . & & & & . \\ . & & & . & & & . \\ . & & & & . & & . \\ . & & & & -r & 1+2r & -r \\ . & . & . & . & . & -2r & 1+2r \end{bmatrix} \begin{bmatrix} u_{0,j+1} \\ u_{1,J+1} \\ . \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j+1} \\ u_{N,j+1} \end{bmatrix} = \begin{bmatrix} u_{0,j} \\ u_{1,j} \\ . \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j} \\ u_{N,j} \end{bmatrix}$$

$$+ \begin{bmatrix} 2r\Delta x g_1((j+1)\Delta t) \\ 0 \\ . \\ . \\ . \\ . \\ . \\ . \\ 0 \\ 2r\Delta x g_2((j+1)\Delta t) \end{bmatrix} \tag{2.2.22}$$

2.3.3 Crank Nicolson Method with Neumann Boundary

Consider Crank Nicolson method represent as follow

$$-ru_{i-1,j+1} + (2+2r)u_{i,j+1} - ru_{i+1,j+1} =$$

$$ru_{i-1,j} + (2-2r)u_{i,j} + ru_{i+1,j} \ , \qquad (2.2.23)$$

at $x = 0$ gives as

$$-ru_{-1,j+1} + (2+2r)u_{0,j+1} - ru_{1,j+1} = ru_{-1,j} + (2-2r)u_{0,j} + ru_{1,j} \ . \quad (2.2.24)$$

Applying central difference for the boundary at $x = 0$ at time level $j + 1$ and j than we obtain

$$\frac{u_{1,j} - u_{-1,j}}{2\Delta x} = (u_x)_{0,j} \quad , \quad \frac{u_{1,j+1} - u_{-1,j+1}}{2\Delta x} = (u_x)_{0,j+1} \quad , \qquad (2.2.25)$$

substitute into (2.2.1) we obtain an approximation $(0, j\Delta t)$ as

$$u_{-1,j} = u_{1,j} - 2\Delta x g_1(j\Delta t) \quad , \quad u_{-1,j+1} = u_{1,j+1} - 2\Delta x g_1\big((j+1)\Delta t\big) \ . \ (2.2.26)$$

Use Eq(2.2.26) to discretize Crank Nicolson method (2.2.24) than gives as

$$(2+2r)u_{0,j+1} - 2ru_{1,j+1} + 2r\Delta x g_1((j+1)\Delta t)$$

$$= (2-2r)u_{0,j} + 2ru_{1,j} - 2r\Delta x g_1(j\Delta t) \quad , \qquad (2.2.27)$$

at $x = L$ gives as

$$-ru_{N-1,j+1} + (2+2r)u_{N,j+1} - ru_{N+1,j+1} =$$

$$ru_{N-1,j} + (2-2r)u_{N,j} + ru_{N+1,j} \quad (2.2.28)$$

Now, we apply central difference for right boundary condition at $x = L$ to find left side

$$(u_x)_{N,j} = \frac{u_{N+1,j} - u_{N-1,j}}{2\Delta x} \quad , \quad (u_x)_{N,j+1} = \frac{u_{N+1,j+1} - u_{N-1,j+1}}{2\Delta x} \qquad (2.2.29)$$

Substitute into (2.2.1) we obtain an approximation of the Neumann condition at $\big((N+1)\Delta x, j\Delta t\big)$ as

$$u_{N+1,j} = u_{N-1,j} + 2\Delta g_2(j\Delta t)\,, \quad u_{N+1,j+1} = u_{N-1,j+1} + 2\Delta x g_2((j+1)\Delta t) \quad (2.2.30)$$

Use Eq(2.2.30) to discretize Crank Nicolson method (2.2.28) than gives as

$$-2ru_{N-1,j+1} + (2+2r)u_{N,j+1} - 2r\Delta x g_2((j+1)\Delta t) =$$

$$2ru_{i-1,j} + (2-2r)u_{i,j} + 2r\Delta x g_2(j\Delta t) \qquad (2.2.31)$$

We can write in tire-diagonal matrix form

$$\begin{bmatrix} 2+2r & -2r & . & . & . & . & . \\ -r & 2+2r & -r & & & & . \\ . & & . & & & & . \\ . & & & . & & & . \\ . & & & & . & & . \\ . & & & & -r & 2+2r & -r \\ . & . & . & . & . & -2r & 2+2r \end{bmatrix} \begin{bmatrix} u_{0,j+1} \\ u_{1,j+1} \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j+1} \\ u_{N,j+1} \end{bmatrix} + \begin{bmatrix} 2r\Delta x g_1((j+1)\Delta t) \\ 0 \\ . \\ . \\ . \\ . \\ . \\ . \\ -2r\Delta x g_2((j+1)\Delta t) \end{bmatrix} =$$

$$\begin{bmatrix} 2-2r & 2r & . & . & . & . & . \\ r & 2-2r & r & & & & . \\ . & & . & & & & . \\ . & & & . & & & . \\ . & & & & . & & . \\ . & & & & r & 2-2r & r \\ . & . & . & . & . & 2r & 2-2r \end{bmatrix} \begin{bmatrix} u_{0,j} \\ u_{1,j} \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j} \\ u_{N,j} \end{bmatrix} + \begin{bmatrix} -2r\Delta x g_1(j\Delta t) \\ 0 \\ . \\ . \\ . \\ . \\ . \\ . \\ 2r\Delta x g_2(j\Delta t) \end{bmatrix} \quad (2.2.32)$$

2.3.4 Weighted Average Approximation with Neumann Boundary

Consider weighted average method at $x = 0$

$$-r\theta u_{-1,j+1} + (1+2r\theta)u_{0,j+1} - r\theta u_{1,j+1} = r(1-\theta)u_{-1,j} +$$
$$[1-2r(1-\theta)]u_{0,j} + r(1-\theta)u_{1,j} \quad . \quad (2.2.33)$$

Applying central difference for the boundary at $x = 0$ at time level $j + 1$ and j than we obtain (2.2.25). Substitute into (2.2.1) we obtain an approximation of the Neumann condition at $(0, j\Delta t)$ as

$$u_{-1,j+1} = u_{1,j+1} - 2\Delta x g_1\big((j+1)\Delta t\big) \ , \ u_{-1,j} = u_{1,j} - 2\Delta x g_1(j\Delta t) \quad . \qquad (2.2.34)$$

Use equation (2.2.34) to discretize Weight average method (2.2.33) than gives as

$$(1 + 2r\theta)u_{0,j+1} - 2r\theta u_{1,j+1} + 2r\Delta x g_1((j+1)\Delta t) = [1 - 2r(1-\theta)]u_{0,j} + 2r(1-\theta)u_{1,j} - 2r(1-\theta)\Delta x g_1(j\Delta t) \quad . \qquad (2.2.35)$$

Now, consider weighted average at $x = L$ give as

$$-r\theta u_{N-1,j+1} + (1 + 2r\theta)u_{N,j+1} - r\theta u_{N+1,j+1} = r(1-\theta)u_{N-1,j} + [1 - 2r(1-\theta)]u_{N,j} + r(1-\theta)u_{N+1,j} \quad . \qquad (2.2.36)$$

Apply central difference for right boundary condition at $x = L$ as given in (2.2.29). Substitute into (2.2.1) we obtain an approximation of the Neumann condition at $\big((N+1)\Delta x, j\Delta t\big)$ as

$$u_{N+1,j+1} = u_{N-1,j+1} + 2\Delta x g_2((j+1)\Delta t)\ , u_{i+1,j} = u_{i-1,j} + 2\Delta x g_2((j\Delta t) \quad (2.2.37)$$

Use Eq(2.2.37) to discretize weighted average method (2.2.36) than gives as

$$-2r\theta u_{N-1,j+1} + (1 + 2r\theta)u_{N,j+1} - 2r\theta\Delta x g_2((j+1)\Delta t) = 2r(1-\theta)u_{N-1,j} + [1 - 2r(1-\theta)]u_{N,j} + 2r(1-\theta)\Delta x g_2((j\Delta t) \qquad (2.2.38)$$

We can write in the matrix tridiagonal form

$$[\mathrm{I} - r\theta C]u^{j+1} = [\mathrm{I} + r(1-\theta)C]u^{j} + 2rf^{n}, \tag{2.2.39}$$

where

$$u = [u_0, u_1, \dots \dots \dots \dots \dots, u_{N-1}, u_N]^T,$$

and

$$C = \begin{bmatrix} -2 & 2 & . & . & . & . & . \\ 1 & -2 & 1 & & & & . \\ . & & . & & & & . \\ . & & & . & & & . \\ . & & & & . & & . \\ . & & & & 1 & -2 & 1 \\ . & . & . & . & . & 2 & -2 \end{bmatrix}, \quad f^{n} =$$

$$\begin{bmatrix} -(1-\theta)\Delta x g_1(j\Delta t) - \theta\Delta x g_1((j+1)\Delta t) \\ 0 \\ . \\ . \\ . \\ . \\ . \\ . \\ (1-\theta)\Delta x g_2(j\Delta t) + \theta\Delta x g_2((j+1)\Delta t) \end{bmatrix}$$

Chapter 3

LOCAL TRUNCTION ERROR, CONSISTENCY AND STABILITY OF DIFFERENCE SCHEMES

3.1 Local Truncation Error

Local truncation error represents the difference between an exact differential equation and its finite difference representation at a point in space and time. Local truncation error provides a basis for comparing local accuracies of various difference schemes. In particular, if the partial differential equation satisfied by the exact solution U is written $F(U)$ and if $F(u)$ is the equation satisfied by the discrete approximation u then truncation error at the (i, j)th mesh point is $T_{i,j} = F_{i,j}(U)$ [4].

3.2 Local Truncation Error for Diffusion Equation

We analyze the local truncation error for diffusion equation,

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} \tag{3.2.1}$$

at the mesh point (i, j) for three classical schemes and Weighted Average scheme as follows.

3.2.1 Local Truncation Error for Explicit Method (FTCS)

$$F_{i,j}(u) = \frac{u_{i,j+1} - u_{i,j}}{\Delta t} - \frac{u_{i+1,j} - 2u_{i,j} + u_{i_1,j}}{(\Delta x)^2} \quad , \tag{3.2.2}$$

substituting U for u we obtain

$$T_{i,j} = F_{i,j}(U) = \frac{U_{i,j+1} - U_{i,j}}{\Delta t} - \frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{(\Delta x)^2} \tag{3.2.3}$$

Use Taylor's expansion for $U_{i+1,j}, U_{i-1,j}$ and $U_{i,j+1}$,we have the following.

$$U_{i+1,j} = U_{i,j} + \Delta x \left(\frac{\partial U}{\partial x}\right)_{i,j} + \frac{(\Delta x)^2}{2}\left(\frac{\partial^2 U}{\partial x^2}\right)_{i,j} + \frac{(\Delta x)^3}{6}\left(\frac{\partial^3 U}{\partial x^3}\right)_{i,j} + \frac{(\Delta x)^4}{24}\left(\frac{\partial^4 U}{\partial x^4}\right)_{i,j} + \cdots \qquad (3.2.4)$$

$$U_{i-1,j} = U_{i,j} - \Delta x \left(\frac{\partial U}{\partial x}\right)_{i,j} + \frac{(\Delta x)^2}{2}\left(\frac{\partial^2 U}{\partial x^2}\right)_{i,j} - \frac{(\Delta x)^3}{6}\left(\frac{\partial^3 U}{\partial x^3}\right)_{i,j} + \frac{(\Delta x)^4}{24}\left(\frac{\partial^4 U}{\partial x^4}\right)_{i,j} + \ldots \qquad (3.2.5)$$

$$U_{i,j+1} = U_{i,j} + \Delta t \left(\frac{\partial U}{\partial t}\right)_{i,j} + \frac{(\Delta t)^2}{2}\left(\frac{\partial^2 U}{\partial t^2}\right)_{i,j} + \frac{(\Delta t)^3}{6}\left(\frac{\partial^3 U}{\partial t^3}\right)_{i,j} + \frac{(\Delta t)^4}{24}\left(\frac{\partial^4 U}{\partial t^4}\right)_{i,j} + \cdots \quad . \qquad (3.2.6)$$

Substituting equations $(3.2.4 - 3.2.6)$ in equation $(3.2.3)$ then give

$$T_{i,j} = F_{i,j}(U) = \left(\frac{\partial U}{\partial t} - \frac{\partial^2 U}{\partial x^2}\right)_{i,j} + \frac{1}{2}\Delta t \left(\frac{\partial^2 U}{\partial t}\right)_{i,j} - \frac{1}{12}(\Delta x^2)\left(\frac{\partial^4 U}{\partial x^4}\right)_{i,j} + O((\Delta t)^2) + O((\Delta x)^4), \qquad (3.2.7)$$

where $U(x_{i,}t_j)$ is the solution of the differential equation.

$$\left(\frac{\partial U}{\partial t} - \frac{\partial^2 U}{\partial x^2}\right)_{i,j} = 0 \qquad (3.2.8)$$

Therefore the main part of the local truncation error is

$$\frac{1}{2}\Delta t \left(\frac{\partial^2 U}{\partial t^2}\right)_{i,j} - \frac{1}{12}(\Delta x)^2\left(\frac{\partial^4 U}{\partial x^4}\right)_{i,j} \qquad (3.2.9)$$

Hence

$$T_{i,j} = O(\Delta t) + O((\Delta x)^2) \ . \tag{3.2.10}$$

Thus the explicit solution to equation (3.2.1) is $O(\Delta t)$ accurate in time and $O((\Delta x))^2$ accurate in space.

3.2.2 Local Truncation Error for Implicit Method (FTCS)

$$F_{i,j}(u) = \frac{u_{i,j+1} - u_{i,j}}{\Delta t} - \frac{u_{i-1,j+1} - 2u_{i,j+1} + u_{i+1,j+1}}{(\Delta x)^2} \ , \tag{3.2.11}$$

substituting U for u we obtain

$$T_{i,j} = F_{i,j}(u) = \frac{U_{i,j+1} - U_{i,j}}{\Delta t} - \frac{U_{i-1,j+1} - 2U_{i,j+1} + U_{i+1,j+1}}{(\Delta x)^2} \, . \tag{3.2.12}$$

Use Taylor's expansion for $U_{i-1,j+1}, U_{i+1,j+1}$, we have the following

$$\begin{aligned}
U_{i+1,j+1} = U_{i,j} &+ \Delta x \left(\frac{\partial U}{\partial x}\right)_{i,j} + \Delta t \left(\frac{\partial U}{\partial t}\right)_{i,j} + \frac{(\Delta x)^2}{2}\left(\frac{\partial^2 U}{\partial x^2}\right)_{i,j} + \frac{(\Delta t)^2}{2}\left(\frac{\partial^2 U}{\partial t^2}\right)_{i,j} \\
&\Delta x \Delta t \left(\frac{\partial^2 U}{\partial x \partial t}\right)_{i,j} + \frac{(\Delta x)^3}{6}\left(\frac{\partial^3 U}{\partial x^3}\right)_{i,j} + \frac{(\Delta t)^3}{6}\left(\frac{\partial^3 U}{\partial t^3}\right)_{i,j} + \frac{(\Delta x)^2}{2}\Delta t \left(\frac{\partial^3 U}{\partial x^2 \partial t}\right)_{i,j} \\
&+ \Delta x \frac{(\Delta t)^2}{2}\left(\frac{\partial^3 U}{\partial x \partial t^2}\right)_{i,j} + \frac{(\Delta x)^4}{24}\left(\frac{\partial^4 U}{\partial x^4}\right)_{i,j} + \frac{(\Delta t)^4}{24}\left(\frac{\partial^4 U}{\partial t^4}\right)_{i,j} \\
&+ \frac{(\Delta x)^2(\Delta t)^2}{4}\left(\frac{\partial^4 U}{\partial x^2 \partial t^2}\right)_{i,j} + \frac{(\Delta x)^3}{6}\Delta t \left(\frac{\partial^4 U}{\partial x^3 \partial t}\right)_{i,j} \\
&+ \Delta x \frac{(\Delta t)^3}{6}\left(\frac{\partial^4 U}{\partial x \partial t^3}\right)_{i,j} + \cdots \qquad .
\end{aligned} \tag{3.2.13}$$

$$U_{i-1,j+1} = U_{i,j} - \Delta x\left(\frac{\partial U}{\partial x}\right)_{i,j} + \Delta t\left(\frac{\partial U}{\partial t}\right)_{i,j} + \frac{(\Delta x)^2}{2}\left(\frac{\partial^2 U}{\partial x^2}\right)_{i,j} + \frac{(\Delta t)^2}{2}\left(\frac{\partial^2 U}{\partial t^2}\right)_{i,j}$$

$$-\Delta x\Delta t\left(\frac{\partial^2 U}{\partial x\partial t}\right)_{i,j} - \frac{(\Delta x)^3}{6}\left(\frac{\partial^3 U}{\partial x^3}\right)_{i,j} + \frac{(\Delta t)^3}{6}\left(\frac{\partial^3 U}{\partial t^3}\right)_{i,j} + \frac{(\Delta x)^2}{2}\Delta t\left(\frac{\partial^3 U}{\partial x^2\partial t}\right)_{i,j}$$

$$-\Delta x\frac{(\Delta t)^2}{2}\left(\frac{\partial^3 U}{\partial x\partial t^2}\right)_{i,j} + \frac{(\Delta x)^4}{24}\left(\frac{\partial^4 U}{\partial x^4}\right)_{i,j} + \frac{(\Delta t)^4}{24}\left(\frac{\partial^4 U}{\partial t^4}\right)_{i,j}$$

$$+\frac{(\Delta x)^2(\Delta t)^2}{4}\left(\frac{\partial^4 U}{\partial x^2\partial t^2}\right)_{i,j} - \frac{(\Delta x)^3}{6}\Delta t\left(\frac{\partial^4 U}{\partial x^3\partial t}\right)_{i,j}$$

$$-\Delta x\frac{(\Delta t)^3}{6}\left(\frac{\partial^4 U}{\partial x\partial t^3}\right)_{i,j} + \cdots \quad . \tag{3.2.14}$$

Substituting equations (3.2.13), (3.2.14) and (3.2.6) in (3.2.12) then gives.

$$T_{i,j} = \left(\frac{\partial U}{\partial t} - \frac{\partial^2 U}{\partial x^2}\right)_{i,j} + \frac{1}{2}\Delta t\left(\frac{\partial^2 U}{\partial t^2}\right)_{i,j} - \frac{1}{12}\Delta x^2\left(\frac{\partial^4 U}{\partial x^4}\right)_{i,j}$$

$$+O((\Delta t)^2) + O((\Delta x)^4) \quad , \tag{3.2.15}$$

where U is the solution of the differential equation.

$$\left(\frac{\partial U}{\partial t} - \frac{\partial^2 U}{\partial x^2}\right)_{i,j} = 0 \ . \tag{3.2.16}$$

From equation (3.2.15) the principal part of the local truncation error for implicit scheme is

$$\frac{1}{2}\Delta t\left(\frac{\partial^2 U}{\partial t^2}\right)_{i,j} - \frac{1}{12}(\Delta x)^2\left(\frac{\partial^4 U}{\partial x^4}\right)_{i,j} \quad . \tag{3.2.17}$$

Hence

$$T_{i,j} = O(\Delta t) + O((\Delta x)^2) \qquad (3.2.18)$$

Thus the implicit solution to equation (3.2.1) is $O(\Delta t)$ accurate in time and $O((\Delta x)^2)$accurate in space.

3.2.3 Local Truncation Error for Crank Nicolson

Consider the crank Nicolson method

$$F_{i,j}(u) = \frac{u_{i,j+1} - u_{i,j}}{\Delta t} - \frac{1}{2}[\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + \frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{(\Delta x)^2}], \qquad (3.2.19)$$

substituting U for u we obtain

$$T_{i,j} = F_{i,j}(U) = \frac{U_{i,j+1} - U_{i,j}}{\Delta t} - \frac{1}{2}[\frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{(\Delta x)^2} + \frac{U_{i+1,j+1} - 2U_{i,j+1} + U_{i-1,j+1}}{(\Delta x)^2}]. \qquad (3.2.20)$$

Substituting equation$(3.2.4 - 3.2.6)$, $(3.2.13)$and $(3.2.14)$ in $(3.2.20)$ then gives

$$T_{i,j} = \left(\frac{\partial U}{\partial t} - \frac{\partial^2 U}{\partial x^2}\right)_{i,j} + \frac{\Delta t}{2}\frac{\partial}{\partial t}\left(\frac{\partial U}{\partial t} - \frac{\partial^2 U}{\partial x^2}\right)_{i,j} + \frac{(\Delta t)^2}{6}\left(\frac{\partial^3 U}{\partial t^3}\right)_{i,j} - \frac{(\Delta x)^2}{12}\left(\frac{\partial^4 U}{\partial x^4}\right)_{i,j} + O((\Delta t)^3) + O((\Delta x)^3) \quad , \qquad (3.2.21)$$

where U is the solution of the differential equation.

$$\left(\frac{\partial U}{\partial t} - \frac{\partial^2 U}{\partial x^2}\right)_{i,j} = 0 \ . \qquad (3.2.22)$$

From equation (3.2.21) the principal part of the local truncation error for Crank-Nicolson scheme is

$$\frac{(\Delta t)^2}{6}\left(\frac{\partial^3 U}{\partial t^3}\right)_{i,j} - \frac{(\Delta x)^2}{12}\left(\frac{\partial^4 U}{\partial x^4}\right)_{i,j} \qquad (3.2.23)$$

Hence

$$T_{i,j} = O((\Delta t)^2) + O((\Delta x)^2). \qquad (3.2.24)$$

Thus the Crank-Nicolson solution to equation (3.2.1) is $O((\Delta x)^2\ )$ accurate in space and $O((\Delta t)^2)$ accurate in time.

3.2.4 Local Truncation Error for Weighted Average

$$F_{i,j}(u) = \frac{u_{i,j+1} - u_{i.j}}{\Delta t} - \frac{1}{(\Delta x)^2}\ \ [\theta\big(u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}\big)$$
$$+(1-\theta)(u_{i+1,j} - 2u_{i,j} + u_{i-1,j})]\ , \qquad (3.2.25)$$

substituting U for u we obtain

$$T_{i,j} = F_{i,j}(U) = \frac{U_{i,j+1} - U_{i.j}}{\Delta t} - \frac{1}{(\Delta x)^2}\ \ [\theta\big(U_{i+1,j+1} - 2U_{i,j+1} + U_{i-1,j+1}\big)$$
$$+(1-\theta)(U_{i+1,j} - 2U_{i,j} + U_{i-1,j})]\ \ . \qquad (3.2.26)$$

Substituting equation $(3.2.4 - 3.2.6), (3.2.13)$ and $(3.2.14)$ in $(3.2.26)$ than gives

$$T_{i,j} = \left(\frac{\partial U}{\partial t} - \frac{\partial^2 U}{\partial x^2}\right)_{i,j} + \Delta t\,\frac{\partial}{\partial t}\left(\frac{1}{2}\frac{\partial U}{\partial t} - \theta\,\frac{\partial^2 U}{\partial x^2}\right)_{i,j} - \frac{(\Delta x)^2}{12}\left(\frac{\partial^4 U}{\partial x^4}\right)_{i,j}$$
$$+\frac{(\Delta t)^2}{6}\left(\frac{\partial^3 U}{\partial t^3}\right)_{i,j} + \frac{(\Delta t)^3}{24}\left(\frac{\partial^4 U}{\partial t^4}\right)_{i,j} - \theta\,\frac{(\Delta t)^2}{2}\left(\frac{\partial^4 U}{\partial x^2 \partial t^2}\right)_{i,j}\ \ , \qquad (3.2.27)$$

Where U is the solution of differential equation

$$\left(\frac{\partial U}{\partial t}-\frac{\partial^2 U}{\partial x^2}\right)_{i,j}=0 \quad . \tag{3.2.28}$$

If $\theta=\frac{1}{2}$ the equation (3.2.27) gives us Crank Nicolson scheme, which is second order accurate in both Δt and Δx. Another choice to $\theta=0,1$ gives us $O(\Delta t)$ accurate in time and $O((\Delta x)^2)$accurate in space.

3.3 Consistency

The notion of consistency addresses the problem of whether the finite difference approximation is really representing the partial differential equation. We say that a finite difference approximation is consistent with a differential equation if the finite difference equations converge to the original equation as the time and space grids are refined. Hence, if the truncation error goes to zero as time and space grids are refined we conclude that the scheme is consistent [4].

3.3.1 Consistency of Explicit Method

For the explicit solution to the diffusion equation, the truncation error is,

$$\begin{aligned} T_{i,j}=F_{i,j}(U)=\left(\frac{\partial U}{\partial t}+\frac{\partial^2 U}{\partial x^2}\right)_{i,j}+\frac{1}{2}\Delta t\left(\frac{\partial^2 U}{\partial t}\right)_{i,j} \\ -\frac{1}{12}(\Delta x)^2\left(\frac{\partial^4 U}{\partial x^4}\right)_{i,j}+O((\Delta t)^2)+O((\Delta x)^4) \end{aligned} \tag{3.3.1}$$

Thus as $\Delta x\to 0$ and $\Delta t\to 0$ then $T_{i,j}=0$, hence the explicit method is consistent with partial differential equation (3.2.1).

3.3.2 Consistency of Implicit Method

For the implicit solution to the diffusion equation, the truncation error is,

$$T_{i,j} = \left(\frac{\partial U}{\partial t} - \frac{\partial^2 U}{\partial x^2}\right)_{i,j} + \frac{1}{2}\Delta t\left(\frac{\partial^2 U}{\partial t^2}\right)_{i,j} - \frac{1}{12}\Delta x^2\left(\frac{\partial^4 U}{\partial x^4}\right)_{i,j}$$

$$+O((\Delta t)^2) + O((\Delta x)^4) \qquad (3.3.2)$$

Thus as $\Delta x \to 0$ and $\Delta t \to 0$ then $T_{i,j} = 0$, hence the implicit method is consistent with partial differential equation $(3.2.1)$

3.3.3 Consistency of Crank Nicolson Method

For the Crank Nicolson solution to the diffusion equation, the truncation error is,

$$T_{i,j} = \left(\frac{\partial U}{\partial t} - \frac{\partial^2 U}{\partial x^2}\right)_{i,j} + \frac{\Delta t}{2}\frac{\partial}{\partial t}\left(\frac{\partial U}{\partial t} - \frac{\partial^2 U}{\partial x^2}\right)_{i,j} + \frac{\Delta t^2}{6}\left(\frac{\partial^3 U}{\partial t^3}\right)_{i,j}$$

$$-\frac{\Delta x^2}{12}\left(\frac{\partial^4 U}{\partial x^4}\right)_{i,j} + O(\Delta t)^3) + O((\Delta x)^3) \qquad (3.3.3)$$

Thus as $\Delta x \to 0$ and $\Delta t \to 0$ then $T_{i,j} = 0$, hence the Crank Nicolson method is consistent with partial differential equation $(3.2.1)$.

3.4 Stability and Convergence of Finite Difference Schemes

3.4.1 Stability and Convergence

The stability of a numerical scheme is associated with propagation of numerical error. A finite difference scheme is stable if the error stays constant or decrees as the iterative process goes on. On contrary, if the error grows with time, the scheme is said to be unstable

Definition 3.4.1.1 [4]

A finite difference scheme is stable if the scheme do not allows the growth of error in the solution with different time level.

A numerical scheme is convergent if the computed solution of the discretized equation leads to the exact solution of the differential equation as the time and grid spacing lead to zero.

This will have definition as shown below. The computed solution $u_{i,j}$ must approach the exact solution U of the differential equation at any point $x_i = i\Delta x$ and $t_j = j\Delta t$ when Δx and Δt lead to zero while keeping x_i and t_j constant. In other hand, the error

$$\varepsilon_{i,j} = u_{i,j} - U_{i,j} \tag{3.4.1}$$

Satisfying the following convergence condition

$$\lim_{\Delta t, \Delta x \to 0} \left|\varepsilon_{i,j}\right| \to 0 \text{ at fixed } x_i = i\Delta x \text{ and } t_j = j\Delta t \tag{3.4.2}$$

Theorem 3.4.1.1 (Lax theorem) [4]

For a well-posed initial and boundary value problem, if a finite difference scheme is consistent with the partial differential equation, then the stability is the necessary and sufficient condition for convergence that is

Consistency + stability ↔ convergence

3.4.2 Von Neumann Stability Analysis

There are many approaches to analyze whether a finite difference scheme is stable or unstable. In this thesis, we will consider the Von Neumann stability analysis for presented finite difference schemes.

The Von Neumann stability analysis is most commonly used, but it is restricted to linear initial value problems with constant coefficients. For more sophisticated problems including variable coefficients, nonlinearities and complicated boundary conditions, this method is useful to determine necessary conditions for stability. The only class of problems for which Von Neumann analysis provides also sufficient conditions is the class of initial value problems with periodic boundary conditions. The basic idea of this analysis is given by defining the discrete Fourier transform of u as follows [1,3].

The discrete Fourier transform of $u \in \ell_2$ is the function $\tilde{u} \in L_2\,[-\pi, \pi]$ defined by

$$\tilde{u}(\xi) = \frac{1}{\sqrt{2\pi}} \sum_{m=-\infty}^{\infty} e^{-im\xi} u_m \qquad \text{for } \xi \in [-\pi, \pi] \tag{3.4.3}$$

The transform can be inverted by

$$u_2 = \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} e^{-im\xi} \tilde{u}(\xi) d\xi \ , \tag{3.4.4}$$

and then Parselval’s relation is given as given

$$\|\tilde{u}\|_2 = \|u\|_2 \quad . \tag{3.4.5}$$

Consider the difference scheme with discrete Fourier transform and Parselval's identity that gives the inequality as follows.

$$\|u^{n+1}\|_2 \leq Ke^{\beta(n+1)k}\|u^0\|_2 \tag{3.4.6}$$

But since, we can find K and β to satisfy

$$\|\tilde{u}^{n+1}\|_2 \leq Ke^{\beta(n+1)k}\|\tilde{u}^0\|_2 \text{ ,}$$

$$\|\tilde{u}^{n+1}\|_2 \leq \rho(\xi)\|\tilde{u}^0\|_2 \quad , \tag{3.4.7}$$

where $\|\tilde{u}^0\|_2$ is the initial condition. Then the difference scheme is stable in transform space L_2, if

$$\rho(\xi) \leq 1 \quad . \tag{3.4.8}$$

Where $\rho(\xi)$ is the amplification factor for the difference scheme.

Now, we take the discrete Fourier transform without writing all of the summation, let define the operator $f\colon \ell_2 \to L_2\ ([-\pi, \pi])$ as the discrete Fourier transform

$$f(u) = \frac{1}{\sqrt{2\pi}} \sum_{m=-\infty}^{\infty} e^{-im\xi}\, u_m \tag{3.4.9}$$

Where f is linear and preserves the norm. If we define the shift operators as

$$S \pm u = \{v_k\} \qquad \text{where } v_k = v_{k\pm1} \quad , \qquad k = 0, \pm1, \ldots, \tag{3.4.10}$$

then

$$\begin{aligned} f(S \pm u) &= e^{\pm i\xi} f(u) \\ &= e^{\pm i\xi} \tilde{u}(\xi) \ . \end{aligned} \tag{3.4.11}$$

This result will make stability analysis much easier.

3.4.2.1 Stability of Explicit Method

Consider the equation of explicit scheme

$$u_{i,j+1} = ru_{i+1,j} + (1-2r)u_{i,j} + u_{i-1,j} \tag{3.4.12}$$

Apply Von Neumann analysis on(3.4.12) , to get

$$\tilde{u}_{j+1} = re^{i\xi}\tilde{u}_j + (1-2r)\tilde{u}_j + re^{-i\xi}\tilde{u}_j$$

$$= [r\cos\xi + i\sin\xi + r\cos\xi - i\sin\xi + 1 - 2r]\tilde{u}_j$$

$$\tilde{u}_{j+1} = (1 - 2r(1-\cos\xi))\tilde{u}_j$$

$$\tilde{u}_{j+1} = \left(1 - 4r\sin^2\frac{\xi}{2}\right)\tilde{u}_j$$

Then,

$$\tilde{u}_{j+1} = \rho(\xi)\tilde{u}_j \tag{3.4.13}$$

The amplification factor of (3.4.11) is

$$\rho(\xi) = 1 - 4r\sin^2\frac{\xi}{2} \tag{3.4.14}$$

For stability must satisfy $|\rho(\xi)| \leq 1$. That is

$$-1 \leq 1 - 4\sin^2\frac{\xi}{2} \leq 1\ ,$$

$$-2 \leq -4r\sin^2\frac{\xi}{2} \leq 0\ ,$$

$$\frac{1}{2} \geq r\ \sin\frac{\xi}{2} \geq 0\ ,$$

$$0 \leq r\sin^2\frac{\xi}{2} \leq \frac{1}{2}\,.$$

Hence the explicit scheme is conditionally stable and stability criteria is $r \leq \frac{1}{2}$

3.4.2.2 Stability of Implicit Scheme

Consider the equation of implicit scheme.

$$-ru_{i-1,j+1} + (1+2r)u_{i,j+1} - ru_{i+1,j+1} = u_{i,j} \quad . \tag{4.2.15}$$

Apply Von Neumann stability analysis on(4.2.15), therefore.

$$-re^{-i\xi}\tilde{u}_{j+1} + (1+2r)\tilde{u}_{j+1} - re^{i\xi}\tilde{u}_{j+1} = \tilde{u}_j\ ,$$

$$(-r\cos\xi - i\sin\xi + 1 + 2r - r\cos\xi - i\sin\xi)\tilde{u}_{j+1} = \tilde{u}_j\ ,$$

$$[1 + 2r(1-\cos\xi)]\tilde{u}_{j+1} = \tilde{u}_j\ ,$$

$$\left[1 + 4r\ \sin^2\frac{\xi}{2}\right]\tilde{u}_{j+1} = \tilde{u}_j\ ,$$

$$\tilde{u}_{j+1} = \frac{1}{1 + 4r\ \sin^2\frac{\xi}{2}}\tilde{u}_j = \rho(\xi)\tilde{u}_j \quad , \tag{4.2.16}$$

Where amplification factor of (4.2.15) is

$$\rho(\xi) = \frac{1}{1 + 4r\sin^2\frac{\xi}{2}} \ . \qquad (4.2.17)$$

Scheme is stable if $|\rho(\xi)| \leq 1$. That is

$$-1 \leq \frac{1}{1 + 4r\sin^2\frac{\xi}{2}} \leq 1 \qquad (4.2.18)$$

$$-2 \geq 4r\,sin^2\frac{\xi}{2} \geq 0 \ . \qquad (4.2.19)$$

From above inequality (4.2.19) scheme is stable for all positive value of r. that is, implicit scheme is unconditionally stable.

3.4.2.3 Stability of Crank Nicolson Scheme

Consider the equation of Crank Nicolson scheme

$$-ru_{i-1,j+1} + (2+2r)u_{i,j+1} - ru_{i+1,j+1}$$

$$= ru_{i-1,j} + (2-2r)u_{i,j} + ru_{i+1,j} \ . \qquad (4.2.20)$$

Apply Von Neumann analysis on (4.2.20), to achieve

$$-re^{-i\xi}\tilde{u}_{j+1} + (2+2r)\tilde{u}_{j+1} - re^{i\xi}\tilde{u}_{j+1} = re^{-i\xi}\tilde{u}_j + (2-2r)\tilde{u}_j + re^{i\xi}\tilde{u}_j$$

$$(2+2r-2r\cos\xi)\tilde{u}_{j+1} = (2-2r+2r\cos\xi)\tilde{u}_j$$

$$\tilde{u}_{j+1} = \frac{(2-2r+2r\cos\xi)}{(2+2r-2r\cos\xi)}\tilde{u}_j$$

$$\tilde{u}_{j+1} = \left(\frac{1-4r\sin^2\frac{\xi}{2}}{1+4r\sin^2\frac{\xi}{2}}\right) = \rho(\xi)\tilde{u}_j \ . \qquad (4.2.21)$$

The amplification factor of (4.2.20) is

$$\rho(\xi)=\left(\frac{1-4r\sin^2\frac{\xi}{2}}{1+4r\sin^2\frac{\xi}{2}}\right) \quad . \tag{4.2.22}$$

Scheme is stable if $|\rho(\xi)|\leq 1.$ That is

$$-1\leq\frac{1-4r\sin^2\frac{\xi}{2}}{1+4r\sin^2\frac{\xi}{2}}\leq 1 \quad . \tag{4.2.23}$$

From above inequality (4.2.22) scheme is stable for all value of r. Hence Crank Nicolson is unconditionally stable.

3.4.2.4 Stability of Weighted Average Scheme

Consider the equation of weighted average scheme

$$u_{i,j+1}-u_{i.j}=r[\,\theta\big(u_{i+1,j+1}-2u_{i,j+1}+u_{i-1,j+1}\big)$$

$$+(1-\theta)\big(u_{i+1,j}-2u_{i,j}+u_{i-1,j}\big)] \quad . \tag{4.2.24}$$

Apply Von Neumann stability analysis on(4.2.24), therefore

$$(1-2r\theta(\cos\xi-1))\tilde{u}_{j+1}=(1+2r(1-\theta)(\cos\xi-1)\tilde{u}_j$$

$$\tilde{u}_{j+1}=\frac{(1+2r(1-\theta)(\cos\xi-1))\tilde{u}_j}{(1-2r\theta(\cos\xi-1))} \ .$$

Remember that $\cos\xi=1-2sin^2\frac{\xi}{2}$ therefor $\cos\xi-1=-2sin^2\frac{\xi}{2}$ and we obtain

$$\tilde{u}_{j+1}=\left(\frac{1-4r(1-\theta)\sin^2\frac{\xi}{2}}{1+4r\theta\sin^2\frac{\xi}{2}}\right)\tilde{u}_j=\rho(\xi)\tilde{u}_j \tag{4.2.25}$$

The amplification factor of (4.2.24) is

$$\rho(\xi) = \frac{1 - 4r(1-\theta)\sin^2\frac{\xi}{2}}{1 + 4r\theta\,\sin^2\frac{\xi}{2}} \qquad (4.2.26)$$

Scheme is stable if $|\rho(\xi)| \leq 1$. Since $\theta \boldsymbol{\epsilon} [0,1]$ than $4r\theta sin^2\frac{\xi}{2} \geq 0$ we have

$$\rho(\xi) = \frac{1 + 4r\theta sin^2\frac{\xi}{2} - 4r sin^2\frac{\xi}{2}}{1 + 4\theta sin^2\frac{\xi}{2}} \leq 1 \ ,$$

we finally need

$$\frac{1 + 4r\theta sin^2\frac{\xi}{2} - 4r sin^2\frac{\xi}{2}}{1 + 4\theta sin^2\frac{\xi}{2}} \geq -1$$

$$\therefore 1 - 4r(1-\theta)sin^2\frac{\xi}{2} \geq -1 - 4r\theta sin^2\frac{\xi}{2}$$

$$\therefore 1 \geq 2r(1-2\theta)sin^2\frac{\xi}{2}$$

$$\therefore 1 \geq 2r(1-2\theta) \ . \qquad (4.2.27)$$

From above inequality $(4.2.27)$ is satisfied for all positive r if $\theta \geq \frac{1}{2}$ in this case weighted average scheme is unconditionally stable. But if $\theta < \frac{1}{2}$ we require

$$r \leq \frac{1}{2(1-2\theta)}$$

Chapter 4

NUMERICAL RESULTS

In this Chapter we present the numerical results from solving two model problems using finite difference schemes described in Chapter 2. In our computations we used various values of $r = 0.4\,, 0.5\,, 1$ with fixed $\Delta x = 0.05$. In order to check accuracy of u using discussed finite difference schemes the following error calculation is used;

$$\varepsilon = \left\| U_{ij} - u_{ij} \right\|_{\infty} \ ,$$

where u_{ij} is the solution calculate by the numerical methods at the node i in the jth time level and U_{ij} is the exact solution at the node i in the jth time level.

Problem 1 (Dirichlet type of boundary condition)

$$u_t = u_{xx} \qquad 0 < x < 1 \quad , \quad 0 < t \leq 1$$

with initial condition

$$u(x, 0) = \sin \pi x \qquad 0 \leq x \leq 1$$

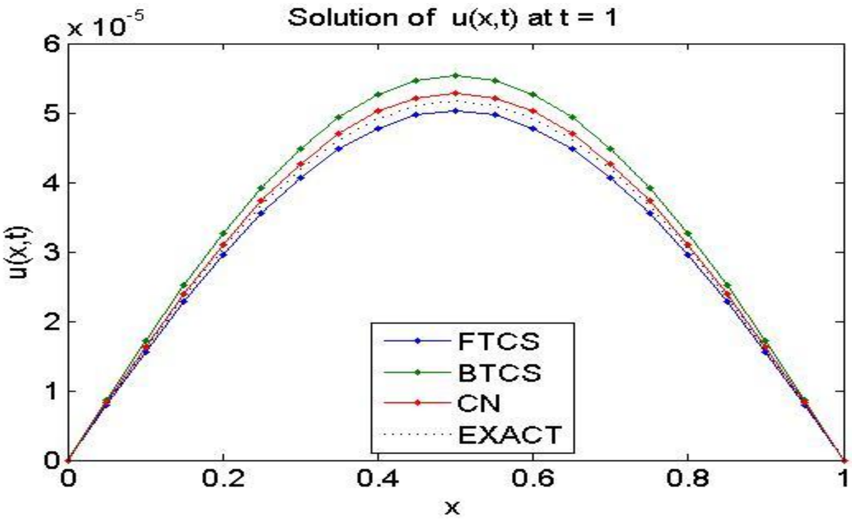
and boundary conditions

$$u(0, t) = 0 \qquad 0 < t \leq 1$$

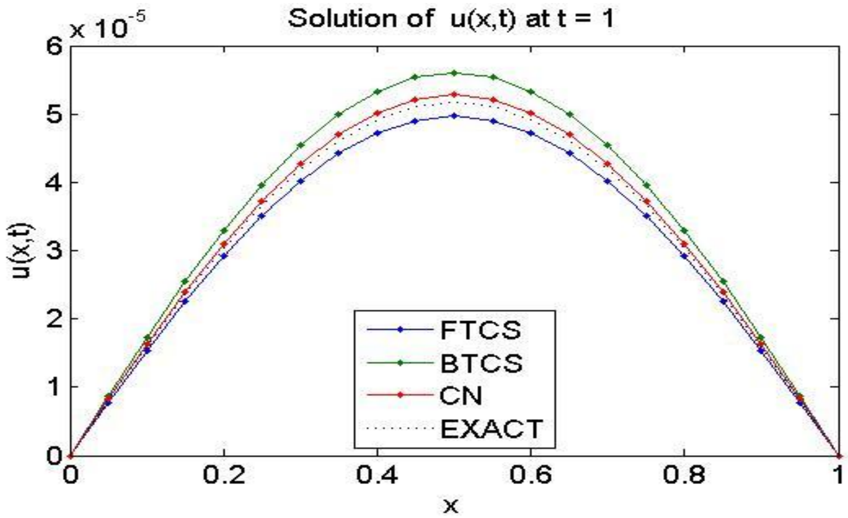
$$u(1, t) = 0 \qquad 0 < t \leq 1$$

where

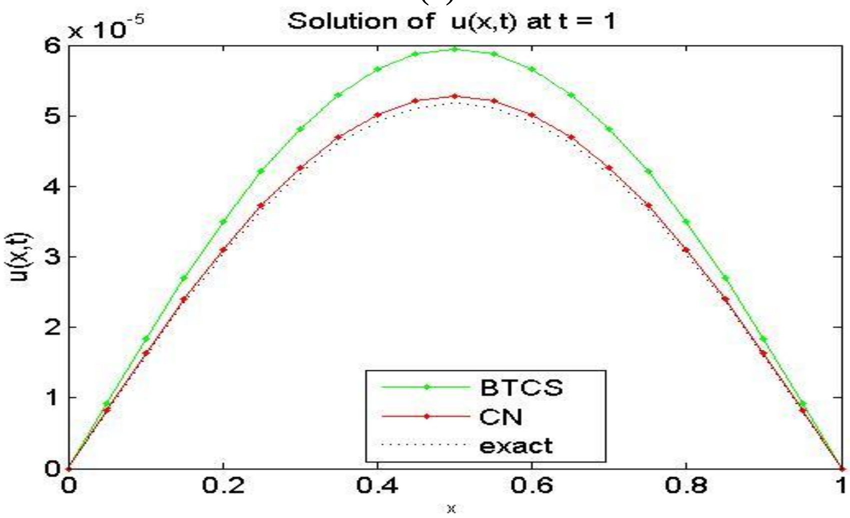
$$u_{exact} = u(x, y) = e^{-\pi^2 t} \sin \pi x$$



(a)

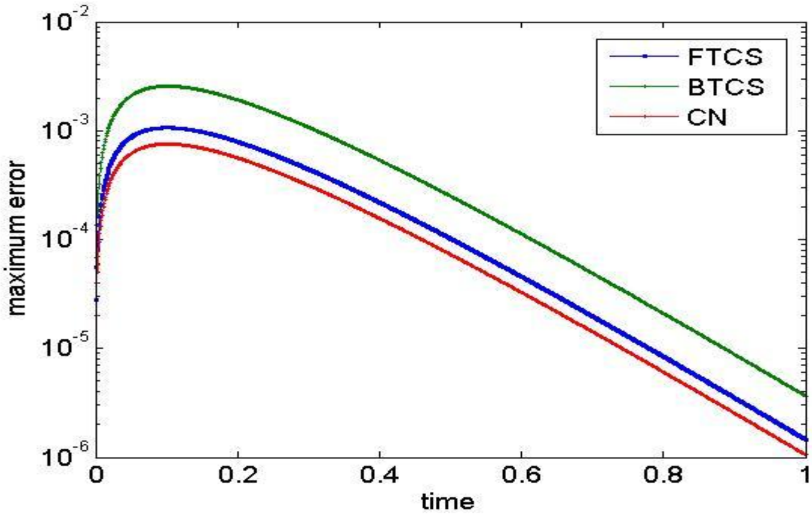


(b)

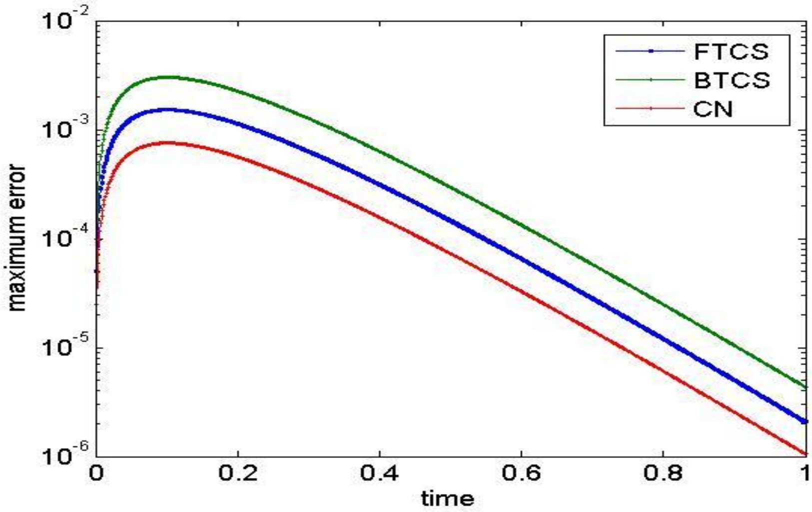


(c)

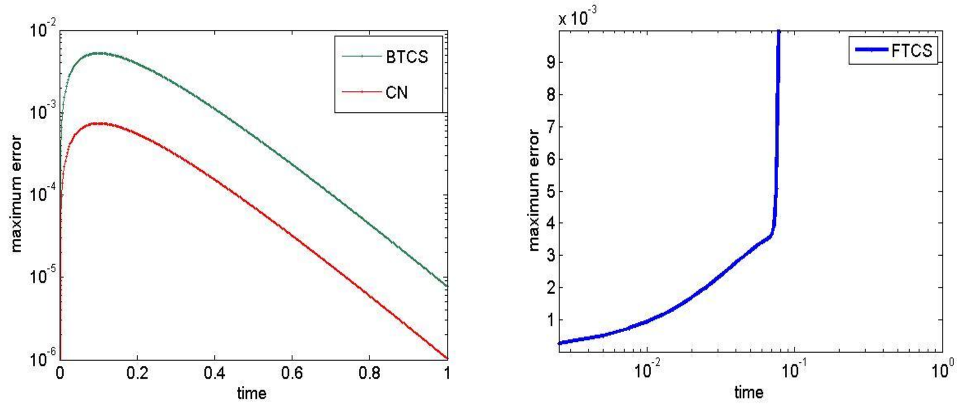
Figure 4.1: Exact and approximate solution of $u(x,t)$ using three different numerical schemes of time level $t = 1$ with different values of
(a) $r = 0.4$ (b) $r = 0.5$ (c) $r = 1$



(a)



(b)



(c)

Figure 4.2: Maximum error vs. time for three different schemes with (a) $r = 0.4$ (b) $r = 0.5$ (c) $r = 1$ respectively

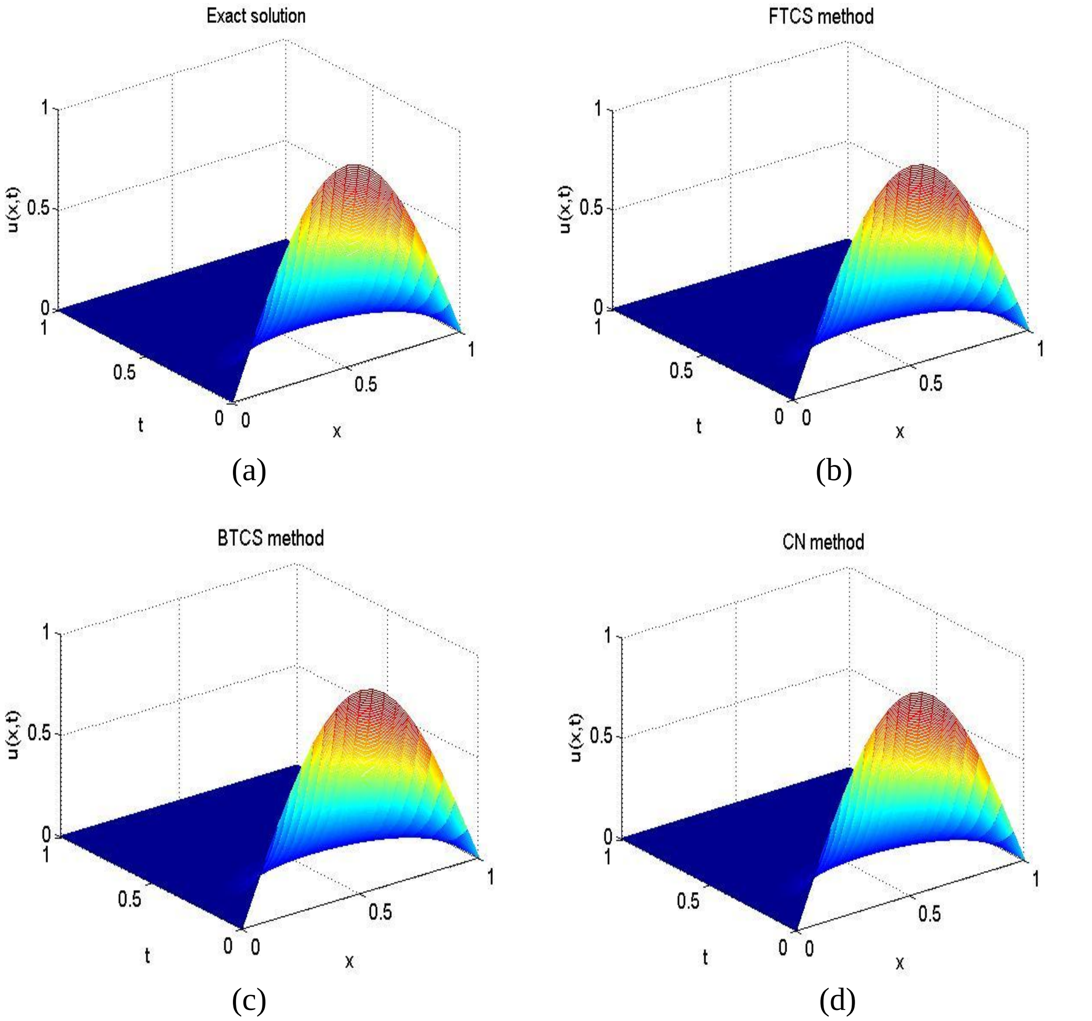


Figure 4.3: Exact and numerical solution of three different schemes with $r = 0.5$ and $\Delta x = 0.05$

Problem 2 (Neumann type of boundary conditions)

$$u_t = u_{xx} \qquad 0 < x < 1 \quad , \quad 0 < t < 1$$

with initial condition

$$u(x,0) = \cos \pi x \qquad 0 \le x \le 1$$

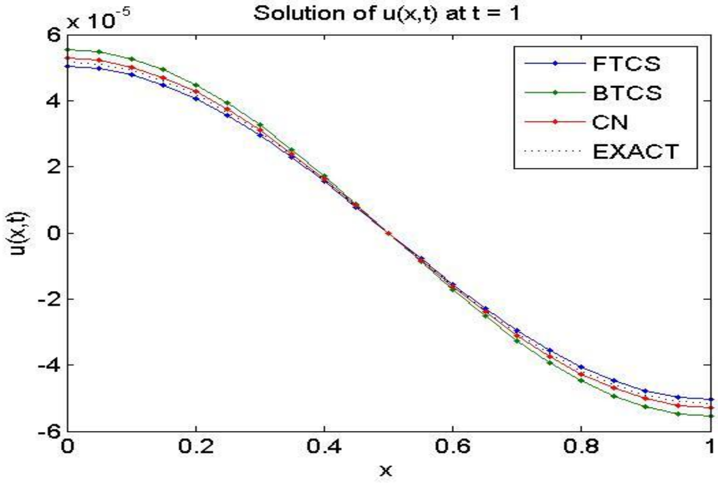
and boundary conditions

$$u_x(0,t) = 0 \qquad 0 < t \le 1$$

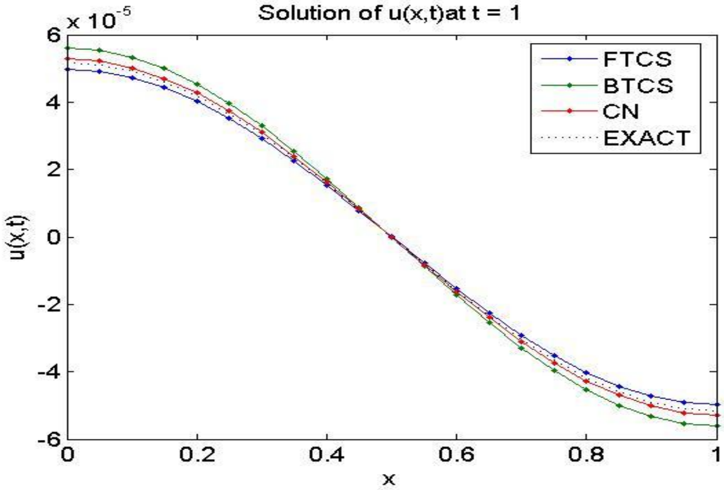
$$u_x(1,t) = 0 \qquad 0 < t \le 1$$

where

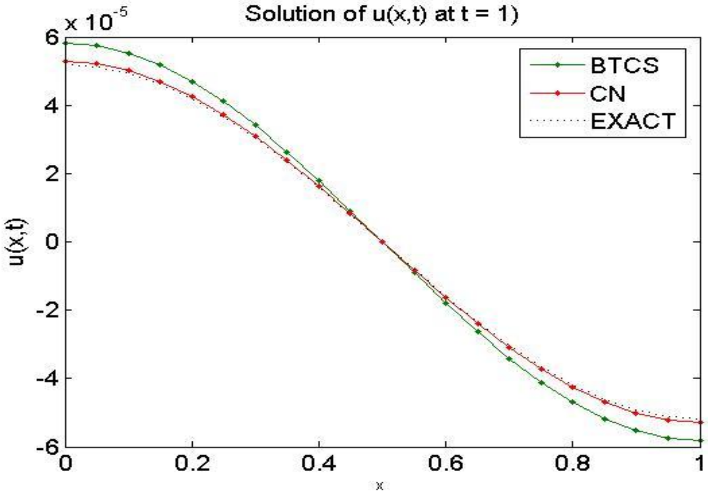
$$u_{exact} = u(x,y = e^{-\pi^2 t} \cos \pi x$$



(a)



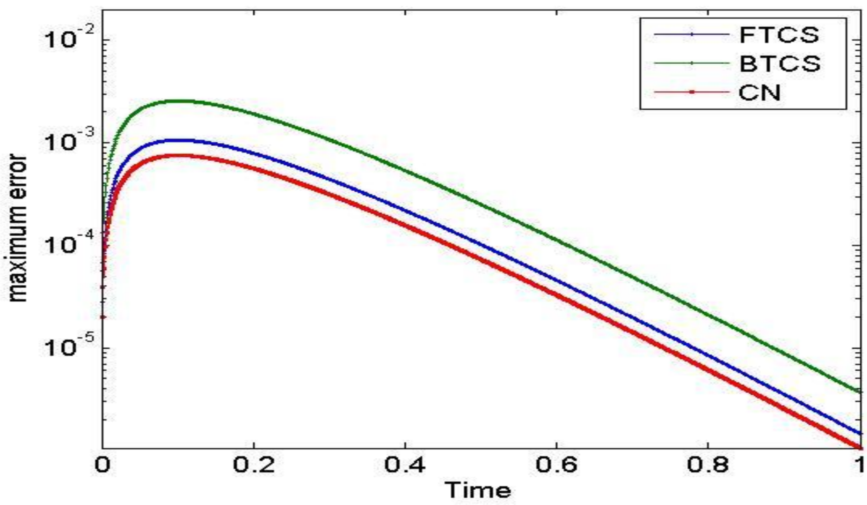
(b)



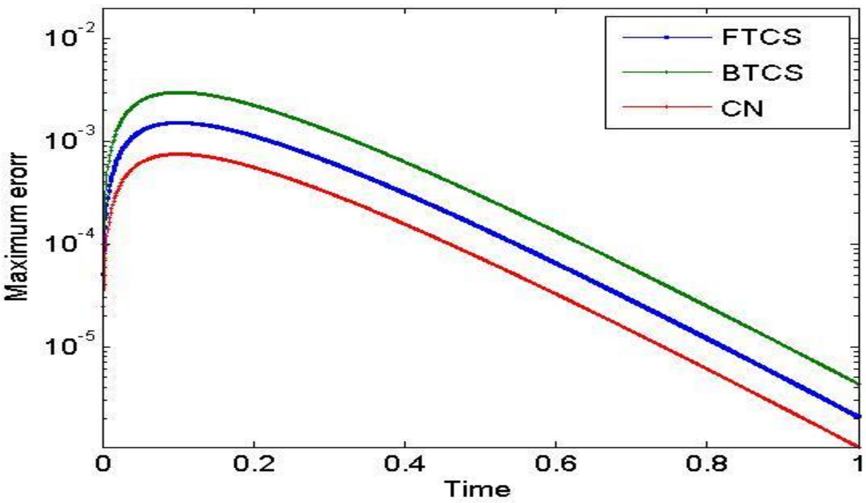
(c)

Figure 4.4: Exact and approximate solution of $u(x,t)$ using three different numerical schemes of time level $t = 1$ with different values of

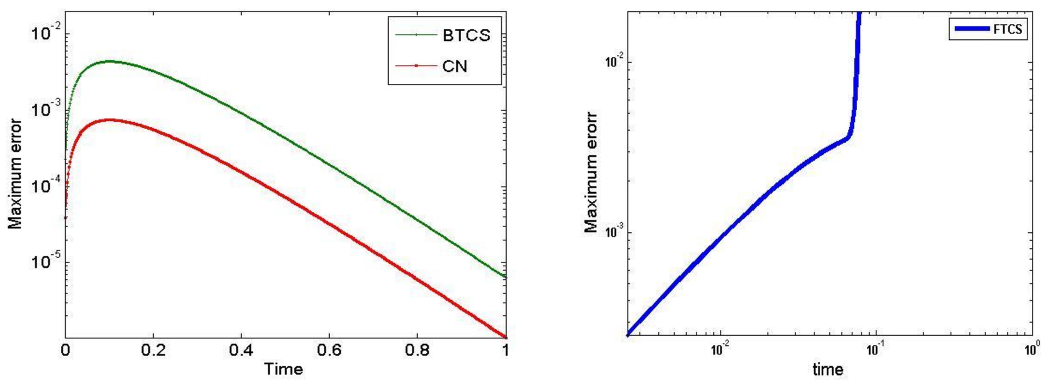
(a) $r = 0.4$ (b) $r = 0.5$ (c) $r = 1$



(a)



(b)



(c)

Figure 4.5: Maximum error vs. time for three different schemes with (a) $r = 0.4$ (b) $r = 0.5$ (c) $r = 1$ respectively

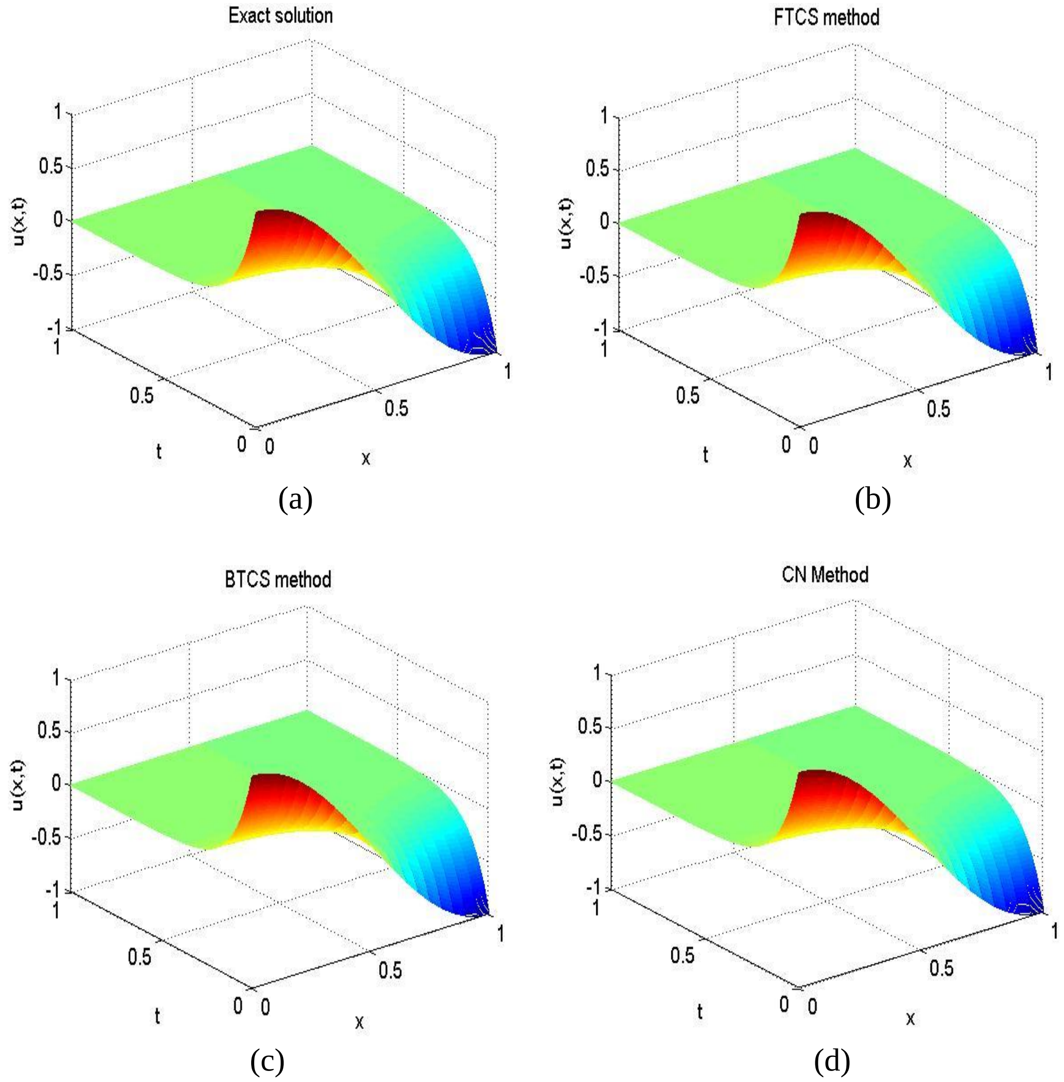


Figure 4.6: Exact and numerical solution of three different schemes with $r = 0.5$ and $\Delta x = 0.05$

Based on the considered comparison factors to evaluate the performance of the three finite difference schemes according to the stability criteria, we observed from the numerical results that these schemes work well and each scheme produced reasonable results for problem 1 and problem 2. Figures $(4.1 \text{ and } 4.4)$ illustrates exact and numerical solution of the three different schemes at time level $t = 1$ and Figure $(4.3 \text{ and } 4.6)$ illustrates exact and approximation solutions of three different schemes for whole domain.

The other factors for comparison worth to consider are the maximum error reduction for each time level. Figures $(4.2 \text{ and } 4.5)$ illustrate the maximum error reduction to solve problem 1 and problem 2 respectively. Using $r = 0.4\,, 0.5\,, 1$.we observed from both Figures that Crank Nicolson scheme is most accurate than other schemes.

Chapter 5

CONCLUSION

In this thesis, FTCS, BTCS and Crank Nicolson scheme were applied to the one dimensional diffusion equation. We observed from numerical computation that these methods worked well according to the stability criteria and each scheme produced reasonable results for evaluating approximation of u. Each of the finite difference methods considered its own advantages and disadvantages. Explicit method is very easy to calculate numerically but has low accuracy must use small Δt and unstable for $r > 0.5$. Implicit and Crank Nicolson methods are unconditionally stable, computer time required at each step is higher. On the other hand Crank Nicolson method is more accurate, and faster than Implicit and Explicit methods according to the order of truncation error.Therefore difficult to judgment of the best scheme according to their own advantages and disadvantages.

REFERENCES

[1] Berntaz, A. R. (2010). *Fourier Series and Numirecal Method for Partial Difference Equation* . New Jersey: Johan Wiley & Sons, Inc.

[2] Causon, D. M., & Mingham, C. G. (2010). *Introductory Finite Difference Methods for PDEs.* Ventus Pablishing Aps.

[3] Smith, G. D. (1985). *Numerical Solution of Partial Differential Equation.* Clarendon press: Oxford .

[4] Thomas, J. W. (1995). *Numerical Partial Differential Equation: Finite Difference Method* . Springer.

[5] Morton, K. W., & Mayers, D. F. (2005). *Numerical solution of partial differential equation.* new york: cambirdge universty.

[6] Noye, J. (1984). *Finite Difference Tachniques for Partial Differntial Equation* . North Holland: B.V.

[7] Steven C. C., & Raymond, P. C. (2010). *Numerical Methods for Engineers.* New York: McGraw.Hill.