An-Najah National University

Faculty of Graduate Studies

Analytical and Numerical Methods for Solving Heat Conduction problems Transient

By

Abdullah Edwan Abdullah Nassar

Supervisor

Prof. Naji Qatanani

This Thesis is Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Mathematics, Faculty of Graduate Studies, An-Najah National University, Nablus-Palestine.

2017

Analytical and Numerical Methods for Solving Heat Conduction problems Transient

By

Abdullah Edwan Abdullah Nassar

This Thesis was Defended Successfully on 15/2/2017 and approved by:

Defense Committee Members	**Signature**
1. Prof. Naji Qatanani /Supervisor	**...…………**
2. Dr. Saed Mallak / External Examiner	**....…………**
3. Dr. Anwar Saleh / Internal Examiner	**……………**

Dedication

I dedicate this work to my family and my friends and all the teachers taught me and I thank my doctors and all those who contributed to stand by my side for the completion of this work, and I thank God for the completion and I wish everyone success in this life and the afterlife more important that God Almighty.

الإقرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

Analytical and Numerical Methods for Solving Transient Heat Conduction problems

أقر بأن ما اشتملت عليه هذه الرسالة إنما هو من نتاج جهدي الخاص بإستثناء ما تمت الإشارة اليه حيثما ورد، وإن هذه الرسالة ككل، أو أي جزء منها لم يقدم لنيل أي درجة أو لقب علمي أو بحثي لدى أية مؤسسة تعليمية أو بحثية أخرى .

Declaration

The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification.

Student's Name: **إسم الطالب:**

Signature: **التوقيع:**

Date: **التاريخ:**

Contents

Analytical and Numerical Methods for Solving Heat Conduction problems Transient

By

Abdullah Edwan Abdullah Nassar

By

Supervisor

Prof. Naji Qatanani

Abstract

The modeling of systems involving heat conduction problems is widely spread among scientists and engineers due to their wide range of applications in science and technology.

In this work, we will present some important analytical and numerical results concerning heat conduction problems and their applications.

First, we will use the Fourier law of heat conduction to derive the composition equation of heat transfer for different regions. The concept of boundary and initial conditions will also be illustrated. The heat conduction problems subject to some boundary and initial conditions for various domains will be solved analytically using the separation of variables, Laplace transforms, Duhamel's and Green's function methods. Numerical approach based on the finite difference method (FDM) has been analyzed and implemented to solve some heat conduction problems. A comparison between the analytical and numerical results have been drawn. Numerical results have shown to be in a close agreement with the exact ones. In fact, the FDM is one of the most efficient numerical methods for solving heat diffusion problems.

Introduction

Many heat conduction problems encountered in engineering applications involve time as independent variable. The effects of heat exchange are subject to constant laws cannot be discovered without the mathematical analysis of heat exchange models. The object of the theory is to demonstrate these laws. Jean Biot (1774-1862) has studied the heat conduction equation but was unsuccessful at dealing with the problem of incorporating external convection effects in heat conduction analysis, see [13]. Joseph Fourier (1768-1830) determined how to solve the problem of Biot's work in 1807 and gave the well-known Fourier's law of cooling. Ernst Schmidt (1892-1975) was a German scientist and pioneer in the field of Engineering thermodynamic, especially in heat and mass transfer, see [6]. He published papers on the now well-known "Graphical Difference Method for Unsteady Heat Conduction" and on "Schieren and Shadow Method" to make thermal boundaries visible and to obtain local heat transfer coefficients. He was the first to measure the velocity and temperature field in a free convection boundary layer and the large heat-transfer coefficients occurring in doplet conduction. In recent years, many researchers have worked on the mathematical analysis of the heat conduction equation (see for example [1, 2, 3, 7, 11, 12]). For heat conduction equation there are two main research areas in the solution of transient problems: One of these areas is the analysis of transient well-posed problems such as direct heat conduction problems for which all required information such as the boundary and initial conditions as well as the coefficients of the transient heat equation and the geometry of the solution domain are given prior to the solution process. The second important topic is concerned with the analysis of the so called inverse problems. These inverse heat conduction problems arise when not all necessary conditions are given, see [12]. In this case, the numerical solution for the temperature and the heat flux must be recovered with the aid of auxiliary measurements inside the domain. It is important to note that inverse heat conduction problems are widely used in the modeling of industrial problems including atmospheric (for example see [15]), and also in the spray cooling for the quenching of iron ingots, see [20]. The goal of the analysis is to determine the variation of the temperature as a function of time and position $T(x,t)$ in one dimension. In general, we deal with conducting bodies in a three-dimensional Euclidean space in a suitable set of coordinates $x \in R^3$ and the goal is to predict the evolution of the temperature field for future times $(t > 0)$. Here we investigate specifically solutions to selected special cases of the transient heat conduction equation:

$$\rho\, C_p \frac{\partial T}{\partial t} = \nabla.(k\nabla T) + g \qquad (*)$$

where $\nabla T = \left(\frac{\partial T}{\partial x}\vec{i} + \frac{\partial T}{\partial y}\vec{j} + \frac{\partial T}{\partial z}\vec{k}\right)$, g is source of strength for a homogeneous heat, C_p is heat capacity or specific heat, $T = T(x,y,z,t)$

is the temperature and ρ is the density. Equation (*) must be solved on different domains, subject to suitable initial and boundary conditions. Solutions to equation (*) involving analytical and numerical methods, see

[14, 8, 9], will be investigated. In chapter one we study the main characteristic features of heat conduction problems and their inherent complexities. The governing partial differential equation of heat conduction with some types of associated boundary conditions will be presented. In chapter two, we present some analytical methods for the transient heat conduction equation (*) on various domains. This involves separation of variables method, Laplace transform method, Duhamel's method and Green's function method. In chapter three, we investigate the numerical handling of the one-dimensional transient heat conduction equation (*) for plane wall, cylinder, and sphere. This can be a achieved by implementing the finite difference method (FDM). The main idea of the FDM is to replace the partial derivatives equation by finite difference approximations. FDMs are thus discretization methods. Some numerical test cases on heat conduction problems have been solved and the numerical and exact results have been compared.

Chapter One

Derivation and Characteristics of Heat Conduction Equation

1.1. Introduction

Heat conduction is one of the three basic modes of thermal energy transport; convection, radiation and conduction. It is involved in virtually all process of $heat\text{–}transfer$ operations. Many routine process-engineering problems can be solved with acceptable accuracy using simple solutions of the heat conduction equation for plane wall, cylindrical, and spherical geometries. This chapter gives an introduction to the macroscopic theory of heat conduction and presents the mechanism of the heat conduction equation and its characteristics.

1.2. Fourier's Law of Heat Conduction

Joseph Fourier developed the mathematical theory of heat conduction early in the nineteenth century. The theory was based on the results of experiments similar to that illustrated in figure(1.1) in which one side of a plane wall solid is held at temperature T_1, while the opposite side is held at a lower temperature T_2.The other four sides are insulated so that heat can flow only in the *x-direction*, it is found that the rate; Q_x, at which heat (thermal energy) is transferred from the hot side to the cold side is proportional to the cross-sectional area A, across which the heat flows; the temperature difference, $T_1 - T_2$; and inversely proportional to the thickness, B, that is:

$$Q_x \propto (A \ (T_1 - T_2))/B \ .$$

Writing this relationship as an equality, we have:

$$Q_x = \frac{Ak\ (T_1 - T_2)}{B} \qquad (1.1)$$

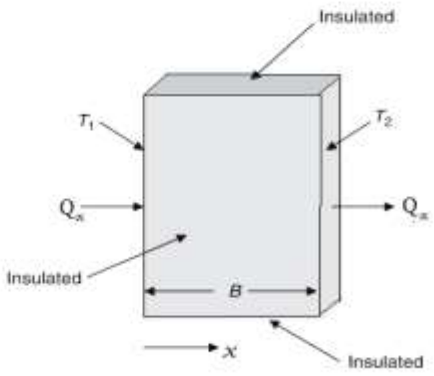

Figure 1.1: one-dimensional heat conduction in a solid.

The constant of proportionality k is called *the thermal conductivity*; it is a property of the material , as such, it is not really a constant, but rather it depends on the nature of material, i.e., on the temperature and pressure of the material, but usually negligible. when the temperature dependence must be taken into account, a linear function is often adequate, particularly for solids, in this case:

$$k = a + bt \qquad (1.2)$$

where a and b are constants. Thermal conductivities of a number of a materials found in many physical References including, see [12]. Methods of estimating thermal conductivities of fluids when data are unavailable can be found in the authoritative book by Polingetal, see [1]. The form of Fourier's law given by equation (1.1) is valid only when the thermal conductivity can be assumed constant, more general result can be obtained by writing the equation for an element of differential thickness. Thus, let the thickness be $\triangle x$ and let $\triangle T \ = T_2 - T_1$, substituting in equation (1.1) gives:

$$Q_x = \frac{(-k\ A.\triangle T)}{\triangle x} \tag{1.3}$$

now in the limit as $\triangle\ x$ approaches to *zero*, $\frac{\triangle T}{\triangle x} \ \longrightarrow \frac{dT}{dx}$,

and equation (1.3) becomes :

$$Q_x \ = -\ k\ A\ \frac{dT}{dx} \tag{1.4}$$

Equation (1.4) is not subject to the restriction of constant k, furthermore, when k is constant, it can be integrated to yield equation (1.1). Hence, equation (1.4) is the general $one\text{–}\ dimensional$ form of *Fourier's law*, the negative sign is necessary because heat flows in the positive *x-direction* when the temperature decreases in the x *–direction*. Thus, according to the standard sign convention that Q_x is positive when the heat flows in the positive *x-direction*, Q_x must be positive when $\frac{dT}{dx}$ is negative. It is often convenient to divide equation (1.4) by the cross–sectional area to give:

$$\bar{Q}_x \ = \frac{Q_x}{A} = \ -k\ \frac{dT}{dx} \tag{1.5}$$

where $\bar{Q}_x$ is *the heat flux*, see [8], equations (1.1), (1.4) and (1.5) are restricted to the situation in which the heat flows in the *x- direction* only. In the general case in which heat flows in all three coordinate directions, the total *heat flux* obtained by adding vector ally the fluxes in the coordinate directions, thus:

$$\vec{\bar{Q}} = \bar{Q}_x \vec{\imath} + \bar{Q}_y \vec{\jmath} + \bar{Q}_z \vec{k} \tag{1.6}$$

where $\vec{\vec{Q}}$ is the heat flux vector and $\vec{i}$, $\vec{j}$, $\vec{k}$ are unit vectors in the x-,y-, z-*directions*, respectively. Each of the component fluxes is given by a one-dimensional Fourier's expression as follows:

$$\bar{Q}_x = -k\frac{\partial T}{\partial x}, \bar{Q}_y = -k\frac{\partial T}{\partial y}, \bar{Q}_z = -k\frac{\partial T}{\partial z} \quad (1.7)$$

partial derivatives are used here since the temperature now varies in all three directions. Substituting above expressions for all fluxes into equation (1.6) gives:

$$\vec{\bar{Q}} = -k\left(\frac{\partial T}{\partial x}\vec{i} + \frac{\partial T}{\partial y}\vec{j} + \frac{\partial T}{\partial z}\vec{k}\right) \quad (1.8)$$

the vector in parenthesis is the temperature gradient vector, and is denoted by $\vec{\nabla}T$. Hence:

$$\overrightarrow{\bar{Q}} = -k\,\vec{\nabla}T \quad (1.9)$$

Fourier's law states that heat flows in the direction of greatest temperature decrease, see [20].

1.3. One-dimensional Heat Conduction Equation

Heat conduction in many geometries shapes, as plane wall, cylinder and sphere can be approximated as being one-dimensional since heat conduction through these geometries is dominant in one direction and negligible in other directions.

1.3.1. Heat Conduction Equation in Plane Wall

Consider a plane wall of thickness$\triangle x$, as shown in figure (1.2), see [4]. the density of the wall ρ, specific heat is C_p, and the area of the wall normal to the direction of heat is A. Therefore, the energy balance during the interval time $\triangle t$ it can be formulated, see [18]:

[*Rate of heat conduction at* x]

-[*Rate of heat conduction at* $x + \triangle x$]

+[*Rate of heat generation inside the element*]

=[*Rate of change of the energy content of the element*]

Or

$$Q_{(x)} - Q_{(x+\triangle x)} + G_{gen\ .element} = \frac{\triangle E.element}{\triangle t} \tag{1.10}$$

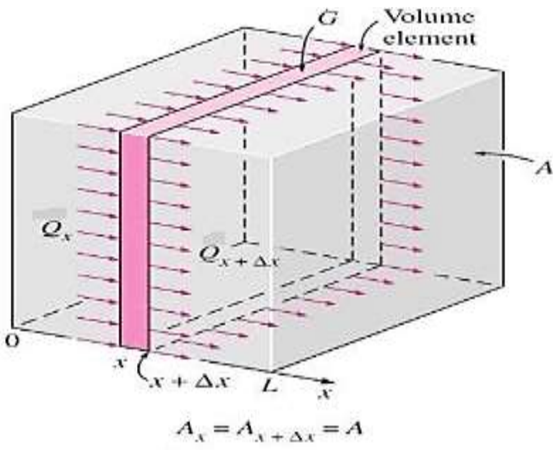

Figure 1.2: one-dimensional heat conduction through a volume element in a large plane wall.

but the change in the energy content of the element and the rate of heat generation with in the element can be expressed as:

$$\Delta E_{element} = E_{(t+\Delta t)} - E_{(t)} = \rho C_p . A . \Delta x (T(t + \Delta t) - T(t)) \qquad (1.11)$$

$$G_{gen.element} = g.(volume\ of\ element) = g.A.\Delta x \qquad (1.12)$$

where g is source of strength for a homogeneous heat, see [17], substituting into equation (1.11), we get:

$$Q_{(x)} - Q_{(x+\Delta x)} + g.A.\Delta x = \rho . C_p . A . \Delta x . \left(\frac{T(t+\Delta t) - T(t)}{\Delta t} \right) \qquad (1.13)$$

dividing by $A \Delta x$, we obtain:

$$\frac{-1}{A} \frac{Q_{(x+\Delta x)} - Q_{(x)}}{\Delta x} + g = \rho . C_p \left(\frac{T(t+\Delta t) - T(t)}{\Delta t} \right) \qquad (1.14)$$

taking the limit as $\Delta x \longrightarrow 0$ and $\Delta t \longrightarrow 0$, the result:

$$\frac{-1}{A} \frac{\partial}{\partial x} (Q_{(x)}) + g = \rho . C_p \frac{\partial T}{\partial t} \qquad (1.15)$$

substituting $Q_{(x)}$ from equation (1.4) in equation (1.15), we get:

$$\frac{1}{A} \frac{\partial}{\partial x} \left(kA \frac{\partial T}{\partial x} \right) + g = \rho C_p \frac{\partial T}{\partial t} \qquad (1.16)$$

Because area A is constant, equation (1.16) becomes:

$$\frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) + g = \rho C_p \frac{\partial T}{\partial t} \qquad (1.17)$$

where the thermal conductivity k is variable, but k in most practical applications is constant, so equation (1.17) becomes:

$$\frac{\partial^2 T}{\partial x^2} + \frac{g}{k} = \frac{\rho C\rho}{k} \frac{\partial T}{\partial t} \qquad (1.18)$$

where $T(x,t)$ is a function of x and t, and $(k/\rho C_p)$ is known as α; *the thermal diffusivity*, see [4], it represents heat quickly spread through the material.

1.3.2. Heat Conduction Equation in Cylinder

Consider a cylinder with thickness Δr as shown in figure (1.3), see [4],

with density ρ, specific heat C_p, and length L, the area of the cylinder is $A = 2\pi\, r\, L$, where r is radius of cylinder, an energy balance of cylinder shell during small time interval Δt, it can be described as:

[*Rate of heat conduction at* r]

-[*Rate of heat conduction at* $r + \Delta r$]

+[*Rate of heat generation inside the element*]

=[*Rate of change of the energy content of the element*]

or

$$Q_{(r)} - Q_{(r+\Delta r)} + G_{gen\,.element} = \frac{\Delta E_{element}}{\Delta t} \tag{1.19}$$

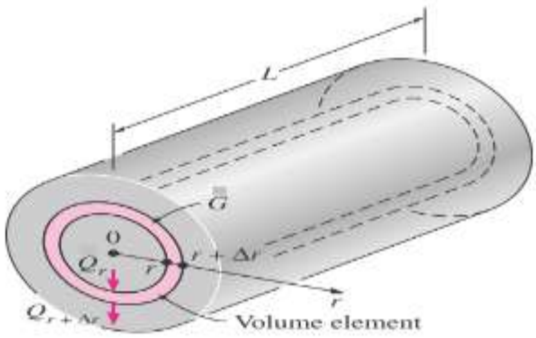

Figure 1.3: one-dimensional heat conduction through a volume element in a long cylinder.

The change in energy content of the element and the rate generation within the element can be expressed as:

$$\triangle E_{element} = E_{(t+\triangle t)} - E_{(t)} = \rho C_p . A . \triangle r (T(t+\triangle t) - T(t)) \quad (1.20)$$

$$G_{generation} = g . (volume) = g . A . \triangle r \quad (1.21)$$

substituting equation (1.21) into equation (1.19), we get:

$$Q_{(r)} - Q_{(r+\triangle r)} + g . A . \triangle r = \rho C_p . A . \triangle r \left(\frac{T(t+\triangle t) - T(t)}{\triangle t} \right) \quad (1.22)$$

now dividing equation (1.22) by $A . \triangle r$, we obtain:

$$\frac{-1}{A} \frac{Q_{(r+\triangle r)} - Q_{(r)}}{\triangle r} + g = \rho C_p \left(\frac{T(t+\triangle t) - T(t)}{\triangle t} \right) \quad (1.23)$$

taking the limit as $\triangle r \longrightarrow 0$ and $\triangle t \longrightarrow 0$, equation (1.23) becomes:

$$\frac{1}{A} \frac{\partial}{\partial r} \left(kA \frac{\partial T}{\partial r} \right) + g = \rho C_p \frac{\partial T}{\partial t} \quad (1.24)$$

where $A = 2\pi r L$, where r is a variable ,so equation (1.24) becomes:

$$\frac{1}{r} \frac{\partial}{\partial r} \left(rk \frac{\partial T}{\partial r} \right) + g = \rho C_p \frac{\partial T}{\partial t} \quad (1.25)$$

where k is a variable. On the other hand if k is a constant then equation (1.25) becomes:

$$\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial T}{\partial r} \right) + \frac{g}{k} = \frac{\rho C_p}{k} \frac{\partial T}{\partial t} \quad (1.26)$$

1.3.3. Heat Conduction Equation in Sphere

Consider a sphere with density ρ, specific heat C_p , and outer radius r and area $A = 4\pi r^2$, where r is the radius of sphere, note the heat transfer area

A depends on radius r, so it varies with location, by considering a thin sphere shell element of thickness Δr and repeating the approach described for the cylinder by using $A = 4\pi r^2$ instead of $A = 2\pi rL$, the one–dimensional transient heat conduction equation for sphere is determined to be figure (1.4), see [18]:

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 k \frac{\partial T}{\partial r}\right) + g = \rho C_p \frac{\partial T}{\partial t} \tag{1.27}$$

where k is variable, on the other hand if k is a constant then equation (1.27) becomes:

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \frac{\partial T}{\partial r}\right) + \frac{g}{k} = \frac{\rho C_p}{k}\frac{\partial T}{\partial t} \tag{1.28}$$

where $\alpha = k/\rho C_p$ is the thermal diffusivity of the material, see [16].

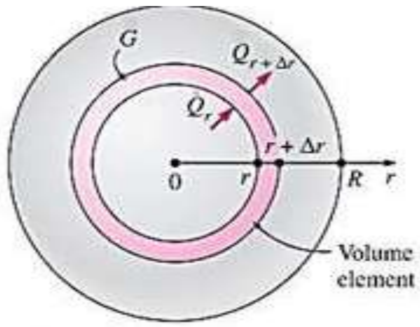

Figure 1.4: one-dimensional heat conduction through a volume element in sphere.

1.3.4. Combined One–dimensional Heat Conduction Equation

All of the one–dimensional transient heat conduction equations for the plane wall, cylinder and sphere can expressed in compact form as:

$$\frac{1}{r^n}\frac{\partial}{\partial r}\left(r^n k \frac{\partial T}{\partial r}\right) + g = \rho C_p \frac{\partial T}{\partial t} \tag{1.29}$$

where $n = 0$ for a plane wall, with change r to $x,\ n = 1$ for a cylinder and $n = 2$ for a sphere, equation (1.29) can be simplified under specified conditions, when k is constant, see [20], these conditions are:

1) **Steady–state**: In the sense that the temperature inside the steel body does not change with time, but vary by location and despite the fact that this assumption is not realistic, but an essential starting point for dealing to simplify matters for the novice, so equation (1.29) becomes:

$$\frac{1}{r^n}\frac{d}{dr}\left(r^n\frac{dT}{dr}\right) + \frac{g}{k} = 0 \qquad (1.30)$$

2) **Transient without heat generation**: There is an emerging energy inside the body and that the thermal energy is transferred through the body of steel from the source only ($g = 0$), equation (1.29) becomes:

$$\frac{1}{r^n}\frac{\partial}{\partial r}\left(r^n\frac{\partial T}{\partial r}\right) = \frac{\rho C_p}{k}\frac{\partial T}{\partial t} \qquad (1.31)$$

3) **Steady–state without heat generation**: In this case $\frac{\partial T}{\partial r} = 0$ and $g = 0$, equation (1.29) becomes:

$$\frac{d}{dr}\left(r^n\frac{dT}{dr}\right) = 0 \quad \text{or} \quad r\frac{d^2T}{dr^2} + n\frac{dT}{dt} = 0 \qquad (1.32)$$

1.4. Initial and Boundary Conditions

In order to obtain a unique solution for a differential equation one needs to specify additional conditions–usually one for every derivative. Since the one dimensional heat equation contains $\frac{\partial T}{\partial t}$, so we will add an initial condition such as:

$T(x, 0) = f(x)$, $0 \le x \le L$ (initial temperature distribution), and the heat equation contain $\frac{\partial^2 T}{\partial x^2}$, so we usually add two boundary conditions. There are many types of boundary condition, see [5], for example:

1) Specified Temperature Boundary Condition:

$$T(0,t) = T_1 \text{ and } T(L,t) = T_2, \ t > 0,$$

where T_1 and T_2 are the specified temperatures. The specified temperatures can be constant, which is the case for steady heat conduction.

2) Specified Heat Flux Boundary Condition:

$$-k\frac{\partial T}{\partial x}(0,t) = c \ \text{ and } \ -k\frac{\partial T}{\partial x}(L,t) = -c\ , t > 0.$$

A Special Case:

$$\frac{\partial T}{\partial x}(0,t) = \frac{\partial T}{\partial x}(L,t) = 0, \ t > 0,$$

called insulated boundary conditions.

3) Convection Boundary Condition:

$$-k\frac{\partial T}{\partial x}(0,t) = h_1[\, T_1 - T(0,t)]$$

and

$$-k\frac{\partial T}{\partial x}(L,t) = h_2[\, T(L,t) - T_2],$$

where h_1 and h_2 are the convection heat transfer coefficients and T_1 and T_2 are the temperatures of the surrounding mediums on the two sides.

Chapter Two

Analytical Methods for Solving Transient Heat Conduction Problems

2.1. Introduction

In this Chapter, we will solve the heat conduction equation (*) in one dimension subject to some specific boundary and initial conditions for plane wall, cylinder and sphere.

Moreover, the thermal conductivity k is considered to be constant in all these cases.

2.2. One–dimensional Heat Conduction Equation

2.2.1. Steady–State

1) Plane Wall

Consider $\frac{\partial T}{\partial t} = 0$ in equation (1.18), we get:

$$\frac{\partial^2 T}{\partial x^2} + \frac{g}{k} = 0,\ k \text{ constant },\ 0 \le x \le L \qquad (2.1)$$

Where g is a function of t, with boundary conditions:

$$T(0) = T_1\ , T(L) = T_2$$

then by integrating equation (2.1) with respect to x twice, we obtain:

$$T(x) = -\frac{g}{2k}x^2 + ax + b \qquad (2.2)$$

Applying the boundary conditions, we get:

$T(0) = b = T_1$ and $T(L) = -\frac{g}{2k}L^2 + aL + b = T_2$

So, we get: $$a = \frac{T_2 - T_1}{L} + \frac{g}{2k}L$$

substituting a and b into equation (2.2), we get :

$$T(x) = -\frac{g}{2k}x^2 + [\frac{g}{2k}L + \frac{(T_2 - T_1)}{L}]x + T_1 \quad (2.3)$$

where $x \in [0, L]$, equation (2.3) is the general formula for one-dimensional heat conduction of plane wall with steady-state condition.

2) Cylinder:

Consider $\frac{\partial T}{\partial t} = 0$ in equation (1.26), we obtain:

$$\frac{1}{r}\frac{d}{dr}\left(r\frac{dT}{dr}\right) + \frac{g}{k} = 0,\ k \text{ constant},\ r_1 \le r \le r_2 \quad (2.4)$$

where g is a function of t, with boundary conditions:

$$T(r_1) = T_1\ , T(r_2) = T_2$$

then integrating equation (2.4) with respect to r, we get :

$$r\ \frac{dT}{dr} = -\frac{gr^2}{2k} + a \quad (2.5)$$

Again we integrate equation (2.5) with respect to r, to obtain:

$$T(r) = \frac{-gr^2}{4k} + aln(r) + b \quad (2.6)$$

Applying the above boundary conditions gives:

$$T(r_1) = \frac{-gr_1^2}{4k} + aln(r_1) + b = T_1$$

and

$$T(r_2) = \frac{-gr_2^2}{4k} + aln(r_2) + b = T_2$$

this gives:

$$a = \frac{(T_2 - T_1)}{ln(\frac{r_2}{r_1})} + \frac{g}{4k}\frac{(r_2^2 - r_1^2)}{ln(\frac{r_2}{r_1})} \quad (2.7)$$

and

$$b = [T_1 ln(r_2) - T_2 ln(r_1)] + \frac{g}{4k}\frac{[r_1^2\, ln(r_2) - r_2^2\, ln(r_1)]}{ln(\frac{r_2}{r_1})} \quad (2.8)$$

If we need to determine $T(r)$ at any $r \in [r_1, r_2]$, first we find a, b from equations (2.7) and (2.8), then we find $T(r)$ from equation (2.6).

3) Sphere

Consider $\frac{\partial T}{\partial t} = 0$ in equation (1.28), we obtain:

$\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{dT}{dr}\right) + \frac{g}{k} = 0$, k constant, $r_1 \le r \le r_2$ (2.9)

Where g is a function of t, with boundary conditions:

$$T(r_1) = T_1 \;\; and \;\; T(r_2) = T_2$$

Then by integrating equation (2.9) with respect to r twice, we obtain:

$$T\,(r) = \frac{-gr^2}{6k} - \frac{a}{r} + b \quad (2.10)$$

Apply the above boundary conditions, we obtain:

$$T(r_1) = \frac{-gr_1^2}{6k} - \frac{a}{r_1} + b = T_1$$

and

$$T(r_2) = \frac{-gr_2^2}{6k} - \frac{a}{r_2} + b = T_2$$

This yields:

$$a = (T_2 - T_1)\frac{r_2 r_1}{r_2 - r_1} + \frac{g}{6k}(r_2 r_1)(r_2 + r_1) \tag{2.11}$$

and

$$b = \left(\frac{T_2 r_2 - T_1 r_1}{r_2 - r_1}\right) + \frac{g}{6k}(r_2^2 + r_1 r_2 + r_1^2) \tag{2.12}$$

If we to need determine $T(r)$ at any $r \in [r_1, r_2]$, first we find a, b from equations (2.11) and (2.12), then we find $T(r)$ from equation (2.10).

2.2.2. Steady–State without Heat Generation

1) Plane wall:

Consider $\frac{\partial T}{\partial t} = 0$ and $g = 0$ in equation (1.18), we get:

$$\frac{d^2 T}{dx^2} = 0 \, , \; 0 \leq x \leq L \tag{2.13}$$

with boundary conditions:

$$T(0) = T_1 \, , T(L) = T_2$$

by integrating equation (2.13) with respect to x twice, we obtain:

$$T(x) = a\,x + b \tag{2.14}$$

Using the above boundary conditions, we get:

$$b = T_1 \qquad \text{and} \qquad a = \frac{T_2 - T_1}{L}$$

Substituting a and b in to equation (2.14) gives:

$$T(x) = \frac{T_2 - T_1}{L} x + T_1 \quad , x \in [0, L] \tag{2.15}$$

2) Cylinder:

Consider $\frac{\partial T}{\partial t} = 0$ and $g = 0$ in equation (1.26), we obtain:

$$\frac{d}{dr}\left(r\frac{dT}{dr}\right) = 0, r_1 \leq r \leq r_2 \qquad (2.16)$$

with boundary conditions:

$$T(r_1) = T_1 \; , \; T(r_2) = T_2$$

then by integrating equation (2.16) with respect to r twice, we have:

$$T(r) = a\, ln(r) + b \qquad (2.17)$$

subject to boundary conditions gives:

$$T(r_1) = aln(r_1) + b = T_1$$

and

$$T(r_2) = aln(r_2) + b = T_2$$

hence:

$$a = \frac{(T_2 - T_1)}{ln(\frac{r_2}{r_1})} \quad \text{and} \quad b = \frac{[T_1\, ln(r_2) - T_2 ln(r_1)]}{ln(\frac{r_2}{r_1})}$$

substituting a and b in equation (2.17), we obtain the general solution:

$$T(r) = \frac{(T_2 - T_1)}{ln(\frac{r_2}{r_1})}\, ln(r) + \frac{[T_1\, ln(r_2) - T_2 ln(r_1)]}{ln(\frac{r_2}{r_1})} \qquad (2.18)$$

where $r \in [r_1, r_2]$.

3) Sphere:

Consider $\frac{\partial T}{\partial t} = 0$ and $g = 0$ in equation (1.28), we get:

$$\frac{d}{dr}\left(r^2 \frac{dT}{dr}\right) = 0 \; , r_1 \leq \; r \leq r_2 \qquad (2.19)$$

with boundary conditions:

$$T(r_1) = T_1 \;\text{ and }\; T(r_2) = T_2$$

integrating equation (2.19) with respect to r twice, we obtain:

$$T\,(r) \;=\; \frac{-a}{r} + b \qquad (2.20)$$

subject to boundary conditions gives:

$$T\,(r_1) = \frac{-a}{r_1} + b = T_1 \quad \text{and} \quad T\,(r_2) = \frac{-a}{r_2} + b = T_2$$

This yields:

$$a = \frac{(T_2 - T_1)}{(r_2 - r_1)} r_2 r_1 \quad \text{and} \quad b = \frac{(T_2 r_2 - T_1 r_1)}{(r_2 - r_1)}$$

substituting a and b in to equation (2.20), we obtain:

$$T\,(r) = \frac{-(T_2 - T_1) r_2 r_1}{(r_2 - r_1) r} + \frac{(T_2 r_2 - T_1 r_1)}{(r_2 - r_1)} \qquad (2.21)$$

where $r \in \; [r_1, r_2]$.

2.2.3. Transient without Heat Generation

Equation (1.31) can be rewritten as:

$$\frac{1}{r^n}\frac{\partial}{\partial r}\left(r^n \frac{\partial T}{\partial r}\right) = \frac{1}{\alpha}\frac{\partial T}{\partial t} \qquad (2.22)$$

where $\alpha = \left(\frac{k}{\rho C_p}\right)$ is called the thermal diffusivity.

This equation subject to some specific boundary and initial conditions will be solved analytically when $n = 0, 1$ and 2 for plane wall, cylinder and sphere, respectively.

1) Separation of Variables Method

The method of separation of variables (sometimes called the method of Fourier) is a convenient method for solving the heat conduction equation, basically, it entails seeking a solution which breaks up into a product of functions, each of which involves only one variable. For example, in two variables, the solution of $R(x, y)$ in general can be written as, see [1]:

$$R(x, y) = f(x)\, g(y)$$

this separates out the partial differential equation into two or three ordinary differential equations, which related by a common constant, see [1], we begin for transient one-dimensional heat conduction equation for:

1) Plane Wall

Consider $g = 0$ in equation (1.18), we get:

$$\frac{\partial^2 T}{\partial x^2} = \frac{1}{\alpha}\frac{\partial T}{\partial t} \tag{2.23}$$

Subject to the boundary conditions:

$$T(0, t) = T(L, t) = 0\,,\ t > 0$$

and the initial condition:

$$T(x, 0) = R(x), \quad 0 < x < L$$

Using separation of variables method:

$$T(x,t) = f(x)\, g(t) \tag{2.24}$$

set equation (2.24) into equation (2.23), we get:

$$f''(x)\, g(t) = \frac{1}{\alpha} f(x)\, g'(t) \tag{2.25}$$

hence:

$$f''(x)/f(x) = g'(t)/\alpha g(t) = m \tag{2.26}$$

m is negative constant because physical reasons of a temperature function that either increases or decreases monotonically depending on the initial conditions and the imposed boundary conditions, the general solutions for the two equations in (2.26) becomes:

$$f(x) = acos(vx) + bsin(vx) \tag{2.27}$$

where $v^2 = -m$, and:

$$g(t) = c\, e^{\alpha m t} \tag{2.28}$$

where a, b and c are constants, substituting equations (2.27) and (2.28) into equation (2.24), we get:

$$T(x,t) = (acos(vx) + bsin(vx))(c\, e^{\alpha m t}) \tag{2.29}$$

now we introduce the boundary conditions:

$T(0,t) = 0$, implies: $0 = acos(vx)$

necessarily $a = 0$, we obtain:

$$f(x) = bsin(vx)$$

then:

$$T(L,t) = 0 \text{ , implies } 0 = bsin(Lv)$$

Necessarily, $sin(Lv) = 0$, hence:

$$v = \frac{n\pi}{L}, \; n \in N^*$$

we get to:

$$v_n = \frac{n\pi}{L}, n \in N^*$$

v_n are *the eigenvalues* and $sin(v_n x)$ are *the eigenfunctions* of the *Stumm-Liouville* problem, see [13], satisfied by $f(x)$. Each value of v_n yields an independent solution satisfying the heat equation as well as the two boundary conditions, we have an infinite number of independent solutions $T_n(x,t), n \in N^*$, then we obtain:

$$T_n(x,t) = d_n(sin(v_n x))\, e^{\alpha m_n t}, n \in N^* \qquad (2.30)$$

Where $d_n = b_n c_n$ and $v_n^2 = - m_n$.

Hence the general solution is:

$$T(x,t) = \sum_{n=1}^{\infty} T_n(x,t) = \sum_{n=1}^{\infty} d_n(sin(v_n x))e^{-\alpha v_n^2 t}$$

$$= \sum_{n=1}^{\infty} d_n(sin(\frac{n\pi}{L}x))e^{-\alpha(\frac{n\pi}{L})^2 t} \qquad (2.31)$$

then by the initial condition, we have:

$$T(x,0) = R(x) = \sum_{n=1}^{\infty} d_n (sin(\frac{n\pi x}{L})) \quad (2.32)$$

then by Fourier cosine series, gives:

$$d_n = \frac{2}{L}\int_0^L R(x)\left(sin\left(\frac{n\pi}{L}\right)x\right)dx \, , n \in N^* \quad (2.33)$$

2) Cylinder

Consider $g = 0$ in equation (1.26), we get:

$$\frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial R}{\partial r}) = \frac{1}{\alpha}\frac{\partial R}{\partial t} \quad (2.34)$$

subject to the boundary conditions:

$$R\ (r_1, t) = R(r_2, t) = 0\ , \forall t > 0$$

and the initial condition:

$$R\ (r, 0) = f(r),\ \forall\ r_1 \leq r\ \leq r_2$$

Using separation of variables method:

$$R(r,t) = M(r).\, g(t) \quad (2.35)$$

set equation (2.35) into equation (2.34), we obtain:

$$\frac{rg(t)M''(r)}{r} + \frac{g(t)M'(r)}{r} = \frac{1}{\alpha}\ M(r)g'(t)$$

hence:

$$\frac{rM''(r)}{rM(r)} + \frac{M'(r)}{rM(r)} = \frac{1}{\alpha}\frac{g'(t)}{g(t)} = -n^2$$

where $-n^2$ is negative constant for the same reason of m in the plane wall previously, we get the following equivalent system of ordinary differential equations:

$$\frac{g\prime(t)}{g(t)} = -\alpha n^2 \qquad \text{and} \qquad \frac{rM\prime\prime(r)}{rM(r)} + \frac{M\prime(r)}{rM(r)} = -n^2$$

the general solutions to these equations are:

$$g(t) = ae^{-\alpha n^2 t} \tag{2.36}$$

and

$$r^2M''(r) + rM'(r) + (rn)^2M(r) = 0 \tag{2.37}$$

since $0 \leq r_1 \leq r \leq r_2$ and $M(r_2) = 0$, then equation (2.37) is a special case of *Bessel's equation*, see [9], therefore, the only bounded solution is:

$$M(r) \ = \ b\,J_0(rn) \tag{2.38}$$

where b is a constant and $J_0(rn)$ *is the Bessel function of first kind of order zero* of the argument given by, see [6] :

$$J_0(rn) = \sum_{s=0}^{\alpha} \frac{(-1)^s(rn)^{2s}}{(2)^{2s}(s!)^2}\,, r > 0 \tag{2.39}$$

since $M(r_2) = 0,$ this requires that $J_0(r_2n\,) = \ 0,$ which defines the *eigenvalues* and *eigen functions* for this problem. The *eigenvalues* are thus the roots of:

$$J_0(r_2n_m) = 0\,,\ m \in Z^* \tag{2.40}$$

The particular solution of equation (2.37) becomes:

$$M_m(r) = b_m J_0(rn_m) \tag{2.41}$$

then equation (2.35) takes the form:

$$R_m(r,t) = M_m(r)g(t)$$

$$= C_m J_0(rn_m)e^{(-\alpha(n_m)^2 t)} \tag{2.42}$$

where $C_m = ab_m$, $m \in Z^*$.

Hence the general solution is:

$$R(r,t) = \sum_{m=1}^{\alpha} R_m(r,t)$$

$$= \sum_{m=1}^{\alpha} M_m(r)g(t)$$

$$= \sum_{m=1}^{\alpha} C_m J_0(r\, n_m)e^{(-\alpha n_m^2 t)} \tag{2.43}$$

for determine the $C_{m's}$, we use the initial condition:

$$R(r,0) = f(r) = \sum_{m=1}^{\alpha} C_m J_0(r\, n_m) \tag{2.44}$$

this is the *Fourier –Bessel series* representation of $f(r)$ and one can use the orthogonality property of *the eigen- functions* to write:

$$\int_0^{r_2} rJ_0(rn_v)f(r)dr = \sum_{m=1}^{\alpha} C_m \int_0^{r_2} rJ_0(rn_v)\, J_0(rn_m)\, dr$$

$$= C_v \int_0^{r_2} r(J_0(rn_v))^2 dr = \frac{r_2^2 C_v}{2} \left[J_0^2(r_2 n_v) + J_1^2(r_2 n_v)\right]$$

$$= \frac{r_2^2 C_v}{2} J_1^2(r_2 n_v) \tag{2.45}$$

Where $J_1(z) = -\frac{dJ_0(z)}{dz}$ is *the Bessel function of first kind of order one* of argument, therefore:

$$C_v = \frac{2}{r_2^2 J_1^2(r_2 n_v)} \int_0^{r_2} r J_0(r n_v) f(r) dr \qquad (2.46)$$

Hence the general solution is:

$$T(r,t) = \frac{2}{r_2^2} \sum_{m=1}^{\alpha} \frac{J_0(r n_v)}{n_v J_1(r_2 n_v)} e^{-\alpha n_m^2 t} \int_0^{r_2} r J_0(r n_v) f(r) dr \qquad (2.47)$$

3) Sphere

Consider $g = 0$ in equation (1.28), we obtain:

$$\frac{1}{r^2} \frac{\partial}{\partial r}\left(r^2 \frac{\partial S}{\partial r}\right) = \frac{1}{\alpha} \frac{\partial S}{\partial t} \qquad (2.48)$$

subject to the boundary conditions:

$$S(r_1, t) = S(r_2, t) = 0, t > 0$$

and the initial condition:

$$S(r, 0) = f(r), r_1 \leq r \leq r_2$$

using separation of variables method:

$$S(r,t) = N(r).G(t) \qquad (2.49)$$

set equation (2.49) into equation (2.48), we get:

$$\frac{2r}{r^2} \frac{N\prime(r)}{N(r)} + \frac{N\prime\prime(r)}{N(r)} = \frac{1}{\alpha} \frac{G\prime(t)}{G(t)} = -v^2$$

where $-v^2$ is negative constant again. Thus, we have:

$$G(t) = a\, e^{-\alpha v^2 t} \tag{2.50}$$

$$N(r) = b\,\frac{sin(vr)}{r} + c\,\frac{cos(vr)}{r} \tag{2.51}$$

but $N(r_2) = 0$ and necessary bounded at $r_1 = 0$, equation (2.51) becomes:

$$N(r) = b\,\frac{sin(vr)}{r} \tag{2.52}$$

where b and c are constants and $c = 0$ since the temperature must be bounded at $r = 0$. Moreover, the boundary condition at $r = r_2$ yields *the eigenvalues*:

$$v_n = \frac{n\pi}{r_2}\ , n \in \mathbb{Z}^*$$

and *the eigenfunctions*:

$$N_n(r) = \frac{b_n}{r} sin(v_n r)\ , n \in \mathbb{Z}^*$$

the particular solution of equation (2.48) becomes:

$$S_n(r,t) = N_n(r)G(t) = \frac{B_n}{r} sin(v_n r)\, e^{-\alpha v_n^2 t} \tag{2.53}$$

where $B_n = ab_n\ and\ n \in \mathbb{Z}^*$. Adding all these fundamental solutions of the problems gives:

$$S(r,t) = \sum_{n=1}^{\alpha} S_n(r,t) = \sum_{n=1}^{\alpha} \frac{B_n}{r} sin(v_n r)\, e^{-\alpha v_n^2 t} \tag{2.54}$$

using the initial condition, we have:

$$S(r,0) = \sum_{n=1}^{\alpha} \frac{B_n}{r} sin(v_n r)$$

then by Fourier sine series, gives:

$$B_n = \frac{2}{r_2} \int_0^{r_2} r\, f(r) si\, n \left(\frac{n\pi\, r}{r_2}\right) dr \qquad (2.55)$$

2) Laplace Transform Method

The Laplace transform method converts the heat conduction equation into an ordinary differential equation. Then the solution of the ODE must be inverted to give the general solution of the original problem.

Definition(1), see [21]: The Laplace transform of $G(x)$ is:

$$\mathcal{L}\,[G(x)] = \overline{G}(s) = \int_{x=0}^{\infty} e^{-xs} G(x) dx \qquad (2.56)$$

and the inverse transform is:

$$G(x) = \mathcal{L}^{-1}[\,\overline{G}(s)] \qquad (2.57)$$

where s *is the Laplace transform variable*. The conditions for the existence of the Laplace transform may be summarized as follows:

1) $G(x)$ is continuous or piecewise continuous.

2) $x^n|G(x)|$ is bounded as $x \to 0^+$ for some number n, such that $n < 1.$

3) $G(x)$ is of exponential order.

Now, we can use the Laplace transform method to solve the following heat conduction problems:

1) Plane Wall

Consider a plane wall, we can solve:

$$\frac{\partial^2 T(x,t)}{\partial x^2} = \frac{1}{\alpha}\frac{\partial T(x,t)}{\partial t} \tag{2.58}$$

where $0 < x < L$, and times $t > 0$,

subject to the boundary conditions:

$$T(0,t) = 0 , T(L,t) = at + b , \quad t > 0$$

and the initial condition:

$$T(x,0) = 0 , 0 < x < L.$$

Taking the Laplace transform for the equation (2.58), we get:

$$\frac{d^2\overline{T}(x,s)}{dx^2} = \frac{s}{\alpha}\overline{T}(x,s) , 0 < x < L \tag{2.59}$$

the laplace transform for the boundary conditions:

$$\overline{T}(0,s) = 0 \quad and \quad \overline{T}(L,s) = \frac{a}{s^2} + \frac{b}{s}$$

Hence, the general solution for equation is:

$$\overline{T}(x,s) = c_1 \cosh\sqrt{\frac{s}{\alpha}}\, x + c_2 \sinh\sqrt{\frac{s}{\alpha}}\, x \tag{2.60}$$

then by the boundary conditions, we have:

$$\overline{T}(0,s) = c_1 = 0 \text{ and } \overline{T}(L,s) = c_2 \sinh\sqrt{\frac{s}{\alpha}}\, L = \frac{a}{s^2} + \frac{b}{s}$$

implies:

$$c_2 = \frac{a+bs}{s^2 \sinh\sqrt{\frac{s}{\alpha}}\, L}$$

then the general solution of equation (2.60) becomes:

$$\overline{T}(x,s) = \frac{a+bs}{s^2 \, sinh\sqrt{\frac{s}{\alpha}}L} sinh\sqrt{\frac{s}{\alpha}}\, x \qquad (2.61)$$

taking the inverse transform for equation (2.61) gives:

$$T(x,t)=a\,\mathcal{L}^{-1}\left[\frac{sinh\sqrt{\frac{s}{\alpha}}x}{s^2.sinh\sqrt{\frac{s}{\alpha}}L}\right] + b\mathcal{L}^{-1}\left[\frac{sinh\sqrt{\frac{s}{\alpha}}x}{s.sinh\sqrt{\frac{s}{\alpha}}L}\right] \qquad (2.62)$$

then by using tables for inverse transform, see [13], we obtain:

$$T(x,t) = b\,\left[\frac{x}{L} + \frac{2}{\pi}\sum_{n=1}^{\infty}\frac{(-1)^n}{n} sin\lambda_n\, x e^{-\alpha\lambda_n^2 t}\right]$$

$$+a\,\left[\frac{xt}{L} + \frac{2}{\alpha L}\sum_{n=1}^{\infty}\frac{(-1)^n}{\lambda_n^3} sin\lambda_n\, x(1-e^{-\alpha\lambda_n^2 t})\right] \qquad (2.63)$$

where $\lambda_n = \frac{n\pi}{L}$.

Given the nature of the time–dependent boundary condition, we note that there is no steady–state solution to this Problem, see [13].

2) Cylinder

Consider a cylinder with radius r, we can solve:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial R(r,t)}{\partial r}\right) = \frac{1}{a}\frac{\partial R(r,t)}{\partial t} \qquad (2.64)$$

or

$$\frac{\partial^2 R(r,t)}{\partial r^2} + \frac{1}{r}\frac{\partial R(r,t)}{\partial r} = \frac{1}{a}\frac{\partial R(r,t)}{\partial t} \qquad (2.65)$$

where $0 < r < b$ and $t > 0$,

subject to the boundary conditions:

$$R(0,t) = R\,(b,t) = 0$$

and the initial condition:

$$R(r,0) = T_0$$

Taking the Laplace transform for equation (2.65) we have:

$$\frac{d^2\overline{R}(r,s)}{dr^2} + \frac{1}{r}\frac{d\overline{R}(r,s)}{dr} = \frac{s}{a}\overline{R}(r,s) - \frac{\overline{R}(r,0)}{a} \qquad (2.66)$$

with boundary conditions:

$$\overline{R}(r,0) = \frac{T_0}{s}$$

Hence, equation (2.66) becomes:

$$\frac{d^2\overline{R}(r,s)}{dr^2} + \frac{1}{r}\frac{d\overline{R}(r,s)}{dr} = \frac{s}{a}\overline{R}(r,s) - \frac{T_0}{as} \qquad (2.67)$$

multiplying equation (2.67) by r^2, we obtain:

$$r^2\,\frac{d^2\overline{R}(r,s)}{dr^2} + r\,\frac{d\overline{R}(r,s)}{dr} - \left(r\sqrt{\frac{s}{a}}\right)^2\overline{R}(r,s) = -\frac{r^2T_0}{as} \qquad (2.68)$$

equation (2.68) is *a modified Bessel equation of order zero*, see [6], let $ir = m$, we obtain:

$$m^2\,\frac{d^2\overline{R}(m,s)}{dm^2} + m\,\frac{d\overline{R}(m,s)}{dm} + \left(m\,\sqrt{\frac{s}{a}}\right)^2\overline{R}(m,s) = \frac{m^2T_0}{a\,s} \qquad (2.69)$$

equation (2.69) *is a nonhomogeneous modified Bessel equation of order zero*, with general solution:

$$\overline{R}(m,s) = \overline{R}_h(m,s) + \overline{R}_p(m,s) \tag{2.70}$$

where $\overline{R}_h(m,s)$ is the homogeneous solution of *modified Bessel equation of order zero*, get as:

$$\overline{R}_h(r,s) = C_1 J_0\left(\sqrt{\frac{s}{a}}m\right) + C_2 W_0\left(\sqrt{\frac{s}{a}}m\right) \tag{2.71}$$

and $\overline{R}_p(m,s)$ is the nonhomogeneous solution of *modified Bessel equation of order zero*, get as:

$$R_p(m,s) = Am^2 + Bm + C \tag{2.72}$$

implies:

$$\frac{dR_p}{dm} = 2Am + B \tag{2.73}$$

and

$$\frac{d^2R_p}{dm^2} = 2A \tag{2.74}$$

Substitute equations (2.72), (2.73) and (2.74) into equation (2.69), we obtain:

$$\overline{R}_p(m,s) = \frac{T_0}{s^2} \tag{2.75}$$

hence:

$$\overline{R}(m,s) = c_1 J_0\left(\sqrt{\frac{s}{a}}m\right) + c_2 W_0\left(\sqrt{\frac{s}{a}}m\right) + \frac{T_0}{s^2} \tag{2.76}$$

Where:

$$J_0\left(\sqrt{\frac{s}{a}}m\right) = \sum_{n=0}^{\infty} \frac{\left(\frac{s}{a}m^2\right)^n}{4^n (n!)^2}, \forall m > 0 \tag{2.77}$$

and $J_0(0) = 1$, see [9], now we have:

$$W_0\left(\sqrt{\frac{s}{a}}m\right) = \frac{2}{\pi}\left[\beta + \ln\left(\sqrt{\frac{s}{a}}\frac{m}{2}\right)\right] J_0\left(\sqrt{\frac{s}{a}}m\right)$$

$$-\sum_{n=1}^{\infty} \frac{H_n(\frac{s}{a}m^2)^n}{4^n(n!)^2} \ ,\forall m > 0 \tag{2.78}$$

Where $W_0(0) = -\infty$, see [10], $H_n = \frac{1}{2} + \frac{1}{4} \dots \dots \dots + \frac{1}{2n}$ and β is *the Euler–Mascheroni constant* defined by:

$\beta = lim_{n\to\infty}(H_n - lnn) \cong 0.5772$. Then by BC's, we have:

$$c_2 \ must \ = 0 \quad \text{and} \quad c_1 = -\frac{T_0}{s^2}.$$

Hence, the general solution of equation (2.70) is:

$$\overline{R}(m, s) = -\frac{T_0}{s^2} J_0\left(\sqrt{\frac{s}{a}}m\right) + \frac{T_0}{s^2} \tag{2.79}$$

Taking $\mathcal{L}^{-1}$, the general solution of equation (2.64) is:

$$R(m, t) = \frac{-1}{2\pi i} \int_{c-i\infty}^{c+i\infty} \frac{T_0}{s^2} J_0\left(\sqrt{\frac{s}{a}}m\right) e^{st} \, ds + T_0 t \tag{2.80}$$

3) Sphere

Consider a sphere of radius r, we can solve:

$$\frac{1}{r}\frac{\partial^2(rS(r,t))}{\partial r^2} = \frac{1}{a}\frac{\partial S(r,t)}{\partial t} \tag{2.81}$$

where $0 < r < b, and \ t > 0$,

subject to the boundary conditions:

$$S(0, t) = S(b, t) = 0$$

and the initial condition: $S(r, 0) = T_0$.

First, let $rS(r,t) = R(r,t)$, so equation (2.81) becomes:

$$\frac{\partial(R(r,t))}{\partial r^2} = \frac{1}{a}\frac{\partial R(r,t)}{\partial t} \tag{2.82}$$

then taking the Laplace transform for the equation (2.82), we obtain:

$$\frac{d^2(\overline{R}(r,s))}{dr^2} - \frac{s}{a}\overline{R}(r,s) = \frac{-rT_0}{a}, \quad 0 < r < b \tag{2.83}$$

this solution requires superposition forms, see [21], we obtain:

$$\overline{R}(r,s) = C_1 \cosh\sqrt{\frac{s}{a}}\, r + C_2 \sinh\sqrt{\frac{s}{a}}\, r + \frac{rT_0}{s} \tag{2.84}$$

then using the boundary conditions, we get:

$$\overline{R}(r,s) = \frac{rT_0}{s} - \frac{bT_0 \sinh\sqrt{\frac{s}{a}}\, r}{s.\sinh\sqrt{\frac{s}{a}}\, b} \tag{2.85}$$

Taking $\mathcal{L}^{-1}$, the general solution of equation (2.82) is:

$$R(r,t) = rT_0 - bT_0\left[\frac{r}{b} + \frac{2}{\pi}\sum_{n=1}^{\infty}\frac{(-1)^n}{n} \sin(\lambda_n r)\, e^{-a\lambda_n^2 t}\right] \tag{2.86}$$

where $\lambda_n = \frac{n\pi}{b}$,

hence, the general solution of equation (2.81) is:

$$S(r,t) = \frac{-2bT_0}{\pi}\sum_{n=1}^{\infty}\frac{(-1)^n}{n}\frac{\sin(\lambda_n r)}{r}\, e^{-a\lambda_n^2 t} \tag{2.87}$$

3) Duhamel's Methods

Duhamel's theorem provides one of extending an analytical solution that derived assuming a time invariant term in order to consider the temperature response to an arbitrary time variation of that term. It is somewhat easier to state Duhamel's theorem than it is understand it; **Duhamel's theorem says**, see [13]:

If $R(x\,,t)$ is the response of a linear system with a $zero$ initial temperature to a single , constant nonhomogeneous term with magnitude of unity , then the response of the same system to a single , time varying nonhomogeneous term with magnitude $B(t)$ can be obtained from the fundamental solution according to:

$$T(x,t) = \int_{v=0}^{t} R(x\,,t-v)\,\frac{dB(v)}{dv}\,dv + B(0)\,R(x,t) \qquad (2.88)$$

where $B(t)$ must be continuous in time. In order to apply Duhamel's theorem , it is necessary to have a problem with a $zero$ initial temperture and a single nonhomogeneous term that varies in time. The problem must be divided into sub problems. Once this has been accomplished, it is necessary to obtain the fundamental solution $R(x,t)$ to the sub problem with the time varying term replaced by a constant value, 1. Finally, Duhamel's theorem can be applied to the fundamental solution according to equation (2.88). Now, we can use Duhamel's theorem to solve the following heat conduction problems:

1) Plane Wall

Consider a plane wall satisfying:

$$\frac{\partial^2 T(x,t)}{\partial x^2} = \frac{1}{a}\frac{\partial T(x.t)}{\partial t} \qquad (2.89)$$

where $0 < x < L, and \;\; t > 0$,

subject to the boundary conditions:

$T(0,t) = 0$, which is necessary restriction for Duhamel's theorem as presented,

$$T(L,t) = B(t) = \begin{cases} bt & ,0 \leq t < v_1 \\ 0 & , \quad t > v_1 \end{cases}$$

and initial condition:

$$T(x,0) = 0.$$

The appropriate auxiliary problem here is:

$$\frac{\partial^2 R(x,t)}{\partial x^2} = \frac{1}{a}\frac{\partial R(x.t)}{\partial t} \qquad (2.90)$$

subject to boundary and initially conditions:

$$R(0,t) = 0 \,, R(L,t) = 1$$

and

$$R(x,0) = 0 \,.$$

The desired function $R(x,t-v)$ is obtained from the general solution of equation (2.90), see [13]:

$$R(x,t) = \frac{x}{L} + \frac{2}{L}\sum_{m=1}^{\infty} e^{-aB_m^2 t}.\frac{(-1)^m}{B_m} sinB_m\, x \qquad (2.91)$$

where $B_m = \frac{m\pi}{L}$, then for $t < v_1$, we obtain:

$$T(x,t) = \int_{v=0}^{t} R(x\,,t-v)\,\frac{dB(v)}{dv}dv$$

$$= \frac{bx}{L}t + \frac{2b}{L}\sum_{m=1}^{\infty}[\frac{(-1)^m}{aB_m^3}\left(1- e^{-aB_m^2 t}\right)sin(B_m\, x)] \qquad (2.92)$$

and for $t > v_1$, we obtain:

$$T(x,t) = \int_{v=0}^{v_1} R(x\,,t-v)\frac{dB(v)}{dv}.dv$$

$$+ \int_{v=v_1}^{t} R(x\,,t-v)\,\frac{dB(v)}{dv}.dv + R(x\,,t-v_1)\,.\Delta B \qquad (2.93)$$

where $\frac{dB}{dv} = b$ when $t < v_1$, $\frac{dB}{dv} = 0$ when $t > v_1$ and $\Delta B = -bv_1$, hence, the general solution of equation (2.93) is:

$$T(x,t) = [\frac{bxv_1}{L} + \frac{2b}{L}\sum_{m=1}^{\infty}\frac{(-1)^m}{aB_m^3}\left(1- e^{-aB_m^2 v_1}\right) sinB_m\, x]$$

$$-[\frac{bv_1 x}{L} + \frac{2bv_1}{L}\sum_{m=1}^{\infty}\frac{(-1)^m}{B_m} sinB_m\, x.\ e^{-aB_m^2(t-v_1)}] \qquad (2.94)$$

2) Cylinder

Consider the heat conduction problem:

$$\frac{\partial^2 R(r,t)}{\partial r^2} + \frac{1}{r}\frac{\partial R(r,t)}{\partial r} = \frac{1}{a}\frac{\partial R(r,t)}{\partial t} \qquad (2.95)$$

where $0 < r < d\ ,\ t > 0$,

subject to the boundary conditions:

$$R(0,t) = 0 \text{ and } R(d,t) = B(t)$$

assume that $B(t)$ has no discontinuities, and initial condition:

$$R(r,0) = 0$$

The appropriate auxiliary problem here is:

$$\frac{\partial^2 Q(r,t)}{\partial r^2} + \frac{1}{r}\frac{\partial Q(r.t)}{\partial r} = \frac{1}{a}\frac{\partial Q(r,t)}{\partial t} \qquad (2.96)$$

subject to boundary and initially conditions:

$$Q(d,t) = 1 \,, Q(0,t) = 0 \text{ and } Q(r,0) = 0$$

the described function $Q(r,t-v)$ is obtained from the general solution of equation (2.96), see [9]:

$$Q(r,t)=1 - \frac{2}{d}\sum_{m=1}^{\infty} \frac{J_0(N_m r)}{N_m J_1(N_m d)} \; e^{-aN_m^2 t} \qquad (2.97)$$

where J_0 , J_1 are *the Bessel's functions of first kind of order zero*, *one* respectively, see [6], and N_m *is the eigenvalues* of the positive roots of $J_0(N_m d) = 0$.

Hence, by *Duhamel's method* and integration equation (2.88) by parts, the general solution for equation (2.95) becomes:

$$R(r,t)= \frac{2a}{d}\sum_{m=1}^{\infty} e^{-aN_m^2 t}\left(\frac{N_m J_0(N_m r)}{J_1(N_m d)}\right)\int_{v=0}^{t} e^{(aN_m^2 v)} B(v)dv \qquad (2.98)$$

Integrating by parts and using **BC's** and **IC**, we obtain:

$$R(r,t)=B(t) - \frac{2}{d}\sum_{m=1}^{\infty}\frac{J_0(N_m r)}{N_m J_1(N_m d)} * \left[B(0)e^{-aN_m^2 t} + \int_{v=0}^{t} e^{-aN_m^2(t-v)}\frac{dB(v)}{dv}dv\right] \quad (2.99)$$

2.2.4 Non-Homogeneous Transient Heat Conduction Problem

In this section we will solve some of the nonhomogeneous heat conduction equation, i.e., the heat generation $g(x,t)$ is source of strength. For solving these equations, we use Green's function method. While the method of separation of variables is applicable to a broad class of problems, the method is not often applicable for solving nonhomogeneous we consider the following one dimensional, non-homogeneous boundary value problem of heat conduction for plane wall and cylinder.

Definition (2), see [15]: Suppose that we want to solve a linear inhomogeneous equation of the form:

$$L\big(u(x)\big) = f(x)$$

where L is a differential operator, $u(x)$ and $f(x)$ are functions whose domain is D, it happens that differential operations often have inverses that are integral operators, so for previous equation, we might expect a solution of the form:

$$u(x) = \int_D G(x,z)f(z)dz$$

if such a representation exists, *the kernel* of this integral operator $G(x,z)$ is called *the Green's function*.

1) Plane wall

Consider a one dimensional plane wall over the domain $0 \leq x \leq L$, for $t > 0$, the non-homogeneous boundary value problem of heat conduction problem, given as:

$$\frac{\partial^2 T(x,t)}{\partial x^2} + \frac{1}{k} g(x,t) = \frac{1}{a} \frac{\partial T(x,t)}{\partial t} \tag{2.100}$$

where $0 < x < L$, $t > 0$ and $g(x,t)$ is *heat generation*, subject to the boundary conditions:

$$T(0,t) = f_1(t) \text{ and } T(L,t) = f_2(t)$$

and initial condition:

$$T(x,0) = R(x).$$

To obtain the temperature distribution $T(x,t)$, for $t > 0$ by the Green's function technique, see [8], we consider the homogeneous version of the problem defined above over the same region:

$$\frac{\partial^2 u(x,t)}{\partial x^2} = \frac{1}{a} \frac{\partial u(x,t)}{\partial t}, \quad 0 < x < L, t > 0 \tag{2.101}$$

subject to the boundary conditions:

$$u(0,t) = u(L,t) = 0$$

and initial condition:

$$u(x,0) = R(x)$$

the general solution for equation (2.101) is, see [8]:

$$u(x,t)= \int_0^L \left[\frac{2}{L}\sum_{n=1}^{\alpha}(sin(\lambda_n\, x)sin(\lambda_n\, z)e^{-a\lambda_n^2 t})\right] * R(z)dz \quad (2.102)$$

where *the eigenvalues* are given by the expression $\lambda_n = \frac{n\pi}{L}$, $n = 1,2,3,...$,

then by comparing this solution by *Green's function method*, we obtain:

$$u(x,t)=\int_0^L G(x,t|\, z,\tau)|_{\tau=0}\, R(z)dz \quad (2.103)$$

from the kernel, which becomes, see[21]:

$$G(x,t|z,\tau)|_{\tau=0} = \frac{2}{L}\sum_{n=1}^{\infty}\left(sin(\lambda_n\, x)\sin(\lambda_n\, z)e^{-a\lambda_n^2(t-\tau)}\right) \quad (2.104)$$

Hence, the general solution of the nonhomogeneous problem for equation (2.100) is given in terms of *the Green's function* as, see [8]:

$$T(x,t) = \int_0^L G(x,t|\, z,\tau)|_{\tau=0}R(z)dz + \frac{a}{k}\int_0^t\int_0^L G(x,t|\, z,\tau)g(z,\tau)dzd\tau + a\int_0^t \frac{\partial G(x,t|z,\tau)}{\partial z}|_{z=0}\, f_1(\tau)d\tau - a\int_0^t \frac{\partial G(x,t|z,\tau)}{\partial z}|_{z=L}\, f_2(\tau)d\tau \quad (2.105)$$

However, depending on the boundary and initial conditions, we obtain:

$$-k\frac{\partial G}{\partial z}|_{z=0} = -h\,G|_{z=0} \quad \text{implies} \quad \frac{\partial G}{\partial z}|_{z=0} = \frac{1}{k}G|_{z=0}$$

and

$$-k\frac{\partial G}{\partial z}|_{z=L} = +h\,G|_{z=L} \quad \text{implies} \quad -\frac{\partial G}{\partial z}|_{z=L} = \frac{1}{k}G|_{z=L}$$

where we have used our sign convention of matching positive conduction and convection at each boundary and have set h to unity. Introducing *the Green's function* of equation (2.105) into (2.106), we obtain, see [8]:

$$T(x,t) = \left[\frac{2}{L}\sum_{n=1}^{\infty} sin(\lambda_n\, x)e^{-a\lambda_n^2 t} * \int_0^L sin(\lambda_n\, z)\, R(z)d\, z\right] +$$
$$[\frac{2a}{kL}\sum_{n=1}^{\infty} sin(\lambda_n x)e^{-a\lambda_n^2 t} * \int_\tau^t \int_0^L sin(\lambda_n z)\, g(z,\tau)e^{-a\lambda_n^2 \tau} d\, z\, d\tau\,]$$
$$+[\frac{2a}{L}\sum_{n=1}^{\infty} \lambda_n sin(\lambda_n\, x)e^{-a\lambda_n^2 t} * \int_0^t e^{-a\lambda_n^2 \tau} f_1(\tau)\, d\tau\,]$$
$$-[\sum_{n=1}^{\infty} (-1)^n \lambda_n sin(\lambda_n\, x)e^{-a\lambda_n^2 t} * \int_0^t e^{-a\lambda_n^2 \tau} f_2(\tau)\, d\tau\,] \qquad (2.106)$$

2) Sphere

Consider a one-dimensional sphere over the domain $a \le r \le b$ that is initially at a temperature $S(r)$, for $t > 0$. We use *Green's function method* for solving nonhomogeneous heat conduction problem given as:

$$\frac{1}{r}\frac{\partial^2}{\partial r^2}(rS) + \frac{1}{k}g(r,t) = \frac{1}{c}\frac{\partial S}{\partial t}\ , a < r < b\ , t > 0 \qquad (2.107)$$

subject to the boundary conditions:

$$S(a,t) = 0 \quad \text{and} \quad S(b,t) = f(t)$$

and initial condition:

$$S(r,0) = R(r).$$

To determine the desired *Green's function*, we consider the homogeneous version of the problem for the same region as, see [8]:

$$\frac{1}{r}\frac{\partial^2}{\partial r^2}(rW) = \frac{1}{c}\frac{\partial W}{\partial t} \;, a < r < b \;, for\; t > 0 \qquad (2.108)$$

subject to the boundary conditions:

$$W(a,t) = W(b,t) = 0$$

and initial condition:

$$W(r,0) = R(r)$$

this homogeneous problem has a solution given as:

$$W(r,t) = \int_0^L [2\sum_{n=1}^{\alpha} \frac{sin(B_n(r-a))\,\sin(B_n(z-a))}{z\,r(b-a)} e^{-cB_n^2 t}] * R(z)z^2 dz \qquad (2.109)$$

where $B_n = \frac{n\pi}{b-a}$, $n = 1,2,3$, ... and $B_o = 0$ is *a trivial eigenvalue* and has been dropped from the summation, then we seek a solution to the homogeneous problem of the form:

$$W(r,t) = \int_{z=a}^{b} G(r,t|\,z,\tau)|_{\tau=0} R(z)z^2 dz \qquad (2.110)$$

where:

$$G(r,t|z,\tau)|_{\tau=0} = 2\sum_{n=1}^{\infty} \frac{sinB_n(r-a)\;sinB_n(z-a)}{z\,r(b-a)} e^{-cB_n^2 t} \qquad (2.111)$$

now replacing t with $(t-\tau)$ into equation (2.111), we obtain:

$$G(r,t|z,\tau) =$$

$$2\sum_{n=1}^{\infty}\frac{sinB_n(r-a)\ sinB_n(z-a)}{z\,r(b-a)}e^{-cB_n^2(t-\ \tau)} \quad (2.112)$$

then by using *Green's function* and boundary conditions the general solution of the nonhomogeneous equation (2.107) is, see [8]:

$$S(r,t)= \int_a^b G(r,t|\,z,\tau)|_{\tau=0}R(z)z^2dz$$

$$+\frac{c}{k}\int_{\tau=0}^{t}\int_a^b G(r,t|\,z,\tau)g(z,\tau)z^2dz\,d\tau -$$

$$c\int_{\tau=0}^{t}[z^2\ \frac{\partial G(r,t|z,\tau)}{\partial z}]\,|_{z=b}\ f(\tau)d\tau \quad (2.113)$$

with the first term accounting for the initial temperature distribution, the second term accounting for the internal energy generation and the third term accounting for the non-homogeneity at $r=b$, we have:

$$-k\frac{\partial G}{\partial z}|_{z=b} = +h\,G|_{z=b} \ \text{ and } -\frac{\partial G}{\partial z}|_{z=b} = \frac{1}{k}\ G|_{z=b}$$

The general solution for equation (2.107) is, see [8]:

$$S(r,t)=$$

$$\left[2\sum_{n=1}^{\infty}\frac{sinB_n(r-a)}{r(b-a)}e^{-cB_n^2t} * \int_a^b sinB_n(z-a)R(z)z\,d\,z\right]$$

$$+[\ \frac{2c}{k}\sum_{n=1}^{\infty}\frac{sinB_n(r-a)}{r(b-a)}e^{-cB_n^2t} *$$

$$\int_0^t\int_a^b sinB_n(\,z-a)\ e^{cB_n^2\tau}g(z,\tau)z\,dz\,d\tau\ ]$$

$$[2c\sum_{n=1}^{\infty}\frac{sinB_n(r-a)[bB_ncosB_n(b-a)-sinB_n(b-a)]}{r(b-a)} *$$

$$e^{-cB_n^2t}\ \int_{\tau=0}^{t}e^{cB_n^2\tau}f(\tau)d\tau] \quad (2.114)$$

Chapter Three

Numerical treatment of heat conduction Problems

3.1. Introduction

In this chapter, we will solve the one-dimensional *heat conduction equation* for plane wall, cylinder and sphere, using *finite difference method* **FTCS** (Forward-Time Central-Space). These equations are:

1) **Plane Wall:**

$$\frac{\partial^2 T}{\partial x^2} + g = \frac{1}{a}\frac{\partial T}{\partial t} \tag{3.1}$$

2) **Cylinder:**

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial R}{\partial r}\right) + g = \frac{1}{a}\frac{\partial R}{\partial t} \tag{3.2}$$

3) **Sphere:**

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\ \frac{\partial S}{\partial r}\right) + g = \frac{1}{a}\frac{\partial S}{\partial t} \tag{3.3}$$

these solutions will be subject to some boundary and initial conditions.

3.2. Finite Difference Method

The finite difference method is one of the methods used to obtain numerical solutions to solve heat conduction equation. The idea of *finite difference methods* is to replace the partial derivatives equation using *finite difference approximations* with $O(h^n)$ *errors* (*where* $h = \Delta x_i$ *= the local distance between adjacent points),* see [10], it is involves using discrete approximations like:

$$\frac{\partial R(x_i)}{\partial x} = R'\,(x_i) \approx \frac{R(x_{i+1}) - R(x_i)}{h} \tag{3.4}$$

where $R\,(x_{i+1}) \approx R(x_i + h)$. This procedure converts the region to a mesh grid of points where the dependent variables approximated.

The replacement of partial derivatives with difference approximations formulas depends on some theories and definitions we will mention them first.

3.2.1. Taylor's Theorem

Let $R(x)$ has $n \in N$ continuous derivatives under the interval $]a, b[$, then for $a < x_i$ and $x_i + h < b$, we can write the value of $R(x)$ and it's derivatives near the point $x_i + h$ as:

$$R(x_i + h) =$$

$$R(x_i) + h\,R'(x_i) + \frac{h^2}{2!}R''(x_i) + \dots\dots + \frac{h^n}{n!}R^{(n)}(x_i) + O(h^n)$$
$$= \sum_{m=0}^{n}\frac{h^m}{m!}R^{(m)}(x_i) + O(h^n) \qquad (3.5)$$

where:

1) $R^{(m)}(x_i)$ Is the m^{th} derivative of *R* with respect to x at the point x_i.

2) $O(h^n)$ (*pronuned as order h to the n*) is an *unknown error* term that satisfies the property : for $g(h) = O(h^n)$ then $\lim_{h\to 0}\frac{g(h)}{h^n} = C$, for any *non-zero* constant C, see [11]. When we eliminate *the error term,* $O(h^n)$, from the equation (3.5), we get an approximation to $R(x_i + h)$. .

3.2.2. First Order Forward Difference Method

When solve the equation (3.5) for $R'(x_i)$, we get:

$$R'(x_i) = \frac{R(x_i+h) - R(x_i)}{h} - \frac{h}{2!}R''(x_i) - \dots$$
$$- \frac{h^{(n-1)}}{(n-1)!}R^{(n-1)}(x_i) - O(h^{n-1}) \qquad (3.6)$$

notice that the powers of h multiplying the partial derivatives have been reduced by one. Substitute the approximate solution for the exact solution, we obtain:

$$R'\ (x_i) \approx \frac{R(x_{i+1})-R(x_i)}{h} - \frac{h}{2!} R''\ (x_i) - \ldots$$

$$- \frac{h^{(n-1)}}{(n-1)!}\ R^{(n-1)}(x_i) - O(h^{n-1}) \qquad (3.7)$$

then by *the mean value theorem*, see [15], can be used to replace the higher order derivatives as:

$$\frac{h}{2!} R''\ (x_i) + \ \ldots\ + \frac{h^{(n-1)}}{(n-1)!}\ R^{(n-1)}(x_i) = \frac{h}{2!} R''\ (s) \qquad (3.8)$$

where $x_i \leq s \leq x_{i+1}$, where the right hand side of equation (3.8) is called *the truncation error of the finite difference approximation*, see [18]. So equation (3.7) becomes:

$$R'\ (x_i) \approx \frac{R(x_{i+1})-R(x_i)}{h} - \frac{h^2}{2!}\ R''\ (s) \qquad (3.9)$$

In general, S and $R(x)$ are unknown so, $R''\ (x)$ cannot be computed, although the exact magnitude of *the truncation error* cannot be known (**unless the analytical solution of $\boldsymbol{R(x)}$ known**). *The truncation error* simply written as:

$$\frac{h^2}{2}\ R''\ (s) = O(h\ ) \qquad (3.10)$$

equation (3.10) means *the truncation error* is a product of an unknown constant and h , so this term approaches $zero$ as $(h)$is reduced, equation (3.9) can be written as:

$$R''(x_i) = \frac{R(x_{i+1})-R(x_i)}{h} + O(h\ ) \qquad (3.11)$$

This equation is called *the forward difference formula*, because it involves nodes x_i and x_{i+1} . *The forward difference approximation* has *a truncation error* that is $O(h)$. The size of *the truncation error* is mostly under our control, because we can choose the mesh size(h).

3.2.3. First Order Backward Difference Method

Replace $h = -h$ in equation (3.5) and similarly the steps in first order forward difference, we have:

$$R'(x_i) = \frac{R(x_i)-R(x_{i-1})}{h} + O(h) \qquad (3.12)$$

this equation is called *the backward difference formula*, because it involves the values of $R(x)$ at x_i and x_{i-1} .

3.2.4 First Order Central Difference Method

When we write *the* Taylor's *series expansions* for $R(x_{i+1})$ and $R(x_{i-1})$ we obtain:

$$R(x_{i+1}) = R(x_i) + h\,R'(x_i) + \frac{h^2}{2!}R''(x_i) + \dots + \frac{h^n}{n!}\,R^{(n)}(x_i) + O(h^n) \qquad (3.13)$$

$$R(x_{i-1}) = R(x_i) - h\,R'(x_i) + \frac{h^2}{2!}R''(x_i) - \dots + \frac{(-h)^n}{n!}\,R^{(n)}(x_i) + O(h^n) \qquad (3.14)$$

Subtracting equation (3.14) from (3.13), we obtain:

$$R(x_{i+1}) - R(x_{i-1}) =$$

$$2h\,R'\,(x_i) + \frac{2h^3}{3!}R'''\,(x_i) + \ldots\;\ldots + O(h^n) \qquad (3.15)$$

solving for $R'\,(x_i)$, we get:

$$R'\,(x_i) = \frac{R(x_{i+1}) - R(x_{i-1})}{2h} + O(h^2) \qquad (3.16)$$

this equation is called *the central difference approximation to* $R'\,(x_i)$.

3.2.5. Second Order Central Difference Method

When we add equations (3.13) and (3.14), we get:

$$R(x_{i+1}) + R(x_{i-1}) =$$

$$2\,R(x_i) + h^2\,R''\,(x_i) + \frac{2h^4}{4!}\,R^{(4)}(x_i) + \ldots\ldots + O(h^n) \qquad (3.17)$$

solving for $R''\,(x_i)$, we obtain:

$$R''\,(x_i) = \frac{R(x_{i+1}) - 2R(x_i) + R(x_{i-1})}{h^2} + O(h^2) \qquad (3.18)$$

this equation called *the central difference approximation to the second derivative* $(R''\,(x_i))$, see [16].

3.2.6. The Discrete Mesh

The finite difference method obtains an approximation solution for $T(x,t)$ at a finite set of x and t. The discrete x are uniformly spaced in the interval $0 \leq x \leq L$ such that $x_i = (i-1)h\,, i = 1,2,3,\ldots, N$ where N is the total number of spatial nodes. Similarly, the discrete t are uniformly spaced in $0 \leq t \leq t_{max.}$, where $t_j = (j-1)\Delta t\ , j = 1\ , 2\ , 3\ , \dots , M$ where M is the number of time steps and Δt is the size of a time step where: $h = \Delta x = \frac{L}{N-1}$ and $\Delta t = \frac{t_{max.}}{M-1}$, see figure (1) where used for solution to *the one-dimensional heat equation*.

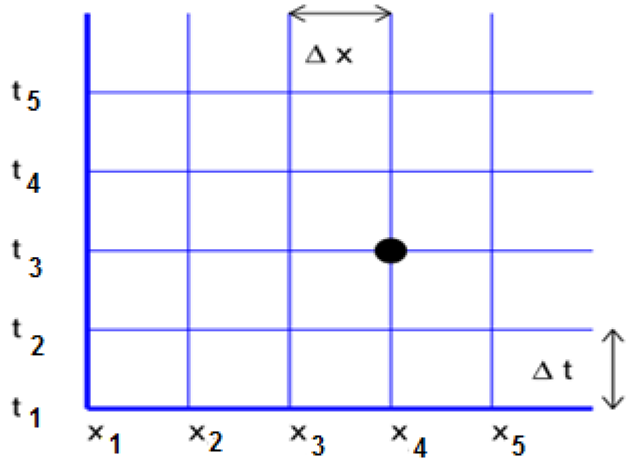

Figure 3.1: finite difference mesh or grid.

3.3. Difference Equations Forms

We use central difference approximation for space derivative and forward difference approximation for time derivative.

3.3.1. Plane Wall:

Consider the heat conduction problem:

$$\frac{\partial^2 T(x,t)}{\partial x^2} + g(x,t) = \frac{1}{a}\frac{\partial T(x,t)}{\partial t} \qquad (3.19)$$

where $0 < x < L$, $T_{max.} > t > 0$, $g(x,t)$ is *heat genaration* with boundary conditions:

$$T(0\ ,t\ ) = b_1,\ T(\ L, t\ )\ = b_n\ \ , T_{max.} > t > 0$$

and initial condition:

$$T(\ x\ ,0\ ) =\ f(x),\ 0 <\ x\ < L$$

then by use finite difference method we have:

$$\frac{T(x_{i+1},t_j)-2T(x_i,t_j)+T(x_{i-1},t_j)}{(\Delta x)^2} + O(\Delta x)^2 + g(x_i, t_j)$$

$$= \frac{1}{a}\ \frac{T(x_i,t_{j+1})-T(x_i,t_j)}{\Delta t} + O(\Delta t) \qquad (3.20)$$

where the discrete domain is:

$x_i = (i-1)\Delta x\ , i = 1, 2, 3, \dots, N$ and $t_j = (j-1)\Delta t$,

$j = 1, 2, 3, \dots, M$, subject to the boundary and initial conditions:

$$T(x_1, t_j) =\ T(0, t_j) = b_1\ , j = 1,2,3, \dots M\ ,$$

$$T(x_N, t_j) =\ T(L, t_j) = b_n\ , j = 1,2,3, \dots M$$

and

$$T(x_i, t_1) =\ T(x_i, 0) = f(x_i)\ , i = 1,2,3, \dots N$$

then solving equation (3.20) for approximate $T(x_i, t_{j+1})$, we have:

$$T(x_i, t_{j+1}) =$$

$$\lambda T(x_{i+1}, t_j) + (1-2\lambda)T(x_i, t_j) +\ \lambda T(x_{i-1}, t_j) + bg(x_i, t_j) \qquad (3.21)$$

where $\lambda = \frac{a\Delta t}{(\Delta x)^2}$, $b = a\Delta t, i = 2,3, \dots, N-1,\ j = 2,3, \dots, M.$

This result with *local truncation error* $(O(\Delta x)^2 +\ O(\Delta t)\ )$, see [18], which has the symbol $\boldsymbol{T_{error}}$.

3.3.2 Cylinder:

Consider the heat conduction problem:

$$\frac{\partial^2 R(r,t)}{\partial r^2} + \frac{1}{r}\frac{\partial R(r,t)}{\partial r} + g(r,t) = \frac{1}{a}\frac{\partial R(r,t)}{\partial t} \tag{3.22}$$

where $0 < r < L,\ 0 < t < T_{max}.$ and $g(r,t)$ *is heat generation*, subject to the boundary conditions:

$$R(0\,,t\,) = b_1, R(\,L,t\,) = b_n$$

and initial condition:

$$R(\,r\,,0\,)\ = f(r)$$

then by (FDM), we have:

$$\frac{R(r_{i+1},t_j)-2R(r_i,t_j)+R(r_{i-1},t_j)}{(\Delta r)^2} + \frac{1}{r_i}\ \frac{R(r_{i+1}\,,t_j)-R(r_i,t_j)}{\Delta r}$$

$$+O(\Delta x)^2+g\left(r_i,t_j\right) = \frac{1}{a}\ \frac{R(r_i,t_{j+1})-R(r_i,t_j)}{\Delta t} + O(\Delta t) \tag{3.23}$$

where, $r_i = (i-1)\Delta r$, $i = 1, 2, 3, \dots, N$ and $t_j = (j-1)\Delta t$

$j = 1, 2, 3, \dots, M$, then the boundary and initial conditions becomes:

$$R\left(0,t_j\right) = b_1, R\left(L,t_j\right) = b_n\ , j = 1,2,3, \dots, M$$

and

$$R(r_i,0) =\ f(r_i), i\ \ = 1,2,3, \dots, N.$$

Solving equation (3.23) for approximate $R\left(r_i, t_{j+1}\right)$, we obtain:

$$R\left(r_i, t_{j+1}\right) = \mu\left(1+\frac{1}{(i-1)}\right)R\left(r_{i+1}, t_j\right) + \left(1-\mu-\frac{\mu}{(i-1)}\right)R\left(r_i, t_j\right)$$

$$+\mu R\left(r_{i-1}, t_j\right) + bg\left(r_i, t_j\right) \qquad (3.24)$$

Or

$$R\left(r_i, t_{j+1}\right) = \mu\left(\frac{i}{(i-1)}\right)R\left(r_{i+1}, t_j\right) + \mu R\left(r_{i-1}, t_j\right)$$

$$+\left(1-\mu-\frac{\mu i}{i-1}\right)R\left(r_i, t_j\right) + bg\left(r_i, t_j\right) \qquad (3.25)$$

where $\mu = \frac{a\Delta t}{(\Delta r)^2}$, $b = a\Delta t$, $i = 2,3,\dots,N-1$ and $j = 2,3,\dots,M$, this result with T_{error} = $(O(\Delta r)^2 + O(\Delta t))$.

3.3.3 Sphere:

Consider the heat conduction problem:

$$\frac{\partial^2 S(r,t)}{\partial r^2} + \frac{2}{r}\frac{\partial S(r,t)}{\partial r} + g(r,t) = \frac{1}{a}\frac{\partial S(r,t)}{\partial t} \qquad (3.26)$$

where $0 < r < L$, $0 < t < T_{max.}$ and $g(r,t)$ is *heat generation*, subject to the boundary conditions*:*

$$S(0,t) = b_1,\ S(L,t) = b_n$$

and initial condition*:*

$$S(r,0) = f(r)$$

then by use the finite difference method, we have:

$$\frac{S(r_{i+1},t_j)-2S(r_i,t_j)+S(r_{i-1},t_j)}{(\Delta r)^2} + O(\Delta r)^2 + g(r_i,t_j)$$
$$+ \frac{2}{r_i}\ \frac{S(r_{i+1}\,,t_j)-S(r_i,t_j)}{\Delta r} = \frac{1}{a}\ \frac{S(r_i\,,t_{j+1})-S(r_i\,,t_j)}{\Delta t} + O(\Delta t) \quad (3.27)$$

where $r_i = (i-1)\Delta r\ , i = 1,2,3,\dots,N$, and $t_j = (j-1)\Delta t$,

$j = 1,2,3,\dots,M$, the boundary and initial conditions becomes:

$$S(0,t_j) = b_1\ , S(L,t_j) = b_n\ , j = 1,2,3,\dots,M$$

and

$$S(r_i,0) = f(r_i)\ , i = 1,2,3,\dots,N$$

then solving equation (3.27) for approximate $S(r_i,t_{j+1})$, we get:

$$S(r_i,t_{j+1}) = \kappa(1+\tfrac{2}{i-1})\ S(r_{i+1},t_j) + \kappa\ \ S(r_{i-1},t_j) +$$
$$(1-2\kappa-\tfrac{2\kappa}{i-1})S(r_i,t_j) + bg(r_i,t_j) \quad (3.28)$$

or

$$S(r_i,t_{j+1}) = \kappa\left(\frac{i+1}{i-1}\right)S(r_{i+1},t_j) + (1-\kappa- \frac{\kappa(i+1)}{i-1}\ S(r_i,t_j) +$$
$$\kappa\, S(r_{i-1},t_j) + bg(r_i,t_j) \quad (3.29)$$

where $\kappa = \frac{a\Delta t}{(\Delta r)^2}$, $b = a\Delta t, i = 2,3,\dots,N-1$ *and* $j = 2,3,\dots,M$,

this result with $T_{error} = (O(\Delta r)^2 + O(\Delta t))$.

Note that: To determine *Truncation error;* T_{error}, when the *Exact solution* is known, we find *Exact solution* for any (x_i , t_j), then *Truncation error:*

$$\boldsymbol{T_{error} = |\, Exact\ solution - Approximate\ solution\, |} \qquad (3.30)$$

Note that: The FTCS method, see [17], for one-dimensional equations is numerically stable *if and only if* the following condition is satisfied:

$$\frac{a\Delta t}{(\Delta x)^2} \leq \frac{1}{2} \quad and \quad \frac{a\Delta t}{(\Delta r)^2} \leq \frac{1}{2} \qquad (3.31)$$

Chapter four

Numerical Examples

In this chapter, we will implement the finite difference method **FTCS** (Forward-Time Central-Space) to solve some heat conduction problems.

4.1 Plane Wall:

Example (4.1): Consider the homogeneous heat conduction problem:

$$\frac{\partial^2 T(x,t)}{\partial x^2} = \frac{1}{8}\frac{\partial T(x.t)}{\partial t} \qquad (4.1)$$

where $0 < x < 5$, $0 < t < t_{max.}$.

Subject to BC's: $T(0,t) = 10$, $T(5,t) = 90$

and IC: $T(x,0) = 16x+10+2\,sin(\pi x)-4sin(2\pi x)+sin(6\pi x)$

we have the *Exact solution*, see [17]:

$$T(x,t) = 16x + 10 + 2e^{-8\pi^2 t}\sin(\pi x) - 4e^{-32\pi^2 t}\sin(2\pi x) + e^{-288\pi^2 t}\sin(6\pi x) \qquad (4.2)$$

then by using equation (3.21), we have:

$$T(x_i, t_{j+1}) = \lambda T(x_{i+1}, t_j) + (1 - 2\lambda)T(x_i, t_j) + \lambda T(x_{i-1}, t_j) \qquad (4.3)$$

where $\lambda = \frac{a\Delta t}{(\Delta x)^2}$, $i = 2,3, \dots, N-1$, $j = 2,3, \dots, M$

subject to BC's: $T(x_1, t_j) = 10, T(x_N, t_j) = 90$, $j = 1,2,3, \dots, M$

and IC: $T(x_i, t_1) =$

$$16x_i + 10 + 2\,sin(\pi x_i) - 4sin(2\pi x_i) + sin(6\pi x_i)$$

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

The *exact solution* at each grid point is given by:

$$T(x_i, t_j) = 16x_i + 10 + 2e^{-8\pi^2 t_j}\sin(\pi x_i) - 4e^{-32\pi^2 t_j}\sin(2\pi x_i) + e^{-288\pi^2 t_j}\sin(6\pi x_i) \qquad (4.4)$$

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

We use C^{++} *language* to solve equation (4.3), we get tables (4.1), (4.2) and (4.3).

Note that: To simplify we will write *apprximate* $T(x_i, t_j) = T_{appx.}(i,j)$ and *Exact* $T(x_i, t_j) = E.T(i,j)$.

Table (4.1): Numerical results for example (4.1) with

N= 5 , M= 49 , $T_{max.}$= 4,Δt = 0.0833 , Δx = 1.25 and λ = 0.4267.

i	*j*	x_i	t_j	$T_{appx.}(i,j)$	$E.T(i,j)$	$T_{error}(i,j)$
:						
2	10	1.2500	0.7500	30.001834738	30.000000000	0.001834738
2	11	1.2500	0.8333	29.999162206	30.000000000	0.000837794
2	12	1.2500	0.9167	30.000382833	30.000000000	0.000382833
2	13	1.2500	1.0000	29.999825	30.000000000	0.000174705
Table (4.1): Numerical results for example (4.1) with λ = *0.426*						

i	j	x_i	t_j	$T_{appx.}(i,j)$	$E.T(i,j)$	$T_{error}(i,j)$
2	14	1.2500	1.0833	30.000080	30.000000000	0.000079904
2	15	1.2500	1.1667	29.999964	30.000000000	0.000036411
2	16	1.2500	1.2500	30.000017	30.000000000	0.000016693
2	17	1.2500	1.3333	29.999992	30.000000000	0.000007577
2	18	1.2500	1.4167	30.000003	30.000000000	0.000003496
: :						
3	25	2.5000	2.0000	50.000000025	50.000000000	0.000000025
3	26	2.5000	2.0833	49.999999994	50.000000000	0.000000006
3	27	2.5000	2.1667	50.000000007	50.000000000	0.000000007
3	28	2.5000	2.2500	50.000000000	50.000000000	0.000000000
3	29	2.5000	2.3333	50.000000002	50.000000000	0.000000002
3	30	2.5000	2.4167	50.000000001	50.000000000	0.000000001
3	31	2.5000	2.5000	50.000000000	50.000000000	0.000000001
: :						

Table (4.1): Numerical results for example (4.1) with $\lambda = 0.4267$

Table (4.2): Numerical results for example (4.1) with

$N = 6$, $M = 261$, $T_{max} = 4$, $\Delta t = .0154$, $\Delta x = 1$ and $\lambda = 0.1231$.

i	j	x_i	t_j	$T_{appx.}(i,j)$	$E.T(i,j)$	$T_{error}(i,j)$
: :						
2	135	1.0000	2.0615	26.000000	26.000000000	0.000000008
2	136	1.0000	2.0769	26.000000	26.000000000	0.000000007
2	137	1.0000	2.0923	26.000000	26.000000000	0.000000007

Table (4.2): Numerical results for example (4.1) with $\lambda = 0.1231$

i	j	x_i	t_j	$T_{appx.}(i,j)$	$E.T(i,j)$	$T_{error}(i,j)$
2	138	1.0000	2.1077	26.000000	26.000000000	0.000000007
2	139	1.0000	2.1231	26.000000	26.000000000	0.000000006
2	140	1.0000	2.1385	26.000000	26.000000000	0.000000006
2	141	1.0000	2.1538	26.000000	26.000000000	0.000000006
: :						
3	25	2.5000	2.0000	50.000000025	50.000000000	0.000000025
3	26	2.5000	2.0833	49.999999994	50.000000000	0.000000006
3	27	2.5000	2.1667	50.000000007	50.000000000	0.000000007
3	28	2.0000	0.4154	41.999999998	42.000000000	0.000002111
3	29	2.0000	0.4308	41.999999998	42.000000000	0.000002013
3	30	2.0000	0.4462	41.999999998	42.000000000	0.000001920
3	31	2.0000	0.4615	41.999999998	42.000000000	0.000001831
3	32	2.0000	0.4769	41.999999998	42.000000000	0.000001745
3	33	2.0000	0.4923	41.999999998	42.000000000	0.000001664
3	34	2.0000	0.5077	41.999999998	42.000000000	0.000001586
3	35	2.0000	0.5231	41.999999998	42.000000000	0.000001512
: :						
4	90	3.00	1.3692	58.0000000	58.000000000	0.000000107
4	91	3.00	1.3846	58.0000000	58.000000000	0.000000102
4	92	3.00	1.4000	58.0000000	58.000000000	0.000000097
4	93	3.00	1.4154	58.0000000	58.000000000	0.000000093
4	94	3.00	1.4308	58.0000000	58.000000000	0.000000088
4	95	3.00	1.4462	58.0000000	58.000000000	0.000000084
4	96	3.0000	1.4615	58.0000000	58.000000000	0.000000080
4	97	3.0000	1.4769	58.0000000	58.000000000	0.000000077
4	98	3.0000	1.4923	58.0000000	58.000000000	0.000000073
4	99	3.0000	1.5077	58.0000000	58.000000000	0.000000070

Table (4.2): Numerical results for example (4.1) with $\lambda = 0.123$

i	j	x_i	t_j	$T_{appx.}(i,j)$	$E.T(i,j)$	$T_{error}(i,j)$
: :						
5	53	4.0000	0.8000	74.000000	74.000000000	0.000000394
5	54	4.0000	0.8154	74.000000	74.000000000	0.000000375
5	55	4.0000	0.8308	74.000000	74.000000000	0.000000358
5	56	4.0000	0.8462	74.000000	74.000000000	0.000000341
5	57	4.0000	0.8615	74.000000	74.000000000	0.000000325
5	58	4.0000	0.8769	74.000000	74.000000000	0.000000310
5	59	4.0000	0.8923	74.000000	74.000000000	0.000000295
5	60	4.0000	0.9077	74.000000	74.000000000	0.000000281
5	61	4.0000	0.9231	74.000000	74.000000000	0.000000268
5	62	4.0000	0.9385	74.000000	74.000000000	0.000000255
5	63	4.0000	0.9538	74.000000	74.000000000	0.000000243
5	64	4.0000	0.9692	74.000000	74.000000000	0.000000232
5	65	4.0000	0.9846	74.000000	74.000000000	0.000000221
: :						

Table (4.2): Numerical results for example (4.1) with $\lambda = 0.1231$

Table (4.3): Numerical results for example (4.1) with

$N = 10$, $M = 9$, $T_{max.} = 10$, $\Delta t=1.25$, $\Delta x = 0.5556$ ***and*** $\lambda = 32.4$

i	j	x_i	t_j	$T_{appx.}(i,j)$	$E.T(i,j)$	$T_{error}(i,j)$
2	2	0.5556	1.2500	-373.436685	18.888888889	392.325574138
2	3	0.5556	2.5000	37931.849367	18.888888889	37912.960477670
2	4	0.5556	3.7500	-3901330.180	18.888888889	3901349.0698076
2	5	0.5556	5.0000	424466780.59	18.888888889	424466761.709

Table (4.3): Numerical results for example (4.1) with $\lambda = 32.4$

i	*j*	x_i	t_j	$T_{appx.}(i,j)$	$E.T(i,j)$	$T_{error}(i,j)$
2	6	0.5556	6.2500	-48274350151	18.888888889	48274350170.0
: :						
4	2	1.6667	1.2500	-200.287807	36.666666667	236.954473540
4	3	1.6667	2.5000	39262.3390	36.666666667	39225.6723988
4	4	1.6667	3.7500	-5617550.7	36.666666667	5617587.338244
4	5	1.6667	5.0000	754990994	36.666666667	754990957.8125
4	6	1.6667	6.2500	-98233421484	36.666666667	98233421520.7
: :						

Table (4.3): Numerical results for example (4.1) with λ = *32.4*

Note that: Exact solution and approximate solution in tables (4.1) and (4.2) are very close agreement with $\frac{a\Delta t}{(\Delta x)^2} \leq \frac{1}{2}$ and table (4.3) is not close agreement with $\frac{a\Delta t}{(\Delta x)^2} > \frac{1}{2}$.

Example (4.2): Consider the homogeneous heat conduction problem:

$$\frac{\partial^2 T(x,t)}{\partial x^2} = \frac{\partial T(x.t)}{\partial t} \tag{4.5}$$

where $0 < x < \pi , 0 < t < t_{max.}$

subject to BC's: $T(0,t) = T(\pi,t) = 0$

and IC: $T(x,0) = 4\sin(x) + 2\sin(2x) + 7\sin(3x)$

where the *Exact solution* is, see [5]:

$$T(x,t) = 4e^{-t}\sin(x) + 2\,e^{-4t}\sin(2x) + 7e^{-9t}\sin(3x) \qquad (4.6)$$

We use C^{++} *language* to solve equation (4.3), we get tables (4.4), (4.5) and (4.6).

Table(4.4): Numerical results for example (4.2) with

$\boldsymbol{N = 5, T_{max.} = 3, M = 12, \Delta t = 0.\overline{27}}$, ***Δx= 0.7854 and λ = 0.442***

i	j	x_i	t_j	$T_{appx.}(i,j)$	$E.T(i,j)$	$T_{error}(i,j)$
: :						
3	5	1.5708	1.0909	0.734215	1.343263	0.609047831
3	6	1.5708	1.3636	1.134046	1.022884	0.111162191
3	7	1.5708	1.6364	0.539724	0.778744	0.239019531
3	8	1.5708	1.9091	0.553108	0.592860	0.039752230
3	9	1.5708	2.1818	0.331814	0.451345	0.119530504
3	10	1.5708	2.4545	0.285641	0.343609	0.057967899
3	11	1.5708	2.7273	0.191401	0.261590	0.070188421
3	12	1.5708	3.0000	0.152153	0.199148	0.046995324
4	1	2.3562	0.0000	5.778178	5.778178	0.000000000
4	2	2.3562	0.2727	-0.657605	1.906655	2.564259984
4	3	2.3562	0.5455	2.811289	1.450147	1.361141553
4	4	2.3562	0.8182	0.492987	1.175327	0.682340148
4	5	2.3562	1.0909	1.186022	0.924905	0.261117030
4	6	2.3562	1.3636	0.461891	0.714781	0.252889863
4	7	2.3562	1.6364	0.554855	0.547786	0.007069008
4	8	2.3562	1.9091	0.302848	0.418250	0.115402399

Table (4.4): Numerical results for example (4.2) with λ = *0.44*

i	j	x_i	t_j	$T_{appx.}(i,j)$	$E.T(i,j)$	$T_{error}(i,j)$
4	9	2.3562	2.1818	0.279597	0.318825	0.039227410
4	10	2.3562	2.4545	0.179066	0.242859	0.063793336
4	11	2.3562	2.7273	0.147016	0.184935	0.037919508
4	12	2.3562	3.0000	0.101640	0.140807	0.039166836
5	1	3.1416	0.0000	0.000000	-0.000007	0.000007275
5	2	3.1416	0.2727	0.000000	-0.000001	0.000001214
5	3	3.1416	0.5455	0.000000	-0.000001	0.000000700
5	4	3.1416	0.8182	0.000000	-0.000001	0.000000563
5	5	3.1416	1.0909	0.000000	-0.000000	0.000000448
5	6	3.1416	1.3636	0.000000	-0.000000	0.000000348
5	7	3.1416	1.6364	0.000000	-0.000000	0.000000268
5	8	3.1416	1.9091	0.000000	-0.000000	0.000000205
5	9	3.1416	2.1818	0.000000	-0.000000	0.000000156
5	10	3.1416	2.4545	0.000000	-0.000000	0.000000119
5	11	3.1416	2.7273	0.000000	-0.000000	0.000000091
5	12	3.1416	3.0000	0.000000	-0.000000	0.000000069

Table (4.4): Numerical results for example (4.2) with $\lambda = 0.442$

Table(4.5): Numerical results for example (4.2) with $N = 10, T_{max.} = 4, M = 80, \Delta t = 0.038$, $\Delta x = 0.349$ and $\lambda = 0.312$

i	j	x_i	t_j	$T_{appx.}(i,j)$	$E.T(i,j)$	$T_{error}(i,j)$
: :						
2	1	0.3491	0.0000	8.715834	8.715834	0.000000000
2	2	0.3491	0.0380	6.587602	6.728747	0.141144837
2	3	0.3491	0.0759	5.077469	5.277128	0.199658827

Table (4.5): Numerical results for example (4.2) with $\lambda=0.312$

i	j	x_i	t_j	$T_{appx.}(i,j)$	$E.T(i,j)$	$T_{error}(i,j)$
2	4	0.3491	0.1139	3.997862	4.210237	0.212374066
2	5	0.3491	0.1519	3.218987	3.420418	0.201430512
2	6	0.3491	0.1899	2.650917	2.830705	0.179788915
2	7	0.3491	0.2278	2.231249	2.386003	0.154754208
: :						
4	1	1.0472	0.0000	5.196150	5.196150	0.000000000
4	2	1.0472	0.0380	4.813350	4.822980	0.009629865
4	3	1.0472	0.0759	4.472279	4.489018	0.016739094
4	4	1.0472	0.1139	4.167381	4.189238	0.021856315
4	5	1.0472	0.1519	3.893891	3.919301	0.025409758
4	6	1.0472	0.1899	3.647720	3.675465	0.027745219
4	7	1.0472	0.2278	3.425355	3.454496	0.029140931
4	8	1.0472	0.2658	3.223781	3.253601	0.029819876
4	9	1.0472	0.3038	3.040402	3.070362	0.029959946
4	10	1.0472	0.3418	2.872986	2.902689	0.029702320
4	11	1.0472	0.3797	2.719611	2.748769	0.029158363
4	12	1.0472	0.4177	2.578618	2.607034	0.028415309
4	13	1.0472	0.4557	2.448578	2.476119	0.027540915
4	14	1.0472	0.4937	2.328255	2.354842	0.026587282
4	15	1.0472	0.5316	2.216578	2.242172	0.025593973
4	16	1.0472	0.5696	2.112622	2.137213	0.024590565
: :						
6	70	1.7453	2.6203	0.280040	0.286695	0.006655510
6	71	1.7453	2.6582	0.269514	0.276014	0.006499940
6	72	1.7453	2.6962	0.259384	0.265731	0.006346639
6	73	1.7453	2.7342	0.249635	0.255831	0.006195661
6	74	1.7453	2.7722	0.240252	0.246299	0.006047054
6	75	1.7453	2.8101	0.231222	0.237122	0.005900856

Table (4.5): Numerical results for example (4.2) with $\lambda=0.312$

Table(4.6): Numerical results for example (4.2) with

$\boldsymbol{N = 10, T_{max.} = 5, M = 10, \Delta t=0.0556,\ \Delta x=0.349}$ ***and*** $\boldsymbol{\lambda= 4.56}$

i	j	x_i	t_j	$T_{appx.}(i,j)$	$E.T(i,j)$	$T_{error}(i,j)$
: :						
2	10	0.3491	5.0000	-556101	0.009218	556101
3	1	0.6981	0.0000	10.603	10.602943	0.0000
3	2	0.6981	0.5556	-22.653	1.729496	24.383
3	3	0.6981	1.1111	79.857	0.869810	78.987
3	4	0.6981	1.6667	-276.02	0.488136	276.51
3	5	0.6981	2.2222	976.47	0.278902	976.1
: :						
4	4	1.0472	1.6667	-2.206016	0.656489	2.862504238
4	5	1.0472	2.2222	3.000135	0.375637	2.624498148
4	6	1.0472	2.7778	-3.174407	0.215412	3.389818217
4	7	1.0472	3.3333	3.761217	0.123581	3.637636250
4	8	1.0472	3.8889	-7.528046	0.070904	7.598949404
: :						
Table (4.6): Numerical results for example (4.2) with $\lambda= 4.56$						

Note that: Exact solution and approximate solution in tables (4.4) and (4.5) are very close agreement with $\frac{a\Delta t}{(\Delta x)^2} \leq \frac{1}{2}$ and table (4.6) is not close agreement with $\frac{a\Delta t}{(\Delta x)^2} > \frac{1}{2}$.

Example (4.3): Consider the nonhomogeneous heat conduction problem:

$$\frac{\partial^2 T(x,t)}{\partial x^2} + g(x,t) = \frac{\partial T(x.t)}{\partial t} \tag{4.7}$$

where $0 < x < 1 , 0 < t < t_{max.}$ and $g(x,t) = e^{-t}\sin(\pi x)$.

subject to BC's: $T(0,t) = 0 , T(1,t) = 0$

and IC: $T(x,0) = f(x) = \sin(2\pi x)$

where the *Exact solution* is :

$T(x,t) =$

$$\frac{e^{-\pi^2 t}}{1-\pi^2}\sin(\pi x) - \frac{e^{-t}}{(1-\pi^2)}\sin(\pi x) + e^{-4\pi^2 t}\sin(2\pi x) \tag{4.8}$$

then by using equation (3.21), we have:

$$T(x_i, t_{j+1}) = \lambda T(x_{i+1}, t_j) + (1-2\lambda)T(x_i, t_j)$$

$$+\lambda T(x_{i-1}, t_j) + bg(x_i, t_j) \tag{4.9}$$

where $\lambda = \frac{a\Delta t}{(\Delta x)^2}$, $b = a\Delta t,\ i = 2,3, \dots, N-1\ , j = 2,3, \dots, M,$

$g(x_i, t_j) = e^{-t_j}\sin(\pi x_i)\ , i = 1,2, \dots, N\ , j = 1,2, \dots, M$

subject to BC's: $T(x_1, t_j) = T(x_N, t_j) = 0 , j = 1,2,3, \dots, M$

and IC: $T(x_i, t_1) = f(x_i) = \sin(2\pi x_i) , i = 2,3, \dots, N-1$

where the *Exact solution:*

$$T(x_i, t_j) =$$

$$\frac{e^{-\pi^2 t_j}}{1-\pi^2}\sin(\pi x_i) - \frac{e^{-t_j}}{(1-\pi^2)}\sin(\pi x_i) + e^{-4\pi^2 t_j}\sin(\pi 2 x_i) \quad (4.10)$$

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

We use C^{++} *language* to solve equation (4.9), we get tables (4.7), (4.8) and (4.9).

Table (4.7): Numerical results for example (4.3) with

N= 5 , M= 101 , $T_{max.}$=3 ,Δt =0.03 , Δx = 0.25 , and λ =0.48 .

i	***j***	x_i	t_j	$T_{appx.}(i,j)$	***E.T(i,j)***	$T_{error}(i,j)$
: :						
3	8	0.5000	0.2100	0.084820	0.077199	0.007620423
3	9	0.5000	0.2400	0.085288	0.078135	0.007152988
3	10	0.5000	0.2700	0.084906	0.078218	0.006687387
3	11	0.5000	0.3000	0.083934	0.077686	0.006247436
3	12	0.5000	0.3300	0.082558	0.076714	0.005844319
3	13	0.5000	0.3600	0.080912	0.075431	0.005481590
3	14	0.5000	0.3900	0.079092	0.073933	0.005158501
: :						
Table(4.7): Numerical results for example (4.3) with λ =*0.48*						

i	j	x_i	t_j	$T_{appx.}(i,j)$	$E.T(i,j)$	$T_{error}(i,j)$
4	22	0.7500	0.6300	0.044818	0.042301	0.002517619
4	23	0.7500	0.6600	0.043514	0.041086	0.002427864
4	24	0.7500	0.6900	0.042243	0.039899	0.002344181
4	25	0.7500	0.7200	0.041005	0.038740	0.002265657
4	26	0.7500	0.7500	0.039801	0.037610	0.002191557
4	27	0.7500	0.7800	0.038630	0.036509	0.002121285
4	28	0.7500	0.8100	0.037493	0.035438	0.002054363
4	29	0.7500	0.8400	0.036387	0.034397	0.001990403
4	30	0.7500	0.8700	0.035314	0.033385	0.001929094
4	31	0.7500	0.9000	0.034272	0.032402	0.001870183
4	32	0.7500	0.9300	0.033260	0.031447	0.001813462
: :						
4	90	0.7500	2.6700	0.005838	0.005521	0.000317387
4	91	0.7500	2.7000	0.005666	0.005358	0.000308007
4	92	0.7500	2.7300	0.005498	0.005199	0.000298904
4	93	0.7500	2.7600	0.005336	0.005046	0.000290070
4	94	0.7500	2.7900	0.005178	0.004897	0.000281497
4	95	0.7500	2.8200	0.005025	0.004752	0.000273177
4	96	0.7500	2.8500	0.004877	0.004611	0.000265104
4	97	0.7500	2.8800	0.004732	0.004475	0.000257269
4	98	0.7500	2.9100	0.004593	0.004343	0.000249665
4	99	0.7500	2.9400	0.004457	0.004215	0.000242287
4	100	0.7500	2.9700	0.004325	0.004090	0.000235126
4	101	0.7500	3.0000	0.004197	0.003969	0.000228177
: :						

Table(4.7): Numerical results for example (4.3) with $\lambda = 0.48$

Table (4.8): Numerical results for example (4.3) with

$N = 11, M = 601, T_{max.} = 3$, Δt= .005, Δx = 0.1 , and λ = 0.5.

i	j	x_i	t_j	$T_{appx.}(i,j)$	$E.T(i,j)$	$T_{error}(i,j)$
: :						
4	302	0.3000	1.5050	0.020431	0.020251	0.000180633
4	303	0.3000	1.5100	0.020329	0.020150	0.000179732
4	304	0.3000	1.5150	0.020228	0.020049	0.000178836
4	305	0.3000	1.5200	0.020127	0.019949	0.000177943
4	306	0.3000	1.5250	0.020027	0.019850	0.000177056
4	307	0.3000	1.5300	0.019927	0.019751	0.000176172
4	308	0.3000	1.5350	0.019827	0.019652	0.000175293
4	309	0.3000	1.5400	0.019729	0.019554	0.000174419
4	310	0.3000	1.5450	0.019630	0.019457	0.000173549
4	311	0.3000	1.5500	0.019532	0.019360	0.000172683
4	312	0.3000	1.5550	0.019435	0.019263	0.000171822
4	313	0.3000	1.5600	0.019338	0.019167	0.000170964
4	314	0.3000	1.5650	0.019241	0.019071	0.000170112
4	315	0.3000	1.5700	0.019146	0.018976	0.000169263
: :						
10	370	0.9000	1.8450	0.005555	0.005506	0.000049112
10	371	0.9000	1.8500	0.005527	0.005478	0.000048867
10	372	0.9000	1.8550	0.005499	0.005451	0.000048624
10	373	0.9000	1.8600	0.005472	0.005424	0.000048381
10	374	0.9000	1.8650	0.005445	0.005397	0.000048140
10	375	0.9000	1.8700	0.005418	0.005370	0.000047900

Table (4.8): Numerical results for example (4.3) with λ = *0.5*

i	j	x_i	t_j	$T_{appx.}(i,j)$	$E.T(i,j)$	$T_{error}(i,j)$
10	376	0.9000	1.8750	0.005391	0.005343	0.000047661
10	377	0.9000	1.8800	0.005364	0.005316	0.000047423
10	378	0.9000	1.8850	0.005337	0.005290	0.000047187
10	79	0.9000	0.3900	0.023098	0.022846	0.000251019
10	80	0.9000	0.3950	0.023013	0.022765	0.000248561
10	81	0.9000	0.4000	0.022928	0.022682	0.000246151
10	82	0.9000	0.4050	0.022841	0.022597	0.000243787
10	83	0.9000	0.4100	0.022754	0.022512	0.000241468
10	84	0.9000	0.4150	0.022666	0.022426	0.000239194
10	85	0.9000	0.4200	0.022577	0.022340	0.000236964
10	86	0.9000	0.4250	0.022487	0.022252	0.000234777
10	87	0.9000	0.4300	0.022396	0.022164	0.000232631
10	88	0.9000	0.4350	0.022305	0.022075	0.000230526
10	89	0.9000	0.4400	0.022214	0.021985	0.000228461
10	90	0.9000	0.4450	0.022122	0.021895	0.000226436
10	91	0.9000	0.4500	0.022029	0.021804	0.000224448
10	92	0.9000	0.4550	0.021936	0.021713	0.000222498
10	93	0.9000	0.4600	0.021843	0.021622	0.000220584
10	94	0.9000	0.4650	0.021749	0.021530	0.000218706
10	95	0.9000	0.4700	0.021655	0.021438	0.000216863
10	96	0.9000	0.4750	0.021561	0.021346	0.000215054
10	97	0.9000	0.4800	0.021466	0.021253	0.000213278
10	98	0.9000	0.4850	0.021372	0.021160	0.000211535
10	99	0.9000	0.4900	0.021277	0.021067	0.000209823
10	100	0.9000	0.4950	0.021182	0.020974	0.000208141
: :						

Table (4.8): Numerical results for example (4.3) with $\lambda = 0.5$

Table (4.9): Numerical results for example (4.3) with

$N = 5, M = 21, T_{max} = 5, \Delta t = 0.25$, $\Delta x = 0.25$, *and* $\lambda = 4$

i	j	x_i	t_j	$T_{appx.}(i,j)$	$E.T(i,j)$	$T_{error}(i,j)$
: :						
2	2	0.2500	0.2500	-6.82	0.055379	6.878603454
2	3	0.2500	0.5000	48.9	0.047781	48.85247550
2	4	0.2500	0.7500	-342.759	0.037610	342.7966406
2	5	0.2500	1.0000	2400.76	0.029324	2400.733337
2	6	0.2500	1.2500	-16806.65	0.022841	16806.67469
2	7	0.2500	1.5000	117649.03	0.017788	117649.0170
2	8	0.2500	1.7500	-823548	0.013854	823548.7415
2	9	0.2500	2.0000	5764881	0.010789	5764881.13
2	10	0.2500	2.2500	-40354631	0.008403	40354631.29
2	11	0.2500	2.5000	282488227	0.006544	282488227.6
2	12	0.2500	2.7500	-1977491035	0.005096	1977491035.9
2	13	0.2500	3.0000	13843366700	0.003969	13843366700
2	14	0.2500	3.2500	-96915330664	0.003091	96915330664
: :						
Table (4.9): Numerical results for example (4.3) with $\lambda = 4$						

Note that: The exact solution and approximate solutions in tables (4.7) and (4.8) are in a close agreement, with $\frac{a\Delta t}{(\Delta x)^2} \leq \frac{1}{2}$. However, the results in table (4.9) are not close agreement, with $\frac{a\Delta t}{(\Delta x)^2} > \frac{1}{2}$.

4.2 Cylinder: Consider heat conduction problem:

$$\frac{\partial^2 R(r,t)}{\partial r^2} + \frac{1}{r}\frac{\partial R(r,t)}{\partial r} + g(r,t) = \frac{1}{a}\frac{\partial R(r,t)}{\partial t} \qquad (4.11)$$

where $0 < r < L$, $0 < t < T_{max.}$ and $g(r,t)$ *is heat generation*,

subject to the boundary conditions:

$$R(0,t) = b_1, R(L,t) = b_n$$

and initial condition:

$$R(r,0) = f(r)$$

then by using equation (3.25), we have:

$$R(r_i, t_{j+1}) = \mu\left(\frac{i}{(i-1)}\right)R(r_{i+1}, t_j) + \mu R(r_{i-1}, t_j)$$

$$+\left(1 - \mu - \frac{\mu i}{i-1}\right)R(r_i, t_j) + bg(r_i, t_j) \qquad (4.12)$$

where $\mu = \frac{a\Delta t}{(\Delta r)^2}$ $, b = a\Delta t, i = 2,3, \dots, N-1$ *and*

$j = 2,3, \dots, M$. This result with $T_{error} = (O(\Delta r)^2 + O(\Delta t))$.

Example(4.4): Consider equation (4.11) for the homogeneous case,

subject to BC's: $R(r_1, t_j) = 10, R(r_N, t_j) = 100, j = 2,3, \dots, M$

and IC: $R(r_i, t_1) = f(r_i) = 10 + sin(\pi r_i), i = 2, 3, \dots, N-1$

use equation (4.12), we have tables (4.10) and (4.11).

Table (4.10): Numerical results for example (4.4) with $N = 5, M = 5,$

$T_{max.} = 10, L = 10, a = 1, \Delta t$=2.5, Δr= 2.5 , and μ = 0.4.

i	j	r_i	t_j	$f(r_i)$	$R(i,j)$
1	1	0.0000	0.0000	10.000000	10.000000
1	2	0.0000	2.5000	10.000000	10.000000
1	3	0.0000	5.0000	10.000000	10.000000
1	4	0.0000	7.5000	10.000000	10.000000
1	5	0.0000	10.0000	10.000000	10.000000
2	1	2.5000	0.0000	11.000000	11.000000
2	2	2.5000	2.5000	11.000000	9.799999
2	3	2.5000	5.0000	11.000000	9.880000
2	4	2.5000	7.5000	11.000000	32.967999
2	5	2.5000	10.0000	11.000000	29.903467
3	1	5.0000	0.0000	9.999998	9.999998
3	2	5.0000	2.5000	9.999998	9.800000
3	3	5.0000	5.0000	9.999998	38.679999
3	4	5.0000	7.5000	9.999998	40.621333
3	5	5.0000	10.0000	9.999998	56.915022
4	1	7.5000	0.0000	9.000000	9.000000
4	2	7.5000	2.5000	9.000000	57.933333
4	3	7.5000	5.0000	9.000000	61.115556
4	4	7.5000	7.5000	9.000000	72.879703
4	5	7.5000	10.0000	9.000000	74.440514
5	1	10.0000	0.0000	0.000000	100.000000
5	2	10.0000	2.5000	0.000000	100.000000
5	3	10.0000	5.0000	0.000000	100.000000
5	4	10.0000	7.5000	0.000000	100.000000
5	5	10.0000	10.0000	0.000000	100.000000
Table (4.10): Numerical results for example (4.4) with μ = *0.4*					

Table (4.11): Numerical results for example (4.4) with $L = 7, a = 8,$

$T_{max.} = 4, N = 11,\ M = 261, \Delta t = 0.0154,\ \Delta r= 0.7$, and $\mu=0.251$

i	j	r_i	t_j	$f(r_i)$	$R(i,j)$
: :					
7	256	4.2000	3.9231	10.587786	85.130317
7	257	4.2000	3.9385	10.587786	85.133979
7	258	4.2000	3.9538	10.587786	85.137552
7	259	4.2000	3.9692	10.587786	85.141039
7	260	4.2000	3.9846	10.587786	85.144443
7	261	4.2000	4.0000	10.587786	85.147764
8	1	4.9000	0.0000	10.309015	10.309015
8	2	4.9000	0.0154	10.309015	10.017320
8	3	4.9000	0.0308	10.309015	9.995559
8	4	4.9000	0.0462	10.309015	12.035551
8	5	4.9000	0.0615	10.309015	14.885237
8	6	4.9000	0.0769	10.309015	17.981194
: :					
10	250	6.3000	3.8308	10.809018	96.880474
10	251	6.3000	3.8462	10.809018	96.881599
10	252	6.3000	3.8615	10.809018	96.882698
10	253	6.3000	3.8769	10.809018	96.883770
10	254	6.3000	3.8923	10.809018	96.884816
10	255	6.3000	3.9077	10.809018	96.885837
: :					

Table (4.11): Numerical results for example (4.4) with $\mu=0.251$

Example(4.5): Consider equation (4.11) for the nonhomogeneous case,

subject to BC's: $R(r_1, t_j) = R(r_N, t_j) = 0$, $j = 2,3, \dots , M$

and IC: $R(r_i, t_1) = f(r_i) = sin\left(\frac{\pi r_i}{3}\right), i = 2, 3, \dots , N - 1,$

where the $heat\ genaration\ g(r_i, t_j) = e^{-4\pi^2 t_j} \sin\left(\frac{\pi r_i}{3}\right),\ i = 1, 2, .., N,\ j = 1,2, \dots , M,$ use equation (4.12), we have tables (4.12) and (4.13).

Table (4.12): Numerical results for example (4.5) with $N = 5$, $M = 5$,

$T_{max.} = 10, L = 10, a = 1, \Delta t = 2.5$, $\Delta r = 2.5$, *and* $\mu = 0.4$.

i	j	r_i	t_j	$g(i,j)$	*f(r*$_i$*)*	*R(i,j)*
: :						
2	1	2.5000	0.0000	0.500000	0.500000	0.500000
2	2	2.5000	2.5000	0.000000	0.500000	0.457179
2	3	2.5000	5.0000	0.000000	0.500000	-1.183486
2	4	2.5000	7.5000	0.000000	0.500000	1.448718
2	5	2.5000	10.0000	0.000000	0.500000	-0.859503
3	1	5.0000	0.0000	-0.866025	-0.866025	-0.866025
3	2	5.0000	2.5000	-0.000000	-0.866025	-1.365063
3	3	5.0000	5.0000	-0.000000	-0.866025	1.515026
3	4	5.0000	7.5000	-0.000000	-0.866025	-0.712199
3	5	5.0000	10.0000	-0.000000	-0.866025	0.927173
4	1	7.5000	0.0000	1.000000	1.000000	1.000000
4	2	7.5000	2.5000	0.000000	1.000000	2.220257
4	3	7.5000	5.0000	0.000000	1.000000	-0.398008
: :						
Table (4.12): Numerical results for example (4.5) with $\mu = 0.4$						

Table (4.13): Numerical results for example (4.5) with $L = 5, a = 8$

$N = 6$, $M = 261, T_{max.} = 4$, Δt= 0.0154 ,Δr=1, and μ = 0.123

i	j	r_i	t_j	$g(i,j)$	$f(r_i)$	$R(i,j)$
: :						
5	12	4.0000	0.1692	-0.001086	-0.866026	-0.032845
5	13	4.0000	0.1846	-0.000592	-0.866026	-0.018457
5	14	4.0000	0.2000	-0.000322	-0.866026	-0.007575
5	15	4.0000	0.2154	-0.000176	-0.866026	0.000588
5	16	4.0000	0.2308	-0.000096	-0.866026	0.006644
5	17	4.0000	0.2462	-0.000052	-0.866026	0.011069
5	18	4.0000	0.2615	-0.000028	-0.866026	0.014234
5	19	4.0000	0.2769	-0.000015	-0.866026	0.016428
5	20	4.0000	0.2923	-0.000008	-0.866026	0.017874
5	21	4.0000	0.3077	-0.000005	-0.866026	0.018748
5	22	4.0000	0.3231	-0.000003	-0.866026	0.019186
5	23	4.0000	0.3385	-0.000001	-0.866026	0.019295
5	24	4.0000	0.3538	-0.000001	-0.866026	0.019158
5	25	4.0000	0.3692	-0.000000	-0.866026	0.018838
5	26	4.0000	0.3846	-0.000000	-0.866026	0.018386
5	27	4.0000	0.4000	-0.000000	-0.866026	0.017841
5	28	4.0000	0.4154	-0.000000	-0.866026	0.017233
5	29	4.0000	0.4308	-0.000000	-0.866026	0.016584
: :						

Table (4.13): Numerical results for example (4.5) with μ = *0.123*

4.3. Sphere

Consider heat conduction problem:

$$\frac{\partial^2 S(r,t)}{\partial r^2} + \frac{2}{r}\frac{\partial S(r,t)}{\partial r} + g(r,t) = \frac{1}{a}\frac{\partial S(r,t)}{\partial t} \qquad (4.13)$$

where $0 < r < L$, $0 < t < T_{max.}$ and $g(r,t)$ *is heat generation*, subject to the boundary conditions:

$$S(0,t) = b_1,\ S(L,t) = b_n$$

and initial condition: $S(r,0) = f(r)$

then by using equation (3.29), we have:

$$S(r_i, t_{j+1}) = \kappa\left(\frac{i+1}{i-1}\right)S(r_{i+1}, t_j) + (1-\kappa-\frac{\kappa(i+1)}{i-1})S(r_i, t_j)$$

$$+\kappa\, S(r_{i-1}, t_j) + bg(r_i, t_j) \qquad (4.14)$$

where $\kappa = \frac{a\Delta t}{(\Delta r)^2}$, $b = a\Delta t, i = 2,3, \dots, N-1$ *and* $j=2,3,\dots,M.$

This result with $T_{error} = (O(\Delta r)^2 + O(\Delta t))$.

Example (4.6): Consider equation (4.13) for the homogeneous case, subject to BC's:

$$S(r_1, t_j) = 5, S(r_N, t_j) = 15\ , j = 2,3, \dots, M$$

and IC: $S(r_i, t_1) = f(r_i) = 5 + sin(\pi r_i)$

where $i = 2, 3, \dots, N-1$, use equation (4.14), we have tables (4.14) and (4.15).

Table (4.14): Numerical results for example (4.6) with $T_{max.} = 10$,

$N = 5$, $M = 5$, $L = 10$, $a = 1$, $\Delta t = 2.5$, $\Delta r = 2.$ and $\kappa = 0.45$

i	j	r_i	t_j	$f(r_i)$	$R(i,j)$
1	1	0.0000	0.0000	5.000000	5.000000
1	2	0.0000	2.5000	5.000000	5.000000
1	3	0.0000	5.0000	5.000000	5.000000
1	4	0.0000	7.5000	5.000000	5.000000
1	5	0.0000	10.0000	5.000000	5.000000
2	1	2.5000	0.0000	6.000000	6.000000
2	2	2.5000	2.5000	6.000000	4.399998
2	3	2.5000	5.0000	6.000000	4.880002
2	4	2.5000	7.5000	6.000000	11.343997
2	5	2.5000	10.0000	6.000000	5.697070
3	1	5.0000	0.0000	4.999998	4.999998
3	2	5.0000	2.5000	4.999998	4.600000
3	3	5.0000	5.0000	4.999998	10.226665
3	4	5.0000	7.5000	4.999998	8.752890
3	5	5.0000	10.0000	4.999998	13.469806
4	1	7.5000	0.0000	4.000000	4.000000
4	2	7.5000	2.5000	4.000000	11.733333
4	3	7.5000	5.0000	4.000000	11.057778
4	4	7.5000	7.5000	4.000000	13.353481
4	5	7.5000	10.0000	4.000000	12.610924
5	1	10.0000	0.0000	0.000000	15.000000
5	2	10.0000	2.5000	0.000000	15.000000
5	3	10.0000	5.0000	0.000000	15.000000
5	4	10.0000	7.5000	0.000000	15.000000
5	5	10.0000	10.0000	0.000000	15.000000

Table (4.14): Numerical results for example (4.6) with $\kappa = 0.4$

Table (4.15): Numerical results for example (4.6) with $T_{max.} = 4$,

$N = 6, M = 261, L = 5, a = 8, \Delta t$=0.0154, Δr=1and κ = 0.123

i	j	r_i	t_j	$f(r_i)$	$R(i,j)$
: :					
3	258	2.0000	3.9538	5.000001	13.000000
3	259	2.0000	3.9692	5.000001	13.000000
3	260	2.0000	3.9846	5.000001	13.000000
3	261	2.0000	4.0000	5.000001	13.000000
4	1	3.0000	0.0000	4.999999	4.999999
4	2	3.0000	0.0154	4.999999	5.000000
4	3	3.0000	0.0308	4.999999	5.378698
4	4	3.0000	0.0462	4.999999	5.895282
4	5	3.0000	0.0615	4.999999	6.444860
4	6	3.0000	0.0769	4.999999	6.982270
4	7	3.0000	0.0923	4.999999	7.489387
4	8	3.0000	0.1077	4.999999	7.960413
4	9	3.0000	0.1231	4.999999	8.395042
4	10	3.0000	0.1385	4.999999	8.795253
4	11	3.0000	0.1538	4.999999	9.163812
4	12	3.0000	0.1692	4.999999	9.503596
4	13	3.0000	0.1846	4.999999	9.817309
4	14	3.0000	0.2000	4.999999	10.107391
4	15	3.0000	0.2154	4.999999	10.375999
: :					
Table (4.15): Numerical results for example (4.6) with κ = 0.123					

Example(4.7): Consider equation (4.13) for the nonhomogeneous case, subject to BC's: $S(r_1, t_j) = S(r_N, t_j) = 0\ , j = 2,3, \dots, M$

and IC: $S(r_i, t_1) = f(r_i) = sin\left(\frac{\pi r_i}{5}\right), i = 2, 3, \dots, N-1$

where the *heat genaration* $g(r_i, t_j) = e^{-t_j} \sin\left(\frac{\pi r_i}{5}\right)$,

$i = 1, 2, 3, \dots, N\ , j = 1,2,3, \dots, M$, use equation (4.14), we have tables (4.16) and (4.17).

Table (4.16): Numerical results for example (4.7) with $T_{max.} = 10$,

$N = 5, M = 5, L = 10, a = 1$, Δt=2.5 , Δr= 2.5 and κ = 0.4

i	*j*	r_i	t_j	$g(i,j)$	$f(r_i)$	*R(i,j)*
: :						
2	1	2.5000	0.0000	1.000000	1.000000	1.000000
2	2	2.5000	2.5000	0.082085	1.000000	1.900000
2	3	2.5000	5.0000	0.006738	1.000000	-1.414788
2	4	2.5000	7.5000	0.000553	1.000000	-0.462282
2	5	2.5000	10.0000	0.000045	1.000000	-0.329617
3	1	5.0000	0.0000	-0.000000	-0.000000	-0.000000
3	2	5.0000	2.5000	-0.000000	-0.000000	-0.400001
3	3	5.0000	5.0000	-0.000000	-0.000000	-1.106667
3	4	5.0000	7.5000	-0.000000	-0.000000	-0.506974
3	5	5.0000	10.0000	-0.000000	-0.000000	-0.440301
4	1	7.5000	0.0000	-1.000000	-1.000000	-1.000000
4	2	7.5000	2.5000	-0.082085	-1.000000	-2.433333

Table (4.16): Numerical results for example (4.7) with $\kappa = 0.4$

i	j	r_i	t_j	$g(i,j)$	$f(r_i)$	$R(i,j)$
4	3	7.5000	5.0000	-0.006738	-1.000000	-0.202991
4	4	7.5000	7.5000	-0.000553	-1.000000	-0.445979
4	5	7.5000	10.0000	-0.000045	-1.000000	-0.174441
5	1	10.0000	0.0000	0.000001	0.000000	0.000000
5	2	10.0000	2.5000	0.000000	0.000000	0.000000
5	3	10.0000	5.0000	0.000000	0.000000	0.000000
5	4	10.0000	7.5000	0.000000	0.000000	0.000000
5	5	10.0000	10.0000	0.000000	0.000000	0.000000

Table (4.16): Numerical results for example (4.7) with $\kappa = 0.4$

Table (4.17): Numerical results for example (4.7) with $T_{max.} = 4$, $N = 6, M = 261, L = 5, a = 8, \Delta t=0.0154$, $\Delta r=1$ and $\kappa= 0.123$

i	j	r_i	t_j	$g(i,j)$	$f(r_i)$	$R(i,j)$
: :						
2	254	1.0000	3.8923	0.011990	0.587785	0.043256
2	255	1.0000	3.9077	0.011807	0.587785	0.042596
2	256	1.0000	3.9231	0.011626	0.587785	0.041945
2	257	1.0000	3.9385	0.011449	0.587785	0.041305
2	258	1.0000	3.9538	0.011274	0.587785	0.040674
2	259	1.0000	3.9692	0.011102	0.587785	0.040053
2	260	1.0000	3.9846	0.010933	0.587785	0.039442
2	261	1.0000	4.0000	0.010766	0.587785	0.038840
3	1	2.0000	0.0000	0.951057	0.951057	0.951057
3	2	2.0000	0.0154	0.936537	0.951057	1.023399

Table (4.17): Numerical results for example (4.7) with $\kappa= 0.123$

i	j	r_i	t_j	$g(i,j)$	$f(r_i)$	$R(i,j)$
3	2	2.0000	0.0154	0.936537	0.951057	1.023399
3	3	2.0000	0.0308	0.922239	0.951057	1.094223
3	4	2.0000	0.0462	0.908159	0.951057	1.157886
3	5	2.0000	0.0615	0.894294	0.951057	1.213661
: :						
4	252	3.0000	3.8615	0.020006	0.951056	0.044641
4	253	3.0000	3.8769	0.019701	0.951056	0.043959
4	254	3.0000	3.8923	0.019400	0.951056	0.043288
4	255	3.0000	3.9077	0.019104	0.951056	0.042627
4	256	3.0000	3.9231	0.018812	0.951056	0.041976
4	257	3.0000	3.9385	0.018525	0.951056	0.041335
4	258	3.0000	3.9538	0.018242	0.951056	0.040704
4	259	3.0000	3.9692	0.017964	0.951056	0.040083
4	260	3.0000	3.9846	0.017689	0.951056	0.039471
4	261	3.0000	4.0000	0.017419	0.951056	0.038868
5	1	4.0000	0.0000	0.587785	0.587785	0.587785
5	2	4.0000	0.0154	0.578811	0.587785	0.596324
5	3	4.0000	0.0308	0.569975	0.587785	0.606366
5	4	4.0000	0.0462	0.561273	0.587785	0.616840
5	5	4.0000	0.0615	0.552704	0.587785	0.627225
5	6	4.0000	0.0769	0.544266	0.587785	0.637201
5	7	4.0000	0.0923	0.535957	0.587785	0.646546
5	8	4.0000	0.1077	0.527774	0.587785	0.655109
5	9	4.0000	0.1231	0.519717	0.587785	0.662790
5	10	4.0000	0.1385	0.511783	0.587785	0.669535

Table (4.17): Numerical results for example (4.7) with $\kappa = 0.123$

4.4 Conclusion

In this work, we have presented one of the most important topic in thermal engineering, namely; heat conduction and diffusion processes.

The main focus is to solve heat conduction problems in some specific domains. These include plane wall, cylinder and sphere.

Analytical methods involving separation of variables, Laplace transform, Duhamel and Green function methods have been introduced to solve these problems. For the numerical handling of heat conduction problems, we have implemented the finite difference method **(FTCS).** Numerical results have shown to be in a close agreement with the exact ones. In fact, we strongly believe that the **FTCS** is an efficient methods for solving these types of problems. On the other hand, we note that the exact and the approximate solutions are in very close agreement with the stability condition $\frac{a\Delta t}{(\Delta x)^2} \leq \frac{1}{2}$.

References

1- I. Alzeer and N. Qatanani, ***An efficient numerical solution for acoupled heat conduction and enclosure radiation***, International Journal of Engineering Mathematics, Vol. 2, 2007, 1-9.

2- I. Alzeer and N. Qatanani, ***Numerical simulation of the steady state heat conduction equation***, International Journal of Mathematical Modeling, Simulation and Applications, Vol.2, 1, 2008.

3- A. Barham, N. Qatanani and Q.Heeh, **Existence and uniqueness of the coupled conduction radiation energy transfer on diffuse-gray surfaces**, Surveys in Mathematics and its Applications, Vol.2 (2007), 43-58.

4- Y. Cengel and J. Cimbala, **Fluid Mechanics: Fundamentals and Applications**, the McGraw-Hill Companies, 2006.

5- Y. Cengel, **Heat Transfer,** a practical approach, The McGraw-Hill Companies, 2008.

6- K. Cheng and T. Fujii, **Heat Transfer Engineering,** University of Southampton, 2012.

7- D. Dewitt and F. Incropera, **Fundamentals of heat and mass transfer**, John Wiley & Sons, New York, 1985.

8- D. Hahn and M. Necati, **Heat Conduction,** New Jersey, 2012.

9- L. Jiji**, Heat Conduction,** Third Edition, New York, New York 10031, USA, 2009.

10- L. Kettle**, Numerical solution of partial differential Equations,** The University of Queensland, 2012.

11-R. LeVeque**, Finite Difference Methods for Differential Equations,** University of Washington**,** September, 2005.

12- J. Matthew**, The 1-D Heat Equation,** Linear Partial Differential Equations, Hancock, 2006.

13- M. Necati, **Heat Conduction,** second edition**,** North Carolina, 1993.

14- B. Neta, **Partial differential equations ma 3132 lecture notes,** Monterey, California 93943, October 16, 2012.

15- R. Pich'e, **Partial differential equations**, Tampere university of Tochnology, 2010.

16- G. Recktenwald**, Finite-Difference Approximations to the Heat Equation,** Portland State University**,** 2011.

17- G. Recktenwald, **FTCS Solution to the Heat Equation**, Me 448/548 Notes, Portland State University, 2010.

18- G. Rorrer, C. Wicks, R. Wilson and J. Welty, **Fundamentals of Momentum, Heat, and Mass Transfer,** United States of America**,** 2008.

19- Z. Tseng, **Second Order Linear Partial Differential Equations,** University of California, 2008.

20- L. Wang, X. Wei and X. Zhou, **Heat Conduction**, **Mathematical Models and Analytical Solutions**, Verlag Berlin Heidelberg, 2008.

21- I. Yanovsky, **Partial Differential Equations: Graduate Level Problems and Solutions,** Igor Yanovsky, 2005.

Appendix

Appendix A

C⁺⁺ code for example (4.1):

```
#define _USE_MATH_DEFINES // Define the value of pi
#include <stdio.h>
#include <math.h>
#define PI 3.141593
int main()
{
   printf("Example_3!\n\n");
   int N ,M ,Tmax;
   int L = 5 , a =8 ;
   printf("Enter these values \n");
   printf("N = ");
   scanf("%d",&N);
   printf("M = ");
   scanf("%d",&M);
   printf("Tmax = ");
   scanf("%d",&Tmax);
   printf("L = %d\n",L);
   printf("a = %d\n",a);
   double Xi; // Xi = (i-1) * delta_X
   double Tj; // Tj = (j-1) * delta_T
   double delta_X = (double) L/(N-1);
   double delta_T = (double) Tmax/(M-1);
   double Lamda = a * ((double) delta_T/pow(delta_X, 2.0));
   double R[N+1][M+2];
   double g[N][M];
   double Te[N][M];
```

```
printf("delta_X = %.4lf\n",delta_X);
printf("delta_T = %.4lf\n",delta_T);
printf("Lamda = %.4lf\n",Lamda);
// At j = 1
for(int i = 2 ; i<N ; i++)
{
    Xi = (i-1) * delta_X;
    R[i][1] = 10 + 16 * Xi + 2 * sin(PI * Xi) - 4 * sin(2 * PI * Xi) + sin(PI * Xi) ;
}
// At i = N OR i = 1 then
for (int j = 1 ; j<=M+1 ; j++)
{
    R[1][j] = 10 ;
    R[N][j] = 90 ;
}
for(int j = 1 ; j<=M ; j++)
{
    for(int i = 2 ; i<N ; i++)
        R[i][j+1] = Lamda * R[i-1][j] + (1-2*Lamda) * R[i][j] + Lamda * R[i+1][j];
}
printf("i      j        Xi           Tj                    T(i,j)                  ET(i,j)      Te(i,j)\n");
printf("-----------------------------\n");
for(int i = 1 ; i<=N ; i++)
{
    for(int j = 1 ; j<=M ; j++)
    {
        Tj = (j-1) * delta_T;
```

```
        Xi = (i-1) * delta_X;
        // find the function of G
        double Exp1 = -8  * pow(PI, 2.0) * Tj ;
        double Exp2 = -32 * pow(PI, 2.0) * Tj ;
        double Exp3 = -288 * pow(PI, 2.0) * Tj ;
        g[i][j] = 10 + 16 * Xi + 2 * exp(Exp1) * sin(PI * Xi) - 4 *
exp(Exp2) * sin(2 * PI * Xi) + exp(Exp3) * sin(PI * Xi) ;
        //
        // Find the Error
        Te[i][j] = g[i][j] - R[i][j] ;
                abs(g[i][j] - R[i][j] );
        printf("%d    %d    %.4lf    %.4lf      %lf       %.9lf
%.9f\n",i,j,Xi,Tj,R[i][j],g[i][j],Te[i][j]);
        printf("----------------------------\n");
      }
   }
   return 0;
```

Appendix B

C++ code for example (4.2):

```
#define _USE_MATH_DEFINES // Define the value of pi
#include <stdio.h>
#include <math.h>
#define PI 3.141593
int main() {
    // insert code here...
    printf("Example_2!\n");
    int N ,M ,Tmax  ;
    int a = 1;
    double L = PI;
    printf("Enter these values \n");
    printf("N = ");
    scanf("%d",&N);
    printf("M = ");
    scanf("%d",&M);
    printf("Tmax = ");
    scanf("%d",&Tmax);
    printf("L = %lf\n",L);
    printf("a = %d\n",a);
    double delta_X = (double) L/(N-1);
    double delta_T = (double) Tmax/(M-1);
    double Lamda = a * ((double) delta_T/pow(delta_X, 2.0));
    double R[N+1][M+2];
    double g[N][M];
    double Te[N][M];
    double Xi;
    printf("delta_X = %.4lf\n",delta_X);
```

```
printf("delta_T = %.4lf\n",delta_T);
printf("Lamda = %.4lf\n",Lamda);
// At j = 1
for(int i = 2 ; i<N ; i++)
{
    Xi = (i-1) * delta_X;
    R[i][1] = 4*sin(Xi) + 2*sin(2*Xi) + 7*sin(3*Xi);
}
// At i = N OR i = 1 then R = 0
for (int j = 1 ; j<=M+1 ; j++)
    R[1][j] = R[N][j] = 0.0;
for(int j = 1 ; j<=M ; j++)
{
    for(int i = 2 ; i<N ; i++)
        R[i][j+1] = Lamda * R[i-1][j] + (1-2*Lamda) * R[i][j] + Lamda *
R[i+1][j] ;
}
printf("i    j     Xi       Tj        T(i,j)       ET(i,j)       Te(i,j)\n");
printf("----------------\n");
for(int i = 1 ; i<=N ; i++)
{
    for(int j = 1 ; j<=M ; j++)
    {
        double Tj = (j-1) * delta_T;
        double Xi = (i-1) * delta_X;
        g[i][j] = 4*sin(Xi)*exp(-1*Tj) + 2*sin(2*Xi)*exp(-4*Tj) +
7*sin(3*Xi)*exp(-9*Tj);
        Te[i][j] = g[i][j] - R[i][j] ;
                abs(g[i][j] - R[i][j] );
        printf("%d    %d    %.4lf    %.4lf     %lf      %lf
```

```
%.9f\n",i,j,Xi,Tj,R[i][j],g[i][j],Te[i][j]);
            printf("---\n");
        }
    }
    return 0;
}
```

Appendix C

C^{++} code for example (4.3):

```
#define _USE_MATH_DEFINES // Define the value of pi
#include <stdio.h>
#include <math.h>
#define PI 3.141593
int main() {
    // insert code here...
    printf("Example_5!\n");
    int N ,M ,Tmax  ;
    int a = 1;
    int L = 1;
    printf("Enter these values \n");
    printf("N = ");
    scanf("%d",&N);
    printf("M = ");
    scanf("%d",&M);
    printf("Tmax = ");
    scanf("%d",&Tmax);
    printf("L = %d\n",L);
    printf("a = %d\n",a);
    double delta_X = (double) L/(N-1);
    double delta_T = (double) Tmax/(M-1);
    double Lamda = a * ((double)delta_T/pow(delta_X, 2.0));
    double b = a * delta_T;
    double R[N+1][M+1];
    double g[N+1][M+1];
    double Texact[N][M];
    double Terorr[N][M];
```

```
double Xi,Tj;
printf("delta_X = %.4lf\n",delta_X);
printf("delta_T = %.4lf\n",delta_T);
printf("Lamda = %.4lf\n",Lamda);
// At j = 1
for(int i = 2 ; i<N ; i++)
{
    Xi = (i-1) * delta_X;
    R[i][1] = sin(2*PI*Xi);
}
// At i = N OR i = 1 then R = 0
for (int j = 1 ; j<=M ; j++)
    R[1][j] = R[N][j] = 0.0;
//Function OF G(Xi,Tj)
for(int i=1 ; i<=N ; i++)
{
    for(int j=1 ; j<=M ; j++)
    {
        Tj = (j-1) * delta_T;
        Xi = (i-1) * delta_X;
        double ex = -1 * Tj;
        g[i][j] = exp(ex) * sin(PI * Xi);
    }
}
printf("\n\n\n");
//Whole Function
for(int j = 1 ; j<=M ; j++)
{
    for(int i = 2 ; i<N ; i++)
```

```
      {
         R[i][j+1] = Lamda * R[i-1][j] + (1-2*Lamda) * R[i][j] +Lamda *
R[i+1][j] + b * g[i][j] ;}
   }
   printf("i    j     Xi      Tj       T(i,j)       Tex(i,j)      Terr(i,j)\n");
   printf("\n");
   //Exact Function
   for(int i=1 ; i<=N ; i++)
   {
      for(int j=1 ; j<=M ; j++)
      {
         Tj = (j-1) * delta_T;
         Xi = (i-1) * delta_X;
         double exp1 = -1 * pow(PI, 2.0) * Tj;
         double exp2 = -4 * pow(PI, 2.0) * Tj;
Texact[i][j] = ((1.0/(1-pow(PI, 2.0))) * exp(exp1) * sin(PI * Xi)) - ((exp(-
1*Tj)/(double)(1-pow(PI, 2.0))) * sin(PI*Xi)) + ((exp(exp2)) * sin(2 * PI
*Xi)) ;
         Terorr[i][j] = Texact[i][j] - R[i][j];
                  abs ( Texact[i][j] - R[i][j] );
         printf("%d    %d    %.4lf    %.4lf     %lf      %lf
%.9f\n",i,j,Xi,Tj,R[i][j],Texact[i][j],Terorr[i][j]);
         printf("\n");
      }
   }
   return 0;
}
```

Appendix D

C^{++} code for example (4.4):

```
#define _USE_MATH_DEFINES // Define the value of pi
#include <stdio.h>
#include <math.h>
#define PI 3.141593
int main() {
    // insert code here...
    printf("Cylinder Example_1!\n");
    int N ,M ,Tmax  ;
    double a,L;
    printf("Enter these values \n");
    printf("N = ");
    scanf("%d",&N);
    printf("M = ");
    scanf("%d",&M);
    printf("Tmax = ");
    scanf("%d",&Tmax);
    printf("L = ");
    scanf("%lf",&L);
    printf("a = ");
    scanf("%lf",&a);
    double delta_R = (double) L/(N-1);
    double delta_T = (double) Tmax/(M-1);
    double Mue = a * ((double) delta_T/pow(delta_R, 2.0));
    //double b = a * delta_R;
    printf("delta_R = %lf\n",delta_R);
    printf("delta_T = %lf\n",delta_T);
    printf("Mue = %lf\n",Mue);
```

```
double Ri,Tj;
double R[N+1][M+2];
double F[N+1];
// At j = 1
for(int i = 1 ; i<N ; i++)
{
    Ri = (i-1) * delta_R;
    R[i][1] = F[i] = 10 + sin(PI *Ri);
}
// At i = N OR i = 1 then R = 0
for (int j = 1 ; j<=M+1 ; j++)
{
    R[1][j] = F[1] = 10;
    R[N][j] = 100;
}
for(int j = 1 ; j<=M ; j++)
{
    for(int i = 2 ; i<N ; i++)
        R[i][j+1] = Mue * ((double)i/(i-1)) * R[i+1][j] + Mue * R[i-1][j] +
(1-Mue-(Mue * i/(double)(i-1))) * R[i][j];
}
printf("i    j     Ri      Tj          F[ri]        R(i,j)\n");
printf("\n");
for(int i = 1 ; i<=N ; i++)
{
    for(int j = 1 ; j<=M ; j++)
    {
        Tj = (j-1) * delta_T;
        Ri = (i-1) * delta_R;
```

```
        printf("%d    %d    %.4lf    %.4lf    %lf
%lf\n",i,j,Ri,Tj,F[i],R[i][j]);
        printf("\n");
      }
    }
    return 0;
}
```

Appendix E

C^{++} code for example (4.5):

```
#define _USE_MATH_DEFINES // Define the value of pi
#include <stdio.h>
#include <math.h>
#define PI 3.141593
int main() {
   // insert code here...
   printf("Cylinder Example_2!\n");
   int N ,M ,Tmax  ;
   double a,L;
   printf("Enter these values \n");
   printf("N = ");
   scanf("%d",&N);
   printf("M = ");
   scanf("%d",&M);
   printf("Tmax = ");
   scanf("%d",&Tmax);
   printf("L = ");
   scanf("%lf",&L);
   printf("a = ");
   scanf("%lf",&a);
   double delta_R = (double) L/(N-1);
   double delta_T = (double) Tmax/(M-1);
   double Mue = a * ((double) delta_T/pow(delta_R, 2.0));
   double b = a * delta_T;
   printf("delta_R = %lf\n",delta_R);
   printf("delta_T = %lf\n",delta_T);
   printf("Mue = %lf\n",Mue);
```

```
double Ri,Tj;
double R[N+1][M+2];
double F[N+1];
double G[N+1][M+1];
   // At j = 1
for(int i = 1 ; i<=N ; i++)
{
   Ri = (i-1) * delta_R;
   R[i][1] = F[i] = sin(PI * Ri / 3.0);
}
// At i = N OR i = 1 then R = 0
for (int j = 1 ; j<=M+1 ; j++)
   R[1][j] = R[N][j] = F[1] = F[N] =0;
//Find G[i][j]
for(int j = 1 ; j<=M ; j++)
{
   for(int i = 1 ; i<=N ; i++)
   {
      Ri = (i-1) * delta_R;
      Tj = (j-1) * delta_T;
      G[i][j] = exp(-4 * pow(PI, 2.0) *Tj) * sin(PI * Ri / 3.0);
   }
}
for(int j = 1 ; j<=M ; j++)
{
   for(int i = 2 ; i<N ; i++)
      R[i][j+1] = Mue * ((double)i/(i-1)) * R[i+1][j] + Mue * R[i-1][j] +
(1-Mue-(Mue * i/(double)(i-1))) * R[i][j] + b * G[i][j];
}
```

```
printf("i    j      Ri           Tj            G[i][j]      F[ri]        R(i,j)\n");
printf("--------\n");
for(int i = 1 ; i<=N ; i++)
{
    for(int j = 1 ; j<=M ; j++)
    {
        Tj = (j-1) * delta_T;
        Ri = (i-1) * delta_R;
        printf("%d    %d      %.4lf       %.4lf          %lf      %lf
%lf\n",i,j,Ri,Tj,G[i][j],F[i],R[i][j]);
        printf("-----\n");
    }
    return 0;
}
```

Appendix F

C^{++} code for example (4.6):

```
#define _USE_MATH_DEFINES // Define the value of pi
#include <stdio.h>
#include <math.h>
#define PI 3.141593
int main() {
    // insert code here...
    printf("Sphare Example_1!\n");
    int N ,M ,Tmax  ;
    double a,L;
    printf("Enter these values \n");
    printf("N = ");
    scanf("%d",&N);
    printf("M = ");
    scanf("%d",&M);
    printf("Tmax = ");
    scanf("%d",&Tmax);
    printf("L = ");
    scanf("%lf",&L);
    printf("a = ");
    scanf("%lf",&a);
    double delta_R = (double) L/(N-1);
    double delta_T = (double) Tmax/(M-1);
    double K = a * ((double) delta_T/pow(delta_R, 2.0));
    //double b = a * delta_T;
    double Ri,Tj;
    double S[N+1][M+1];
    double F[N+1];
```

```
// At j = 1
for(int i = 1 ; i<N ; i++)
{
    Ri = (i-1) * delta_R;
    S[i][1] = F[i] = 5 + sin(PI*Ri);
}
// At i = N OR i = 1 then R = 0
for (int j = 1 ; j<=M ; j++)
{
    S[1][j] = F[1] = 5;
    S[N][j] = 15;
}
for(int j = 1 ; j<M ; j++)
{
    for(int i = 2 ; i<N ; i++)
        S[i][j+1] = K * ((double)(i+1)/(i-1)) * S[i+1][j] + K * S[i-1][j] + (1-
K-(K * (i+1)/(double)(i-1))) * S[i][j];
}
printf("i    j      Ri          Tj          F[ri]        S(i,j)\n");
printf("\n");
for(int i = 1 ; i<=N ; i++)
{
    for(int j = 1 ; j<=M ; j++)
    {
        Tj = (j-1) * delta_T;
        Ri = (i-1) * delta_R;
        printf("%d    %d      %.4lf       %.4lf        %lf

%lf\n",i,j,Ri,Tj,F[i],S[i][j]);
```

```
        printf("\n");
      }
  }
  return 0;
}
```

Appendix G

C^{++} code for example (4.7):

```
#define _USE_MATH_DEFINES // Define the value of pi
#include <stdio.h>
#include <math.h>
#define PI 3.141593
int main() {
    // insert code here...
    printf("Sphare Example_2!\n");
    int N ,M ,Tmax  ;
    double a,L;
    printf("Enter these values \n");
    printf("N = ");
    scanf("%d",&N);
    printf("M = ");
    scanf("%d",&M);
    printf("Tmax = ");
    scanf("%d",&Tmax);
    printf("L = ");
    scanf("%lf",&L);
    printf("a = ");
    scanf("%lf",&a);
    double delta_R = (double) L/(N-1);
    double delta_T = (double) Tmax/(M-1);
    double K = a * ((double) delta_T/pow(delta_R, 2.0));
    double b = a * delta_T;
    double Ri,Tj;
```

```
double G[N+1][M+1];
double S[N+1][M+1];
double F[N+1];
// At j = 1
for(int i = 1 ; i<N ; i++)
{
   Ri = (i-1) * delta_R;
   S[i][1] = F[i] = sin(PI * Ri /5.0);
}
// At i = N OR i = 1 then R = 0
for (int j = 1 ; j<=M ; j++)
{
   S[1][j] = S[N][j] = 0;
}
// Find Function Of G
for(int j = 1 ; j<=M ; j++)
{
   for(int i = 1 ; i<=N ; i++)
   {
      Tj = (j-1) * delta_T;
      Ri = (i-1) * delta_R;
      G[i][j] = exp(-1*Tj) * sin(PI * Ri / 5.0) ;
   }
}
for(int j = 1 ; j<M ; j++)
{
   for(int i = 2 ; i<N ; i++)
```

```
            S[i][j+1] = K * ((double)(i+1)/(i-1)) * S[i+1][j] + K * S[i-1][j] + (1-K-(K * (i+1)/(i-1))) * S[i][j] + b * G[i][j];
    }
    printf("i    j       Ri          Tj          G[i][j]       F[ri]       S(i,j)\n");
    printf("-\n");
    for(int i = 1 ; i<=N ; i++)
    {
        for(int j = 1 ; j<=M ; j++)
        {
            Tj = (j-1) * delta_T;
            Ri = (i-1) * delta_R;
            printf("%d    %d      %.4lf       %.4lf        %lf      %lf      %lf\n",i,j,Ri,Tj,G[i][j],F[i],S[i][j]);
            printf("\n");
        }
    }
    return 0;
}
```

جامعة النجاح الوطنية

كلية الدراسات العليا

الطرق التحليلية والعددية لحل معادلة التوصيل الحراري

إعداد

عبدالله عدوان عبدالله نصار

إشراف

أ.د. ناجي قطناني

قدمت هذه الأطروحة استكمالا لمتطلبات الحصول على درجة الماجستير في الرياضيات بكلية الدراسات العليا في جامعة النجاح الوطنية، نابلس–فلسطين.

2017

الطرق التحليلية والعددية لحل معادلة التوصيل الحراري

إعداد

عبدالله عدوان عبدالله نصار

إشراف

أ. د. ناجي قطناني

الملخص

الكثير من الظواهر الفيزيائية والهندسية تظهر على شكل معادلات تفاضلية جزئية تصفها، وفي هذه الرسالة أخذنا معادلة التوصيل الحراري في عدة أوساط هي: السطح المستوي، والأسطوانة، والكرة, وقمنا بكتابة الصيغة الرياضية لهذه المعادلة، ثم قمنا بحل هذه الصيغ بعدة طرق تحليلية وفق شروط حدودية معينة وهذه الطرق هي:

Separation Of Variables Method (طريقة فصل المتغيرات), Laplace Transform Method (طريقة تحويل لابلاس) , Duhamel's Method

(طريقة دوهمل), and Green's Function Method (طريقة اقتران غرين).

ثم قمنا بحل معادلة التوصيل الحراري عدديا معتمدين على الطريقة:

Finite Difference Method (طريقة الفروق الحدودية)

وطبقناها على مجموعة من الأمثلة وأوجدنا حلولا تقريبية لها اعتمادا على لغة البرمجة $\mathbf{C}^{++}$ ، وخرجنا منها ببعض النتائج التي تؤكد على أن هذه الطريقة مقدار الخطأ(T_{error}) فيها يكون قليل جدا عند مقارنة الحل الحقيقي عند نفس القيم مع الحل التقريبي مع الأخذ بعين الاعتبار أن:

$$\frac{a\Delta t}{(\Delta x)^2} \leq \frac{1}{2} \quad \textit{and} \quad \frac{a\Delta t}{(\Delta r)^2} \leq \frac{1}{2}$$

ويكون مقدار الخطأ غير معقول في حال إهمال هذا الشرط .