ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

THE INFLUENCE OF NONLINEAR REACTION, MEMORY AND A HEAVISIDE FUNCTION SOURCE TERM ON SCALAR TRANSPORT

By

Habtamu Bekele Nora

A Thesis Submitted In (Partial) Fulfillment of the Requirements for the Degree of Masters of Science (Computational Science) of Addis Ababa University

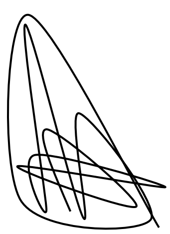
Prof… Okey Oseloka Onyejekwe
(Advisor)

June, 2016

Addis Ababa University

Declaration

I declare that this thesis is my own work, except where otherwise acknowledged. It has not been submitted before for any degree or examination purpose at any other institution.



……………………………….. 26/07/2016

Habtamu Bekele(student) Date

Certification

The undersigned certify that they have read and hereby recommend for acceptance by the University of Addis Ababa a thesis entitled "the influence of nonlinear reaction, memory and a Heaviside function source term on scalar transport" in partial fulfillment of the requirements for the degree of Masters of Science (Computational Science) of the University of Addis Ababa.

Prof. Okey Oseloka Onyejekwe
(Advisor)

....

Signature Date

THE INFLUENCE OF NONLINEAR REACTION, MEMORY AND A HEAVISIDE FUNCTION SOURCE TERM ON SCALAR TRANSPORT

By: Habtamu Bekele

APPROVED BY BOARD OF EXAMINERS

CHAIRMAN, DEPARTMENT

GRADUATE COMMITTEE

Prof Okey Oseloka Onyejekwe (Advisor)	Signature	Date
Dr. Lemi Demeyu (External examiner)	Signature	Date
Dr. Tsegaye Gedif (External examiner)	Signature	Date
Dr. Tilahun Abebaw (Chairman)	Signature	Date

JUNE, 2016
ADDIS ABABA UNIVERISTY

Abstract

In this study, a numerical method based on finite difference is presented for the numerical solution of a generalized Fisher-Integro differential equation. A composite weighted trapezoidal rule is manipulated to handle the numerical integrations which results in a closed difference scheme. The non-linear terms are linearized by one of the finite difference linearization techniques. Three different methods are used; Left end point rule, right end point rule and trapezoidal end point rule. The numerical solutions obtained for the three methods indicate that, the approach is reliable and yields results compatible. The discretization of the governing equation is made by explicit, Implicit and Crank-Nicolson method time scheme.

The flow of an incompressible viscous fluid between a uniformly porous upper plate and a lower impermeable plate that is subjected to a FKPP is modeled and analyzed in this study. The Model equations are presented in terms of Left end point rule, Right end point rule and trapezoidal end point rule. For the Left end point rule, we are using the explicit method, for the right end point rule we are using the implicit method and for the trapezoidal end point rule we are using both implicit and explicit (Crank-Nicolson) method.

In this study, the researcher looked how to discretize integro-partial differential equation (memory term) using trapezoidal, left endpoints, right endpoints and Simpson's rule. The literature highlights how the Fisher Kolomogrov-Petrovskii-Piskunov Equation is developed and used. The main part of this paper is dedicated in discretizing FKPP and developing a computer program to compute the solution.

Acknowledgements

Firstly, I would like to thank the Almighty God whose mercy and power has enabled me to conduct this study. Then, I would like to express my sincere thanks and deep gratitude to my advisor, Professor Okey Oseloka Onyejekwe, for his assistance, keen interest and supervisory guidance on my research. I feel so blessed to have him as my advisor because he was very helpful to an extent that I find the research part of my Masters degree more enjoyable. His generosity has inspired me to help others especially my students and be approachable to other people who will be in need of my support without looking at material benefits. Many thanks go to Dr. Tilahun Abebaw, the head of computational science department at Addis Ababa University for his guidance, fatherly care and encouragement during the course of study.

I am very indebted to my lovely wife, Workinesh Adamu, for all the patience and support she has provided throughout the course of my studies. Her readiness and willingness to trust me and take care of the family in my absence is something that I can never pay her back, but only promise her my unconditional love as a sign of appreciation. Moreover, there is no way I can explain my academic success without mentioning my parents. It has been my pleasure to associate my academic performance with my father Bekele Nora, especially my mother Rude Muleta (her love, care, patience and hospitality) and my brothers and sisters, Shitaye Bekele, Anbessa Bekele, Mulualem Bekele, Lemesa Bekele and Azalech Bekele, who helped me ideally and financially throughout this difficult journey of my loneliness throughout the entire period of my study. Together with them, I would like to recognize the contributions made by my friends Dejene Abdisa, Kefyalew Muleta, Damte Eba, Wagaye Decha, Zemen Weldu and others .

Lastly, I also wish to convey my sincere gratitude to all lecturers who taught me during the coursework part; and the entire mathematics department of the University for creating a conductive environment for me to learn effectively. I am also very grateful and I will always recognize the contribution made by Fitsum Sintayehu, pedagogical teaching skills, inspired me to like and learn Mathematics! I truly appreciate the contributions of my fellow MSc mathematical modeling students, Henok Serste and Getinet Melaku, who through their valuable discussions, kindness and hospitality made my graduate studies pleasant and helped me to experience a tough but winning academic journey. May God bless you!

Table of Contents

List of Figures

Abbreviations

Symbol	Description
u	The dependent variable (The fluid flow)
λ	The lunch velocity(Lambda), ($\lambda = -0.5$)
μ	The Diffusion coefficient (Micro),($\mu = 0.1$)
$\frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds$	The memory term
D	The constant parameter ($D = 0.1$)
τ	The constant parameter (Tao), ($\tau = 0.01$)
$u(1-u)$	The Fisher nonlinear term
λu_x	The convection term
μu_{xx}	The diffusion term
u_t	The transient term
$(T_w - u).H(x)$	The Heaviside function source term

CHAPTER ONE

INTRODUCTION

1.1 Background

The theory and application of integro-differential equation play an important role in the mathematical modeling of many fields [2]: physical and biological phenomena and engineering sciences in which it is necessary to take into account the effect of real world problems. Classically, a heat conduction phenomenon is represented by a parabolic partial differential equation with an infinite heat propagation speed; this is a puzzling contradiction to the physics.

Modeling is the process of representing or describing a real problem by using mathematical language and symbols. The development of mathematical models not only stimulates new experiments but it also allows us to understand the connections between different elements from an unstructured set of observations. The phenomenon of slip flow has attracted the attention of a large number of scholars due to its wide ranging application especially in this era of modern science, technology and vast ranging industrialization [16]. In many practical applications, the particle adjacent to a moving solid surface no longer takes the velocity of the surface; instead, the particle at the surface seems to have a finite tangential velocity, so it slips along the surface. Such flow is called the slip flow regime [9].

Fluid dynamics and transport phenomena, such as heat and mass transfer, play a vitally important role in human life [20]. Gases and liquids surround us, flow inside our bodies, and have a profound influence on the environment in which we live. Fluid flows produce winds, rains, floods, and hurricanes [21]. Convection and diffusion are responsible for temperature fluctuations and transport of pollutants in air, water or soil [1]. The ability to understand, predict, and control transport phenomena is essential for many industrial applications, such as aerodynamic shape design, oil recovery from an underground reservoir, or multiphase/multi component flows in furnaces, heat exchangers, and chemical reactors [22].

This ability offers substantial economic benefits and contributes to human well-being. Heating, air conditioning, and weather forecast have become an integral part of our everyday life. We take such things for granted and hardly ever think about the physics and mathematics behind them [4].

The traditional approach to investigation of a physical process is based on observations, experiments, and measurements. The amount of information that can be obtained in this way is usually very limited and subject to measurement errors. Moreover, experiments are only possible when a small-scale model or the actual equipment has already been built. An experimental investigation may be very time consuming, dangerous, prohibitively expensive, or impossible for another reason. Computational study can be performed on the basis of a suitable mathematical model [24]. As a rule, such a model consists of several differential and/or algebraic equations which make it possible to predict how the quantities of interest evolve and interact with one another.

A drawback to this approach is the fact that complex physical phenomena give rise to complex mathematical equations that cannot be solved analytically, i.e., using paper and pencil. The most detailed models of fluid flow are based on 'first principles', such as the conservation of mass, momentum, and energy [25]. Mathematical equations that embody these fundamental principles have been known for a very long time but used to be practically worthless until numerical methods and digital computers were invented. The second half of the twentieth century has witnessed the advent of Computational Fluid Dynamics (CFD), a new branch of applied mathematics that deals with numerical simulation of fluid flows [9]. Nowadays, computer codes based on CFD models are used routinely to predict a variety of increasingly complex flow phenomena.

The quality of simulation results depends on the choice of the model and on the accuracy of the numerical method. In spite of the inevitable numerical and modeling errors, approximate solutions may provide a lot of valuable information at a fraction of the cost that a full-scale experimental investigation would require. Moreover, the sampling of relevant data is free of errors due to a flow disturbance caused by probe. A further advantage of the computational approach is the fact that it can be applied to flows in domains with arbitrarily large or small dimensions under realistic operating conditions. High pressures, toxic chemicals or hot temperatures pose no hazard to a CFD practitioner.

Last but not least, simultaneous computation of instantaneous density, velocity, pressure, temperature, and concentration fields is feasible [26]. Clearly, no experimental technique can capture the evolution of all flow variables throughout the domain. However, experiments are still required to determine the values of input parameters for a mathematical model and to validate the computational results. The choice of a CFD model is dictated by the nature of the physical process to be simulated, by the objectives of the numerical study, and by the available resources. As a rule of thumb, the mathematical model should be as detailed as possible without making the computations too expensive. In many cases, the desired information can be obtained using a simplified version that exploits some *a* priori knowledge of the flow pattern or incorporates empirical correlations supported by theoretical or experimental studies [27]. Thus, a hierarchy of fundamental, phenomenological, and empirical models is usually available for particularly difficult problems, such as the numerical simulation of turbulence.

Over the past three decades, the market for CFD software has expanded rapidly, and remarkable progress has been made in the development of numerical algorithms. An astonishing variety of finite difference, finite element, finite volume, and spectral schemes were developed for the equations of fluid mechanics and applied to virtually every flow problem of practical importance [28]. Modern CFD codes are equipped with automatic mesh generation/adaptation tools, reliable error control mechanisms, and efficient iterative solvers for sparse linear systems. Unstructured mesh methods are available for flows in complex geometries [29].

The main function of irrigation canal is to deliver water in an accurate and flexible way. The delivery is said to be accurate if the actual supply matches the intended supply, and it is said to be flexible if the delivery meets the changing water requirements of the users. This main function can be translated into a water level control problem consisting of two parts: First, the water levels in the system located just upstream of the off takes and control structures need to be controlled within a sufficiently small range. Second, adjusting the control structure located at the upstream end of the canal to control the preferably water levels. This requirements guarantees that the delivery matches the demands, one of the most useful control strategies that satisfy this requirement is the downstream end of pool water level control in real-time.

The control algorithm should be mathematically simple in order to require small computing effort. Various control algorithms have been proposed in order to reduce the height of dikes in downstream controlled systems. A number of control methods have been proposed for maintaining a constant water level at the downstream end of each pool. For the application/calibration purposes for the presented model in this research; an existing irrigation project channel's data has been used. For this case, the principle in building the model is to internalize as many variables as possible so that the model is nearly closed.

Human life depends on water daily, especially for drinking and food production. Also, human life needs to be protected against excess of water caused by heavy precipitation and floods. People have formed water management organizations to guarantee these necessities of life for communities. These organizations manage a water system within the community and manipulate the water flows in this system to fulfill the water related requirements. To do so, controllable structures, such as gates and pumps are used. The way these structures are controlled, depending on the requirements of the communities, is part of the research field of control on water systems, often referred to as operational water management.

In the research 'Model Predictive Control on Open Water Systems', the relatively new control methodology Model Predictive Control is configured for application of water quantity control on open water systems, especially on irrigation canals and large drainage systems. The methodology applies an internal model of the open water system, by which optimal control actions are calculated over a prediction horizon.

As internal model, two simplified models are used, the Integrator Delay model and the Saint Venant model. Kalman filtering is applied to initialize the internal models. The optimization uses an objective function in which conflicting objectives can be weighed. In most of the cases, these conflicting objectives are keeping the water levels at different locations in the water system within a range around set point and executing this by using as little control effort or energy as possible. To tune the weight factors in the objective function, an estimate of the maximum allowed value of each variable in the objective function is used.

The optimization takes the constraints of the control structures into account. Every control time step, the optimal control actions are calculated, while only the first set of control actions is actually executed. This results in a controlled water system that is constantly maintaining the objective in an optimal way, while taking predictions, such as expected irrigation demands or extreme storm events, and the constraints of the water system into account. To show the potential of Model Predictive Control in controlling water systems, it is compared to the classical control methods Feedback Control and Feed forward Control. This comparison shows that Feedback Control has the lowest performance, as it first requires a deviation from set point to actually start the control actions. Adding Feed forward Control improves the performance. Many water systems are subject to constrained controllability of the structures. For example, pumps have a limited pump capacity and the flow through gates can be limited by the (sea) water level next to the structure. Model predictive control takes these constraints into account while calculating the optimal control actions.

For that reason, model predictive control outperforms feedback control and feed forward control in periods of extreme load. In other periods, the performance of model predictive control is at least comparable to the performance of feedback control in combination with feed forward control. Another advantage of model predictive control is the ability of dealing with conflicting objectives. In the objective function, relative weights are given to the different objectives, resulting in a well-balanced set of control actions for the total water system and potential model predictive control on open Water Systems P.J van over loop vi problems, such as an excess or lack of water, can be dispersed throughout the system as much as possible. The optimization used in this research is suitable for linear internal models. As the water flows in canals and the structure flows are non-linear, a step-wise linearization is used in the optimization. By applying a number of iteration steps, the step-wise linearization approaches the non-linear solution with sufficient accuracy. In this way, all non-linear objects in a water system, even strongly nonlinear structures such as pumps switching off and on, can be taken into account in the optimization.

An extension to the standard model predictive controller is applied, by which uncertainties in predictions and models can be dealt with. Instead of optimizing one model, three parallel models are used in the internal model of the optimization. One model represents the average, most probable case. The other two models correspond to the best and the worst case. By multiplying the outcome of the three models by their probability of occurrence, the risk of high water levels is minimized, instead of minimizing high water level for just one of the possible cases. This stochastic configuration of Model Predictive Control is referred to as Multiple Model Predictive Control. Model Predictive Control and its derived configurations are applied to accurate models of open water systems and on actual irrigation canals and drainage systems in real-time.

The results show a clear improvement compared to the classical control methods. As the controller is set up in a generic way, it can easily be adapted to other water related fields, such as water quality control or waterpower generation and on other types of water systems, such as reservoirs and sewer systems. Finally, it is important that the solution to a control problem has to be as simple as the requirements on the controlled water system, the characteristics of the water system and the constraints of the structures allow it to be. In many cases, local Feedback Controllers have sufficient performance. In other cases, the application of the more complex Model Predictive Controller is unavoidable. A selection procedure for the most appropriate controller is part of this research.

1.1 Research Objectives

1.2.1. General Objective

The general objective of this study is dedicated for seeking a numerical solution of the generalized Fisher Kolomogrov-Petrovskii-Piskunov (FKPP), equation subject to specified initial and boundary conditions by using the left-end point rule, right-end point rule and trapezoidal endpoint rule of numerical integration which is used in paper [8] and demonstrate their effects on fluid flow.

1.2.2. Specific objectives

The following are the specific objectives of this study.

- The effects of memory term to fluid flow.
- The effect of linear reaction term to fluid flow.
- The effects of nonlinear (quadratic) term to fluid flow.
- The effect of a Heaviside function source term to fluid flow.

1.3 Methodology

The methodologies used in this research are:

- Descritization of the governing equation is made by Left, Right and Trapezoidal rule.
- The computer simulation has been made ,
- Graphically analyzed by MATLAB plot platforms.

1.4 Control of water systems

Management of open water systems can be formalized in a set of logical and mathematical rules within a controller. In the past, the different types of controllers have been categorized in their water management related characteristics. One way of categorization is:

- Flow control. A certain flow rate is imposed at a structure by changing the structure settings, such as gate width or gate opening;
- Volume control. The volume in a canal reach is kept as close to a target volume as possible. This type of control can have benefits when the management of the canal reaches is subject to frequent shut downs in which the canal transits from steady flow to zero flow. As the volume cannot be measured directly, more than one water level in the reach is measured, for example at the upstream and downstream end of the reach. From these water levels, the volume of water in the reach can be estimated by some weighing formula. This control method does not differ fundamentally from the next type of control;

- Water level control. The water levels in canal reaches are kept as close to a target water level as possible. This type of control is researched in this dissertation.

Another way of categorization is based on the location of the water level that has to be kept at target level relative to the control structure:

- Downstream control. The water level downstream of the control structure is kept as close to target level as possible. By applying this method, shortage of water in the downstream canal reach is replenished by extra inflow through the upstream control structure. This property makes downstream control highly suitable for control of irrigation systems. Note that the controlled location can be chosen further downstream in the canal reach. Especially steep canal reaches with embankments parallel to the bed slope, require their controlled point at the downstream side of the canal reach in order to avoid overtopping of these embankments. This type of downstream control is referred to as remote downstream control;

- Upstream control. The water level upstream of the control structure is kept as close to target level as possible. By applying this method, abundant water in the upstream canal reach causes the downstream model Predictive Control on Open Water Systems P.J van Over loop control structure to discharge extra water. This property makes upstream control highly suitable for control of drainage systems;

Various controllers have been implemented in practice. In some cases the entire control loop of measuring, computing the control actions and adjusting the structures is fully automated. In other cases, the controller is automatically supplied with measurements and this controller advices the operator. The operator can act according to this advice or can decide to ignore it and use his own judgment. This control system is referred to as Decision Support System (DSS). Many controllers have been designed for water systems, but have not yet been implemented in practice. They are tested though, on accurate hydrodynamic models of the water system to prove their applicability.

Feed back control. Feedback controllers measure the water level, compare this level to the target level and compute the change in structure setting as a function of the deviation. Often this is a proportional Integral controller (PIcontroller) in which the change in structure setting is computed from a proportional gain factor and an integral gain factor multiplied by the change in error and the error itself, respectively. The values for the proportional and integral gain factor, found in a tuning procedure, determine the behavior of the controlled water system. If the gain factors are tuned well, the controlled system will be robustly stable, reasonably fast and without severe fluctuations in structure setting. The feedback controller constantly corrects the difference between measured water level and target level in a repetitive loop. For that reason, this control method is generally referred to as closed loop control.

- Feed forward control. Feed forward controllers use measurements or predictions of a disturbance and an inverse model of the effect this disturbance has on the water level, to compute the required adjustments to the structures. The feed forward control action aims to precisely cancel this effect which would ideally result in a zero water deviation from target level. This control method is generally referred to as open loop control. As the inverse model can never perfectly represent the inverse of the effect the disturbance has on the actual water level, measurements and predictions are often inaccurate and the dynamic behavior of the actual water system changes over time, this deviation will never be zero. A combination of feedback and feed forward is often used to have the feedback control action compensate for the imperfection of the feed forward control action (40).

- Optimal control. The most common optimal controllers in water systems are based on the Linear Quadratic Regulator theory. These optimal controllers minimize an objective function by using a numerical optimization algorithm. In the objective function the square of the deviation between water level and target level is weighted against the square of the change in structure setting. The square sign gives an equal penalty for both positive and negative deviations and structure adjustments. The relative weighing between water level deviation and structure adjustment is found in a tuning procedure.

Heuristic control , opposed to the first three deterministic control methods, a group of control methods can be distinguished that is not based on physical laws, but uses a more heuristic approach. Examples of these methods are control based on rules-of-thumb, neural networks control, fuzzy logic control and genetic algorithm control. Controls based on rules-of-thumb are common for water systems that can be controlled in a straightforward, standardized manner and are not subject to control objective changes over time. Neural network control can be used if a large amount of measurements of water levels and control actions is available and the water system is too complex to model with physical formulas. Fuzzy logic can be relevant when the behavior of multiple operators working on the same control task needs to be reproduced. Genetic algorithms can find an optimal solution faster than numerical deterministic optimization algorithms. For large optimization problems though, this solution is often a local optimum. A drawback of all these methods is that the dynamic behavior of water systems is seen as a black box. Especially on this behavior, extensive research has been done over the last century and accurate formalizations of this behavior are available.

These heuristic methods are not applied to a large extent in controlling open water systems. Model predictive control presented in this research, has elements of the first three control methods namely feedback control on measured water levels, feed forward by using measured and predicted disturbance and optimal control to allow for high performance control of large water systems with interconnected canal reaches.

1.5 Significance of the study

In many cases, the control methods previously described can satisfy the specifications that are given for a controlled open water system. However, in other cases there is a limiting factor that makes it impossible for these control model predictive control on open water systems P.J van Over loop methods to function in a satisfactory manner. This factor is the limited capacity of the structures and transport canals that are used. These limited capacities are referred to as constraints on the system. The limited capacity can also become relevant if the specifications of the controlled system become more stringent over time. This higher requirement is unavoidable, as history has shown that the socioeconomic demands of the society in which the water system functions increase over time. The influence that constraints have on the controlled behavior can be illustrated by two examples; a controlled irrigation system and a controlled drainage system:

- In the past, farmers in dry areas depended on the water that was available to them irrespective of what time of the day that was. This type of water supply led to low efficiency and lo performance of the distribution system. Nowadays though, farmers have gained more political power and want their water on demand. If all farmers start to irrigate at 7 o’clock in the morning the capacity of the canals and the structures might not be high enough to accommodate the large step in flow change. The way the operators deal with this problem is to store more water in the canals before the off take change takes place.

- Drainage systems in lowland areas sometimes have to deal with extreme storm events. The operators have to keep the water levels below a certain maximum level. To achieve this, they have pumping stations with limited capacity at their disposal. If the runoff caused by extreme storms is higher than this capacity plus the available storage between target level and maximum allowable water level, this will result in inundation and consequently in damage. As operators have predictions of the storm event available, they avoid this problem by temporarily lowering the water level in the drainage canals. They prematurely will start pumping out water a couple of hours before the storm event actually takes place.

In both examples, the operator's use the effect a prediction has on their control target and the fact that the controllability is limited by the constraints on the structures they operate. It is clear that for complex water systems with interacting subsystems, water management including feedback, feed forward, weighing of small water level deviations against minimal structure adjustments and constraints on structures becomes a difficult, if not impossible task. Here, control theory comes into play to support the water manager in a formalized and systematic way.

To effectively control water systems that are characterized by optimization problems and constrained structure capacities, more advanced control methods are required. From other engineering fields, especially from control of chemical plants, a certain control methodology has gained more and more popularity over the last few decades. This control methodology is most commonly referred to as model predictive control (MPC). This methodology combines feedback control on the measured water levels and

Due to the rise in demand for water in various places including Ethiopia, water is becoming increasingly scarce resources which need to be used optimally. Successful completion of this study will provide useful information for solving many industrial and domestic problems in relation to the use of water [30]. This study can be used to model several domestic and industrial problems including those related to irrigation and specifically subsurface drip irrigation [6]. One advantage of subsurface drip irrigation is the low cost capability to apply various chemicals such as nutrients, chlorine, acids and pesticides at frequent intervals throughout the crop growing season. Frequent nutrients applications can also reduce leaching losses, especially nitrogen and may reduce the fertilizer requirement.

The model will help irrigation engineers to establish a properly designed, installed and well managed subsurface drip irrigation systems, which will enable water conservation by reducing evaporation rates and eliminating many plant diseases that can be spread through water contact with the foliage. Given the fact that most regions in Ethiopia have severe water shortage, application of the expected model will simplify and improve the agricultural production. The use of mathematics in agricultural practices will definitely modernize it! To future researchers, this study will be a useful reference when doing similar studies on fluid flow problem. To public health policy makers and environmentalists, it will be a useful model to design sewage systems in a way that waste water can be treated and used for subsurface drip irrigation purposes [6].

Water plays an essential role in the life of every person. It is used for drinking, growing crops for food, sewerage, in processes to manufacture products, to generate energy and to carry ships over waterways. An additional aspect of water, especially in lowland areas, is that people have to be protected against extremely high waters caused by high tides, high river discharges and heavy precipitation. As people live and work dispersed over large areas, the water needs to be distributed. This is illustrated by two examples; an irrigation system and a drainage system.

- Farmers have land in remote areas. They need fresh water to grow their crops. However, fresh water is often only available, from a distant source such as a reservoir or river. So canals are dug to convey the water from source to remote area, from supply to demand;
- When there is heavy precipitation, this water can flood land, causing damage and threatening lives. The excess water needs to be drained out of the area towards a river or sea. If the drainage capacity is restricted, it needs to be stored equitably in the distributed available storage in the area.

In both cases, water needs to be transported. To do so, people living together in an area have formed organizations that manage the water flows in the area. For the examples given, the organizations are respectively called irrigation districts and water boards. In order to manage the water flows, the organizations have transport canals and control structures at their disposal.

At locations where control structures are present, flows can be influenced in order to manage the water flows in the canals. In general, this is done by operators who have built up experience in managing the water system. They work at a central location where they can have access to information about the state of the water system and predictions of future changes. For irrigation systems, these predictions can be the off take schedules in which water orders are recorded. For drainage systems, a precipitation forecast is often used to predict future changes to the water system. If the present or the future state of the system is not in some desired state, the operators come up with adjustments to the structures to correct this. The operators at the central location communicate the required actions to operators that are responsible for operating the structures at the remote locations. These local operators change the settings of the structure, such as the position of a gate or the switching off and on of a pump.

This method of managing a water system is referred to as central control. Other organizations operate the structures using only information of the remote location itself. This is referred to as local control. As central control uses information from multiple locations and can make Model Predictive Control on Open Water Systems P.J van Over loop adjustments at multiple locations in the system, it is generally superior to local control in bringing the entire system in the desired state. The operators usually describe the desired state of the water system in terms of target water levels (set points). The reason is that water levels are easily (visually) measurable and they do not change rapidly. If water levels in a supply system are kept to target, the supply to the off-taking users is generally assured. If water levels in a main drainage system are kept to target, risk of flooding is averted and drainage systems can safely evacuate their excess water into the main drains.

Another indication that water levels are the most important control variables is that any deterministic and heuristic formula used in water management contains the variable water level. This is illustrated by three examples.

- If the water levels in the ditches of agricultural land are too low, there is no flow towards the ground water table and the crops will suffer. Yields will be reduced;
- Formulas describing the damage or number of casualties due to inundation use water level as an indicator;
- The available storage volume of temporary storage reservoirs can be easily computed from the water level in the reservoir.

Water system management can be formalized in a general structure diagram as given in Figure 1.1 below. The feedback controller corrects for measured deviations from set point, while the feed forward controller uses an estimate of the disturbance to counter weight the influence of the disturbance on the water level in the open water system. This block diagram holds for both central and local control, although local control generally only applies feedback control. Note that this research focuses on water quantity challenges, but that water quality specifications are often translated in extra specifications on the water quantity management. The water in canal reaches flows, driven by gravitational forces, along the meandering of the reach from the upstream to the downstream side. The velocity and the amount of the one-directional flow at each location along the stretch depend on the dimensions of the canal reach at these locations. The dimensions that influence the flow are the cross sectional area, the steepness and the roughness of the bed. A general accepted way in literature of describing the water levels and water flows in shallow canal reaches is by using the De Saint Venant equations [36].

1.6 Mathematics of Transport Phenomena

We dwell on the numerical treatment of differential equations that govern the evolution of scalar fluid properties. The derivation of these equations is usually based on certain conservation principles, as applied to an arbitrary control volume. If the fluid is in motion, it may flow in and out across the control surface which forms the boundary. Individual molecules may travel across the interface even if the fluid is at rest. Therefore, the physical and chemical properties of the fluid are influenced by those of the surrounding medium. Moreover, some quantities, such as mass, momentum, and energy, are conserved. That is, they may move from one place to another but cannot emerge out of nothing or disappear spontaneously. The physical forces that transport, produce or destroy these quantities are well known, and reliable mathematical models are available. Thus, conservation principles can be expressed in terms of differential equations that describe all relevant transport mechanisms, such as convection (also called advection), diffusion, and dispersion [2].

Instead of considering the spatial and temporal discretization each separately, it is useful to express each of them as a particular case of generic convection diffusion equation. In one dimensional Cartesian coordinate, the convection diffusion equation has the following form [7]:

$$\boxed{\frac{\partial T}{\partial t}+\frac{\partial}{\partial x}(uT)=D\frac{\partial^2 T}{\partial x^2}+S(T)} \tag{1}$$

The above generic equation can generally be simplified in the following form:

$$S(T)=\frac{\partial T}{\partial t}+\frac{\partial}{\partial x}(uT)-D\frac{\partial^2 T}{\partial x^2}$$

$$\underset{Source\,term}{S(T)} = \underset{Transient\,term}{\frac{\partial T}{\partial t}} + \underbrace{\frac{\partial}{\partial x}(uT)}_{Convection\ term} - \underbrace{D\frac{\partial^2 T}{\partial x^2}}_{Diffusion\ term}$$

$$\underset{Source\ term}{\overbrace{S(T)}^{Generation\ of\ T}} = \underset{Transient\ term}{\overbrace{\frac{\partial T}{\partial t}}^{Acummulation\ of\ T}} + \overbrace{\underbrace{\frac{\partial}{\partial x}(uT)}_{Convection\ term} - \underbrace{D\frac{\partial^2 T}{\partial x^2}}_{Diffusion\ term}}^{Transport\ of\ T}$$

The use of generic convection diffusion is not only useful to simplify the discretization, but also provides information about the physical meaning of the terms in the model. It is useful to model the transport of a generic physical magnitude (momentum, energy or mass depending on the equation) in a continuous fluid medium with a velocity field $u = velocity$ is assumed to be known. The changes in this generic magnitude are described in terms of T, the unknown of the equation. It can stand for a variety of different quantities such as mass fraction or a velocity component. As an example, temperature is the variable used to describe the energy in each point of the fluid. None of the terms of the equation has a meaning in absence of the others.

However, if they could be isolated, their role in the transport and generation of T is an infinitesimally small control volume would be (still using the energy equation as an example).
Where;

- The source term describes the generation of energy in the control volume.
- The transient term describes the energy accumulated.
- The convection term describes the flux of energy leaving the control volume due to the velocity u of the fluid medium.
- The diffusion term describes the energy flux leaving the control volume due to molecular diffusion. This process transports energy from points of higher energy to the lower energy concentration, independent of the velocity field u.

Component 1: The first term is the transient term. The word 'transient' indicates that the dependent variable (u in this case), note that the dependent variable "u" is differentiated with respect to time (It varies with time). When this component is zero, the problem is described as 'steady state'. On the other hand if you run a transient problem 'long enough' (i.e. as $t \to \infty$), our solution will approach steady state. Remember what happens to an ice cream left on top of the fridge instead of inside the fridge.

Component 2: The second term is the convective term. It aids the transport of the dependent variable by virtues of its velocity (V). For example, if a pollutant particle is dropped in a flowing stream, it gets carried along by the stream's velocity. This second term accounts for the process conveying term (passing with time).

Component 3: In the third term is the diffusive process is accounted. It happens from regions of higher concentrations or gradients to regions of lower concentrations or gradients as described by Ficks law.

The three components contribute to the naming of the equation: transient convections-diffusion equation, but the convection and diffusion term is steady state convection diffusion equation.

1.7 Convective Diffusive flux with memory

In one dimensional Cartesian co-ordinate the convection diffusion equation has the following form [8]:

$$\frac{\partial T}{\partial t}+\frac{\partial}{\partial x}(uT)=D\frac{\partial^2 T}{\partial x^2}+S(T)+\frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}}T_{xx}(x,s)ds \tag{2}$$

The above generic equation can generally be simplified in the following form:

$$S(T)=\frac{\partial T}{\partial t}+\frac{\partial}{\partial x}(uT)-D\frac{\partial^2 T}{\partial x^2}+\frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}}T_{xx}(x,s)ds$$

$$\underset{\text{Source term}}{S(T)} = \underset{\text{Transient term}}{\frac{\partial T}{\partial t}} + \underbrace{\frac{\partial}{\partial x}(uT)}_{\text{Convection term}} - \underbrace{D\frac{\partial^2 T}{\partial x^2}}_{\text{Diffusion term}} + \underbrace{\frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}}T_{xx}(x,s)ds}_{\text{Memory}}$$

$$\overbrace{\underset{\text{Source term}}{S(T)}}^{\text{Generation of }T} = \overbrace{\underset{\text{Transient term}}{\frac{\partial T}{\partial t}}}^{\text{Acummulation of }T} + \overbrace{\underbrace{\frac{\partial}{\partial x}(uT)}_{\text{Convection term}} - \underbrace{D\frac{\partial^2 T}{\partial x^2}}_{\text{Diffusion term}}}^{\text{Transport of }T} + \underbrace{\frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}}T_{xx}(x,s)ds}_{\text{Memory}}$$

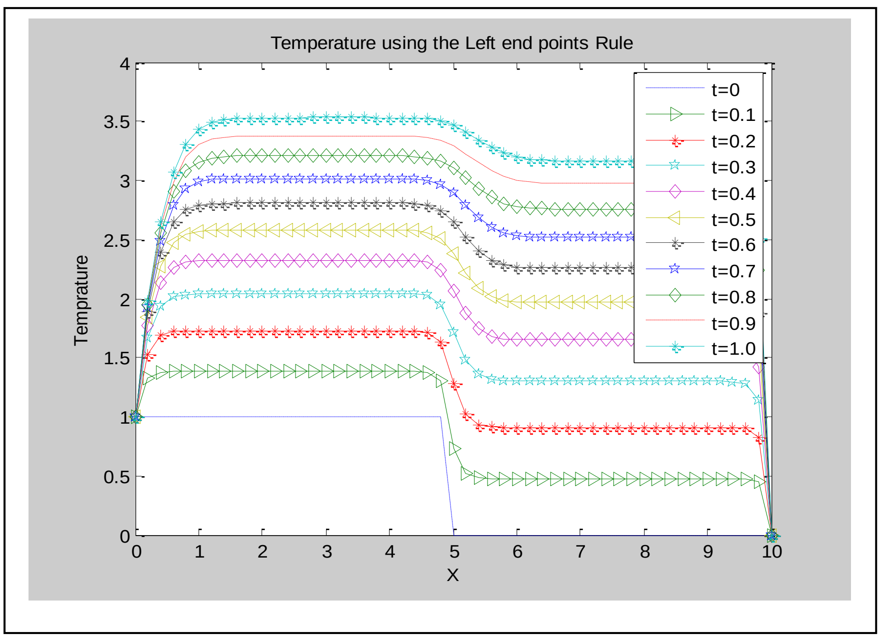


Fig 1.1 Convective Diffusive flux with memory

The above figure shows the fluid flow from one place to the other place. When we add the memory or the integral part to fluid flow the flow is more stable.

1.8 Problem Statements

The problem to be addressed in this research is that of fluid flow through a channel with one permeable wall while the other wall is impermeable. Among other applications, this study can be used to simulate the subsurface drip irrigation process/system. This is the application of water below the soil surface through emitters by using rectangular slits or pipes which are laid under the ground floor. Several mathematical models specific to subsurface drip irrigation have been developed through various studies [10].

Figure 1.2: A rectangular pipe which is laid under the ground floor, beneath the strip of crops for subsurface irrigation purpose.
Source: Irrigation pictures in Wikipedia (Theresia Bonifasi Mkenda M.Sc. (M)).

In these studies, it was emphasized that the measurement of system pressures and flow rates are very useful in monitoring system performance and are used in several methods to determine system uniformity. Many irrigation practices especially in Ethiopia are still done traditionally and we haven't found in the literature, studies which were done recently, which focused on developing mathematical models for irrigation.

Figure 1.3: Other irrigation types (apart from subsurface irrigation)
Source: Irrigation pictures in Wikipedia (Theresia Bonifasi Mkenda M.Sc. (M)).

As compared to other irrigation types, subsurface drip irrigation provides a more efficient delivery system if water and nutrient application are managed properly. Injection of nutrients, pesticides, chlorine, acids and other chemicals to modify water and soil conditions is an important component of subsurface drip irrigation. Until recently irrigation in Ethiopia took place on traditional irrigation schemes, some of which are centuries old. Although such schemes have worked well for several years back, but are now inadequate due to sharp increases in population, wear and tear of the irrigation equipments as well as catchment degradation and other environmental problems such as water logging and salinity. In order to establish efficient irrigation systems, a mathematical model of fluid flow in a channel with permeable wall can be of great help.

Once computer simulation models have been tested with field experiments, and parameters have been derived, numerical experiments may be performed. By doing so, the tedious and time consuming field experiments are replaced by numerical experiments that allow for a fast evaluation of irrigation and drainage systems on crop yields for a number of different climatic years [10].

Although there exists in the literature a number of solutions of the FKPP equations related to laminar flow in both porous rectangular channels and cylindrical pipes [11]. Only few studies have been conducted on flow problems through a channel with one porous wall and flow slip and focus its application directly to irrigation. This study is intended to fill that knowledge gap by developing a model that will include only one permeable wall and explain its practical application in real life problems.

1.9 The Governing Equation of FKPP

The aim of this paper is dedicated for seeking a numerical solution of the generalized Fisher-Kolomogrov-Petrovskii-Piskunov, (FKPP), equation subject to specified initial and boundary conditions is given by [8]:

$$u_t\left(x,t\right) = q(x,t) + f(u) + \lambda u_x(x,t) + \mu u_{xx}(x,t) + \frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds, \tag{4}$$

The integro-differential equation Eq. (3), a prototypical reaction diffusion equation, arises in a substantial of biological and chemical phenomena. Reaction transport systems with memory and long range interaction in some biological application were modeled by FKPP equation. The equation was first suggested as a deterministic version of a stochastic model for the special spread of favored gene in a population.

Chapter Two

2. Literature Review

2.1 Introduction

In recent years, several analytical and numerical studies for the generalized Fisher Kolomogrov-Petrovskii-Piskunov (FKPP), equations of type, complimented with conditions of the form, arises in the literature. A number of numerical solutions, such as a Left end point rule, a Right end point rule and Trapezoidal end point rule method of lines are available for obtaining approximate solutions of the FKPP problem. In this project, a numerical method based on finite differences and composite weighted trapezoidal rule is introduced for the numerical solution of a generalized form of the FKPP integro-differential equation subject to boundary conditions and the initial conditions [8].

2.2 Convection and diffusion equation in fluid flow

There is a growing need to provide a diverse range of agricultural products to feed the increasing world population in the face of a decreasing supply of fresh water. Consumers are demanding food products that are safe to eat and that have been produced in such a way as to minimize the environmental footprint (natural production). This entails effective management of water and agrochemicals. Worldwide there are increasing concerns about the availability and quality of fresh water for agriculture, which must also be shared with other water-user sectors. The problem becomes severe especially in arid and semi-arid areas, where water for irrigation is becoming scarcer due to recurrent droughts. In the last decades, the reaction of the scientific community to this problem has been to invest a substantial amount of research into new irrigation technologies, focusing on more efficient scheduling approaches. Plant-based methods are considered to have the greatest potential for efficient irrigation scheduling method, though with limitations. Improved scientific and practical knowledge on plant responses together with advances in electronic sensors and automated equipment for monitoring and data acquisition are helping to overcome some of these limitations.

2.3 Application for differential equation

The term "Pseudo Differential Operators" was coined [12]. These are extremely useful in Theory of Connections on Vector bundles and of course those exotic objects called "Fourier Integral Operators". The background material (like Fourier analysis, Theory of distributions etc.) is by and large as in the book "Introduction to pseudo differential & Fourier integral operators [13].

In the thesis "Problems For the generalized FKPP", the partial differential equations having the form are discussed in case where the differential operator involved is either linear or non-linear. Here non-trivial solutions in sense of distributions are discussed [14]. The initial problems for linear evolution equations with constant coefficients of any order (on a finite domain) are discussed. Here focus' method is used to fully identify and characterize the pertinent problems [3].

This chapter presents a self-contained review of some promising discretization and stabilization techniques for one dimensional transport equation. The methods to be discussed do not require directional splitting and are readily applicable to unstructured meshes. The group finite element interpolation of the flux function provides a handy link between finite element and finite volume approximations. The existence of conservative flux decomposition paves the way to various extensions of one-dimensional flow [32]. Also, it leads to efficient edge-based data structures that offer a number of significant advantages as compared to the traditional element-based implementation. In this chapter, we consider unstructured grid methods for convection-diffusion equations and discuss relevant algorithmic details, such as matrix assembly. Furthermore, we analyze the properties of discrete operators and describe some popular approaches to the design of stabilized finite element methods for stationary and time-dependent problems [2].

2.4 Application in Agriculture

Agriculture provides the basic platform for the economy of most of the countries. It is main source of livelihood of people. It provides not only food but also some important raw materials. The other advantage that agriculture provides is large scale employment. Indian economy largely depends on the earning from the field of agriculture and its associated allied fields. More than half of population work in agricultural sectors and depends on it.

It will analyze the data and will display the real time values of different parameters. Earlier developed parameter monitoring systems are discussed in next section. Section III describes the proposed monitoring system model.

The research goal of this dissertation is to improve the management of open water systems by applying model based control. To do this in an efficient way, the models of these water systems need to be set up in such a way that they contain the dynamics that are relevant for this management. In most cases, these dynamics are the water levels in canals at various locations and the water flows that influence these water levels. By using structures to manipulate the flows, a controller can achieve the management objective. This is to keep the water levels as close to set point as possible, even though boundary conditions such as tidal water level boundaries or varying in- and outflows disturb the water system.

By modeling all these parts of the water system (canal reaches, structures, disturbances, controller) a model based controller can predict the future water levels and flows that are the result of the disturbances and the control actions. In the next chapter, the model based controller referred to as model Predictive Control is described. This present chapter formalizes all sub models that can be put together to form the entire water system model.

Two general types of water systems are considered in the model predictive controller method that is applied in this research namely irrigation and drainage systems. The system consists of two flat canal reaches. The first reach R1 has rainfall-runoff inflow at the upstream side. The water flows to the downstream side, where a pump station switches on if a certain level is exceeded by the water level. The pump lifts the water to the next reach.

Drip irrigation is artificial method of supplying water to the roots of the plant. It is also called micro irrigation. In past few years there is a rapid growth in this system. The user communicates with the centralized unit through SMS. The centralized unit communicates with the system through SMS which will be received by the GSM with the help of the SIM card. The GSM sends this data to ARM7which is also continuously receives the data from sensors in some form of codes. After processing, this data is displayed on the LCD. Thus in short whenever the system receives the activation command from the subscriber it checks all the field conditions and gives a detailed feedback to the user and waits for another activation command to start the motor.

One of the reasons for this state of affairs is that many one-dimensional concepts and geometric design criteria introduced in the context of finite difference schemes are relatively easy to extend to finite volume discretizations on unstructured meshes, while an extension to finite elements is often nontrivial or even impossible. In the case of low order, finite element approximations, it turns out that finite element and finite volume schemes are largely equivalent, although their derivation is based on entirely different premises. The existing similarities between the two approaches to discretization of transport equations on unstructured meshes have been recognized, documented, and exploited by many authors [15].In spite of its enormous potential, as demonstrated by spectacular simulation results for aerodynamic applications [16], many finite element practitioners are unaware of its existence or reluctant to use an unconventional methodology.

In one of the studies, a novel soil measuring system was explained. This system used hierarchical wireless sensor network for measuring soil parameters such as temperature and humidity. The sensors used were placed underground to collect soil measurement for transmitting the data; these sensor nodes use their radios. To maintain the long life and very low duty cycle, the system used a probabilistic communication protocol. The hierarchical structure of WSN is categorized as sensor node, relay node and base node. Sensors nodes were placed below ground for better judgment of soil condition. The collected information was send to relay node which directs the data to the nearby relay or base node directly consisting of 8051 microcontroller [4].

An irrigation management model for higher crop yield was presented in 2010. It was based on some mathematical calculation used to estimate different agricultural parameters like water availability, soil compaction, biomass yield, etc of potato field, especially. This model consisted of structure of WSN which had intelligent humidity sensor, microcontroller and low power radio transreceiver for communication purpose. These nodes worked in either sensing or sleep mode. They read soil water tension (SWT) value and sent it to base station; this data could be viewed or downloaded if necessary. Based on the value of SWT, the irrigation schedule could be modified [5].

An innovative approach using cell phones was designed. This system was based on miscall and messages facility of cell phones. Specific duration of miscall was used as message/command and based on that the advance virtual RISC (AVR) sent the ON/OFF signal to the motor. Irrigation is one of the fundamental problems of agriculture in developing countries. Typically in these developing countries uneducated farmers tend to use more water than required by manual techniques hence wasting them. Soil moisture sensors are typically needed in such situations to indicate to the former when it is needed to irrigate the field and when not needed.

A survey of literature reveals some work being carried out in the field of soil moisture sensors. The earliest work in the field of soil moisture has been reported by Gaskin et al in [1]. They used a frequency domain sensor to monitor the change in the soil impedance due to variation of water content. Currently frequency domain sensors are commercialized as single and multi sensor capacitor probes with different installation and monitoring techniques [2]. Frequency Domain sensors are much better for humid tropical soil with much easier calibration procedure [4]. Recent development in moisture sensing of soil are with fiber optic sensor [5, 6], dye doped plastic fibers [8], ceramic sensor [10] are either too expensive or has reliability issues.

Method of measuring soil moisture by neutron scattering [11] cannot be applied practically. The water level indicator indicates three levels low, medium, high and also empty tank. Shen etc. al (2007) introduced a GSM-SMS remote measurement and control system for greenhouse based on PC-based database system connected with base station. Base station is developed by using a microcontroller, GSM module, sensors and actuators. In practical operation, the central station receives and sends messages through GSM module.

To be a good irrigator, your number one goal is to use the least amount of water possible to keep the soil moisture content in the root zone at the appropriate levels. To meet this goal, you have to turn on the water before the moisture content drops below the lowest allowed level (maximum allowed depletion), and shut off the water before it goes above the highest allowed level (field capacity). Read more about Soil Moisture Content Levels below. Your second goal should be to water as infrequently as possible because this strategy will promote deeper plant roots while minimizing the incidence of disease. Maximizing the rooting potential of the plant allows it to access as much water and nutrients as possible. Many scientific studies have examined the potential for improving a plant’s water-use efficiency (usually referred to as WUE), and it is widely accepted that a plant’s water use can be optimized with good irrigation practices.

We know deep and infrequent watering can get complicated, and it is not always possible due to watering restrictions, available water supply, or events. Baseline has many tools designed to meet these challenges. For the time being, the nodal values **v** is assumed to be known. In real-life applications, they are computed numerically from a momentum balance equation.

This approximation technique is known as the group finite element formulation. It eliminates the need for dealing with products of basic functions and leads to simpler discretized equations, which results in significant savings especially in the case of highly nonlinear problems [3]. In particular, this kind of approximation provides a natural treatment of in viscid fluxes in conservation laws. In many situations, it turns out to be more accurate than the independent approximation of the involved variables. The group finite element approximation of the convective flux with memory term is given by:

$$u_t\left(x,t\right) = q(x,t) + f(u) + \lambda u_x(x,t) + \mu u_{xx}(x,t) + \frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds, \qquad (3)$$

In order to irrigate properly, you need to understand the basics of your system. Irrigation systems are broken into smaller sections called zones or stations. The zone is made up of a group of sprinkler heads that are turned on by the same valve. Always remember that the smallest area the controller can manage is a zone. If you have a dry or wet zone, you can adjust it in several ways from the controller, but if you have a wet or dry spot inside the zone, you can either choose to fix whatever is causing the issue or you can carry on and find other ways to compensate.

CHAPTER THREE

MATHEMATICAL FORMULATION OF THE MODEL

3.1 Description of transport equation

In this chapter, we elaborate on the qualitative behavior of solutions to one-dimensional equations of generalized FKPP type. We analyze the properties of differential operators and derive *a* priori bounds that depend on the initial and/or boundary conditions. Left, right and trapezoidal end point rule are formulated for one-dimensional model.

The basic fluid equations needed for the problem under investigation are the conservation of mass (continuity equation) and the conservation of momentum (FKPP). The continuity equation requires that the mass of fluid entering a fixed control volume either leaves that volume or accumulates within it. It is thus a "mass balance" requirement posed in mathematical form, and is a scalar equation. The momentum equation may be thought of as a "momentum balance". These are vector equations, that is, there is a separate equation for each of the coordinate directions and they are the fluid dynamics equivalent of Newton's second law: force equals mass times acceleration. In situations where the fluid may be treated as incompressible and temperature differences are small, the continuity and momentum equations are sufficient to specify the velocities and pressure, that is, four equation and four unknown quantities. If the flow is compressible (density is not constant), or if heat flux occurs (temperature not constant), we have to add at least one additional equation and often, the energy equation is used. These equations may be used to analyze the flow of most common fluids in internal or external flow situations.

The governing equation is:

$$u_t = u(1-u) + \lambda u_x + \mu u_{xx} + \frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)\,ds \tag{5}$$

3.2 Discritization of convection diffusion transport model without Heaviside function

The model is:

$$\frac{\partial T}{\partial t}+\frac{\partial}{\partial t}(uT)=\alpha\frac{\partial^2 T}{\partial x^2}+\lambda(T_w-T) \qquad (11)$$

The FDE: Using an implicit FTCS scheme.

$$\frac{T_i^{n+1}-T_i^n}{\Delta t}+u\frac{T_{i+1}^{n+1}-T_{i-1}^n}{2\Delta x}=\alpha\frac{T_{i+1}^{n+1}-2T_i^{n+1}+T_{i-1}^{n+1}}{\Delta x^2}+\lambda(T_w-T_i^n) \qquad (12)$$

Since u = constant, we also can use the upwind scheme as follows:

$$\frac{T_i^{n+1}-T_i^n}{\Delta t}+u\frac{T_{i+1}^{n+1}-T_{i-1}^n}{2\Delta x}=\alpha\frac{T_{i+1}^{n+1}-2T_i^{n+1}+T_{i-1}^{n+1}}{\Delta x^2}+\lambda(T_w-T_i^n) \qquad (13)$$

The source term is treated explicitly.

Rearrange the upwind FDE:

$$-\left[\frac{\alpha\Delta t}{\Delta x^2}+\frac{u\Delta t}{\Delta x}\right]T_{i-1}^{n+1}+\left[1+\frac{u\Delta t}{\Delta x}+\frac{2\alpha\Delta t}{\Delta x^2}\right]T_i^{n+1}-\left[\frac{\alpha\Delta t}{\Delta x^2}\right]T_{i+1}^{n+1}=T_i^n+\lambda\Delta t(T_w-T_i^n) \qquad (14)$$

$$-\left[\frac{\alpha\Delta t}{\Delta x^2}+\frac{u\Delta t}{\Delta x}\right]T_{i-1}^{n+1}+\left[1+\frac{u\Delta t}{\Delta x}+2\left(\frac{\alpha\Delta t}{\Delta x^2}\right)\right]T_i^{n+1}-\left(\frac{\alpha\Delta t}{\Delta x^2}\right)T_{i+1}^{n+1}=T_i^n+\lambda\Delta t(T_W-T_i^n)H(x_i)$$

For the boundary node:

$$i=0,\quad T_0^{n+1}=T_0$$

$$i=nx,\quad T_{nx}^{n+1}=T_{nx-1}^{n+1}$$

In the simulation we set:

The above set of linear equations can be solved by TDMA.

Where; $A = -\left(\frac{\alpha\Delta t}{\Delta x^2} + \frac{u\Delta t}{\Delta x}\right)$, $B = \left(1 + \frac{u\Delta t}{\Delta x} + 2\left(\frac{\alpha\Delta t}{\Delta x^2}\right)\right)$

$C = -\left(\frac{\alpha\Delta t}{\Delta x^2}\right)$, and $D = T_i^n + \lambda\Delta t(T_w - T_i^n)$

3.3 Discritization of an integro-partial differential equation transport model

3.31 Discretizing the Partial Differential Equation

$$u_t = u(1-u) + \lambda u_x + \mu u_{xx} + \frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds + (u_w - u) \tag{14}$$

Table 3.1 The extended nonlinear FKPP equation has two parts, the PDEs and the Integral. The main focus of this paper is to show how the different numerical methods of finding integrals can be used to compute the integro-partial differential equation. So now let us see how first we discretize the PDEs.

Type	Symbol	PDE form	Discretized form	Name
Transient Term	u_t	$\frac{\partial u}{\partial t}$	$\frac{u_i^{k+1} - u_i^k}{\Delta t}$	Forward difference in time
Reaction term	$u(1-u)$	-	-	-
Advection term	u_x	$\frac{\partial u}{\partial x}$	$\frac{u_{i+1}^{k+1} - u_{i-1}^k}{2\Delta x}$	Central difference in space
Diffusion term	u_{xx}	$\frac{\partial^2 u}{\partial x^2}$	$\frac{u_{i+1}^k - 2u_i^k + u_{i-1}^k}{\Delta x}$	Central difference in space

Thus, the PDE is discritized as follows:

$$\boxed{u_t = u(1-u) + \lambda u_x + \mu u_{xx}} \tag{15}$$

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

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

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

$$u_i^{k+1} = u_i^k + \Delta t.u_i^k - \Delta t.(u_i^k)^2 + \frac{\lambda\Delta t}{2\Delta x}u_{i+1}{}^k - \frac{\lambda\Delta t}{2\Delta x}u_{i-1}{}^k + \frac{\mu\Delta t}{\Delta x^2}u_{i+1}{}^k - 2\frac{\mu\Delta t}{\Delta x^2}u_i^k + \frac{\mu\Delta t}{\Delta x^2}u_{i-1}{}^k$$

The main focus of this paper is to show how the different numerical methods of finding integrals can be used to compute the integro-partial differential equation.

3.32 Discretizing the Integro-partial differential equation explicitly

This is used to approximate the shape as a rectangle as depicted in the figure below. The area of a rectangle is computed using height and width. There is only one height and width.

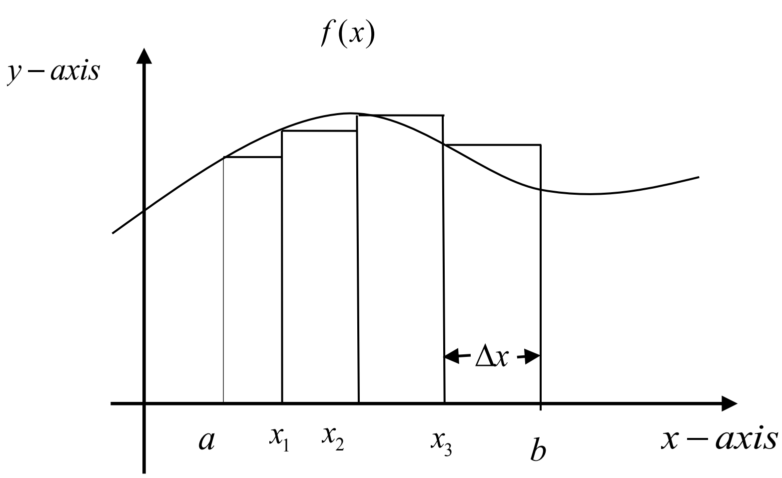


Fig 3.1 A graph shows how the function is segmented and considered

For this rule, the left side of the right from the subinterval is considered to create the rectangle. If the gap is an ascending graph the rectangle will be subscribed by the area under the graph and if the graph is a descending, the rectangle will subscribe the area under the graph.

In both cases the rectangle approximates the area with an error. In an equation form:

$$\int_a^b f(x)dx = f(a).(x_1 - a) \tag{15}$$

When the computation is for the whole graph, it is the sum of individual rectangles and it will be of the following general formula:

$$\int_a^b f(x)dx = \Delta x \sum_{i=1}^{n} f(x_{i-1}) \tag{16}$$

We need to discretize the integral sign according to equation (5)

The following discretization show how this can be done.

$$\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds = \Delta t \sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}} u_{xx}(x,t_k) \tag{17}$$

Using the equation (16) and equation (17) in equation (1):

$$u_t = u(1-u) + \lambda u_x + \mu u_{xx} + \frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds + (u_w - u) \tag{18}$$

Then, using this concept we can define the memory term using time step discretization as follows:

$$\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds = \Delta t \sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}} u_{xx}(x,t_k) \tag{19}$$

Thus, we can descrtization the governing equation as follows:

$$\frac{u_i^{k+1}-u_i^k}{\Delta t}=u_i^k-(u_i^k)^2+\lambda\left[\frac{u_{i+1}{}^k-u_{i-1}{}^k}{2\Delta x}\right]+\mu\left[\frac{u_{i+1}{}^k-2u_i^k+u_{i-1}{}^k}{2\Delta x}\right]+\frac{D}{\tau}\left[\sum_{i=1}^{N}e^{-\frac{(t_{N+1})-(t_k)}{\tau}}u_{xx}(x,t_k)\Delta t\right] \quad (20)$$

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

$$+\frac{D\Delta t^2}{\tau}\left[\sum_{i=1}^{N}e^{-\frac{(t_{N+1})-(t_k)}{\tau}}\frac{u_{i+1}{}^k-2u_i^k+u_{i-1}{}^k}{\Delta x^2}\right]+\Delta t(u_w-u_i^k) \quad (21)$$

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

$$+\frac{D\Delta t^2}{\tau\Delta x^2}\left[\sum_{i=1}^{N}e^{-\frac{t_{N+1-t_k}}{\tau}}(u_{i+1}{}^k-2u_i^k+u_{i-1}{}^k)\right]+\Delta t(u_w-u_i^k) \quad (22)$$

$$u_i^{k+1}=u_i^k+\Delta t.u_i^k-\Delta t.(u_i^k)^2+\frac{\lambda\Delta t}{2\Delta x}u_{i+1}{}^k-\frac{\lambda\Delta t}{2\Delta x}u_{i-1}{}^k+\frac{\mu\Delta t}{\Delta x^2}u_{i+1}{}^k-2\frac{\mu\Delta t}{\Delta x^2}u_i^k+\frac{\mu\Delta t}{\Delta x^2}u_{i-1}{}^k$$

$$+\frac{D\Delta t^2}{\tau\Delta x^2}\left[\sum_{i=1}^{N}e^{-\frac{t(N+1)-t(k)}{\tau}}(u_{i+1}{}^k-2u_i^k+u_{i-1}{}^k)\right]+\Delta t(u_w-u_i^k) \quad (23)$$

Comment: There is only one unknown on the LHS (u_i^{k+1} term), the other terms on the RHS (the k time term in superscript are all known). So we apply an explicit implementation. Next we recast equation (10) in an easily computable form by doing the following:

(A) Put the unknown term on the LHS,

(B) Make the nonlinear term to be easily computable by putting it as the 2nd term on the RHS of eq. (10),

Note: The first term u_i^k comes from the transient term on the LHS.

(C) Separate out the coefficients of $u_{i-1}{}^k$, u_i^k and $u_{i+1}{}^k$

$$u_i^{k+1} = u_i^k + \Delta t.u_i^k - \Delta t.(u_i^k)^2 + \frac{\lambda\Delta t}{2\Delta x}u_{i+1}{}^k - \frac{\lambda\Delta t}{2\Delta x}u_{i-1}{}^k + \frac{\mu\Delta t}{\Delta x^2}u_{i+1}{}^k - \frac{2\mu\Delta t}{\Delta x^2}u_i^k + \frac{\mu\Delta t}{\Delta x^2}u_{i-1}{}^k$$
$$+ \frac{D\Delta t^2}{\tau\Delta x^2}\sum_{i=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}} u_{i+1}{}^k - \frac{2D\Delta t^2}{\tau\Delta x^2}\sum_{i=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}} u_i^k + \frac{D\Delta t^2}{\tau\Delta x^2}\sum_{i=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}} u_{i-1}{}^k \tag{25}$$

$$u_i^{k+1} = \Delta t.u_i^k - \Delta t.(u_i^k)^2 + \frac{\lambda\Delta t}{2\Delta x}u_{i+1}{}^k + \frac{\mu\Delta t}{\Delta x^2}u_{i+1}{}^k + \frac{D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}} u_{i+1}{}^k + u_i^k - \frac{2\mu\Delta t}{\Delta x^2}u_i^k$$
$$- \frac{2D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}} u_i^k - \frac{\lambda\Delta t}{2\Delta x}u_{i-1}{}^k + \frac{\mu\Delta t}{\Delta x^2}u_{i-1}{}^k + \frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}} u_{i-1}{}^k + \Delta t(u_w - u_i^k)$$

Then, we can collect the like terms together by using the above Description:

$$u_i^{k+1} = \Delta t.u_i^k - \Delta t.(u_i^k)^2 + \left(\underbrace{-\frac{\lambda\Delta t}{2\Delta x} + \frac{\mu\Delta t}{\Delta x^2} + \frac{D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}}}_{u_{i-1}{}^k}\right) + \left(\underbrace{1 - \frac{2\mu\Delta t}{\Delta x^2} - \frac{2D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}}}_{u_i^k}\right)$$
$$+ \left(\underbrace{\frac{\lambda\Delta t}{2\Delta x} + \frac{\mu\Delta t}{\Delta x^2} + \frac{D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}}}_{u_{i+1}{}^k}\right) + \Delta t(u_w - u_i^k) \tag{26}$$

$$u_{i-1}^k\left(coefficient\right) = \quad A(coefficient) = \left(\underbrace{-\frac{\lambda\Delta t}{2\Delta x} + \frac{\mu\Delta t}{\Delta x^2} + \frac{D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}}}_{u_{i-1}^k}\right)$$

$$u_i^k\left(coefficient\right) = B\left(coefficient\right) = \left(\underbrace{1 - \frac{2\mu\Delta t}{\Delta x^2} - \frac{2D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}}}_{u_i^k}\right)$$

$$u_{i+1}^{k}\left(coefficient\right)= \quad C\left(coefficient\right)= \left(\underbrace{\frac{\lambda\Delta t}{2\Delta x}+\frac{\mu\Delta t}{\Delta x^{2}}+\frac{D\Delta t}{\tau\Delta x^{2}}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k)}{\tau}}}_{u_{i+1}^{k}}\right)$$

$$D=\Delta t.u_{i}^{k}-\Delta t.(u_{i}^{k})^{2}+\left(\underbrace{-\frac{\lambda\Delta t}{2\Delta x}+\frac{\mu\Delta t}{\Delta x^{2}}+\frac{D\Delta t}{\tau\Delta x^{2}}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k)}{\tau}}}_{u_{i-1}^{k}}\right)+\left(\underbrace{1-\frac{2\mu\Delta t}{\Delta x^{2}}-\frac{2D\Delta t}{\tau\Delta x^{2}}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k)}{\tau}}}_{u_{i}^{k}}\right)$$

$$+\left(\underbrace{\frac{\lambda\Delta t}{2\Delta x}+\frac{\mu\Delta t}{\Delta x^{2}}+\frac{D\Delta t}{\tau\Delta x^{2}}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k)}{\tau}}}_{u_{i+1}^{k}}\right)+\Delta t(u_{w}-u_{i}^{k})$$

(D) Identify the constant terms with letters.

$$let \quad r_{1}=\frac{\lambda\Delta t}{2\Delta x}, \quad r_{2}=\frac{\mu\Delta t}{\Delta x^{2}}, \quad r_{3}=\frac{D\Delta t}{\tau\Delta x^{2}}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k)}{\tau}}$$

Then,

$$u_{i}^{k+1}=\Delta t.u_{i}^{k}-\Delta t.(u_{i}^{k})^{2}+\left(r_{1}+r_{2}+r_{3}\right)u_{i+1}^{k}+\left(1-2r_{1}-2r_{2}\right)u_{i}^{k}+\left(-r_{1}+r_{2}+r_{3}\right)u_{i-1}^{k}+\Delta t(u_{w}-u_{i}^{k})$$

This did not formulate a tri-diagonal matrix. The computation of the result is direct forward substitution.

3.33 Discretizing the Integro-partial differential equation implicitly

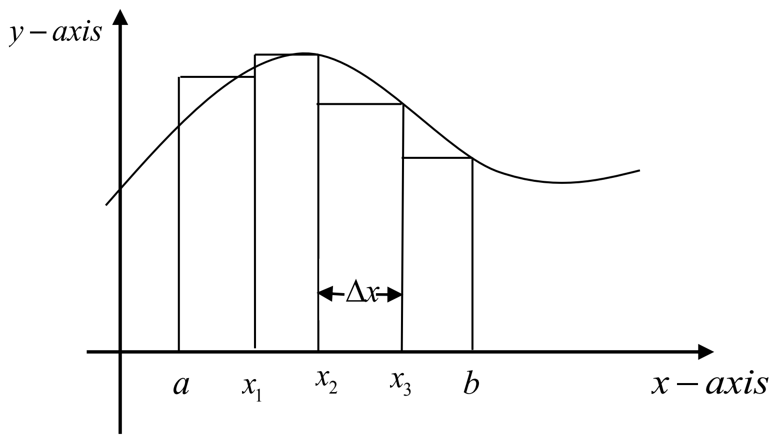


Fig 3.2 A graph shows how the function is segmented and considered

For the right end rule, the right side of the interval functional value is used as a height to approximate the shape into a rectangle. The equation now will have the following form:

$$\int_{a}^{x_1} f(x)dx = f(x_1).(x_1 - a) \tag{27}$$

The general formula:

$$\int_{x_0}^{x_n} f(x)dx = \Delta x \sum_{i=1}^{n} f(x_{i+1}) \tag{28}$$

Substituting the general formula for the given example:

$$\int_{0}^{t} e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds = \Delta t \sum_{k=1}^{N} \left(e^{-\frac{t(N+1)-t(k+1)}{\tau}} u_{xx}(x,t_{k+1}) \right) \tag{29}$$

Then, applying the right end rule into the equation (1) is:

$$u_t = u(1-u) + \lambda u_x + \mu u_{xx} + \frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds + (u_w - u_i^k) \tag{30}$$

Note: The nonlinear term i.e $u(1-u)$ s done implicitly in Appendex A (See appendix A).

Then, how do we relate spatial index to the time index in order to resolve the time discretization of the memory?

<u>**Note**</u>:

$$Rn = \sum f(x_{i+1})\Delta x \tag{31}$$

Then, using this concept we can define the memory term as follows:

$$\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds = \Delta t \sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}} u_{xx}(x,t_{k+1}) \tag{32}$$

Thus, it can be discretized as follows:

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = \lambda\left[\frac{u_{i+1}^{k+1} - u_{i-1}^{k+1}}{2\Delta x}\right] + \mu\left[\frac{u_{i+1}^{k+1} - 2u_i^{k+1} + u_{i-1}^{k+1}}{\Delta x^2}\right] + \frac{D\Delta t}{\tau}\left[\sum_{i=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}} u_{xx}(x,t_{k+1})\right]$$

$$u_i^{k+1} - u_i^k = \Delta t.\lambda\left[\frac{u_{i+1}^{k} - u_{i-1}^{k}}{2\Delta x}\right] + \Delta t.\mu\left[\frac{u_{i+1}^{k} - 2u_i^{k} + u_{i-1}^{k}}{\Delta x^2}\right]$$

$$+\frac{D\Delta t^2}{\tau}\left(\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}} \frac{u_{i+1}^{k+1} - 2u_i^{k+1} + u_{i-1}^{k+1}}{\Delta x^2}\right) + \Delta t(u_w - u_i^k) \tag{33}$$

Know we will discretize with respect to time. The following discretization shows how this can be done.

$$\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds = \Delta t \sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}} u_{xx}(x,t_{k+1})$$

$$u_i^{k+1} - u_i^k = \frac{\lambda \Delta t}{2\Delta x}\left[u_{i+1}{}^k - u_{i-1}{}^k\right] + \frac{\mu \Delta t}{\Delta x^2}\left[u_{i+1}{}^k - 2u_i^k + u_{i-1}{}^k\right]$$
$$+\frac{D\Delta t^2}{\tau \Delta x^2}\left(\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}} (u_{i+1}{}^{k+1} - 2u_i^{k+1} + u_{i-1}{}^{k+1})\right) \tag{34}$$

$$u_i^{k+1} = u_i^k + \frac{\lambda \Delta t}{2\Delta x} u_{i+1}{}^k - \frac{\lambda \Delta t}{2\Delta x} u_{i-1}{}^k + \frac{\mu \Delta t}{\Delta x^2} u_{i+1}{}^k - \frac{2\mu \Delta t}{\Delta x^2} u_i^k + \frac{\mu \Delta t}{\Delta x^2} u_{i-1}{}^k$$
$$+\frac{D\Delta t^2}{\tau \Delta x^2}\sum_{i=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}} (u_{i+1}{}^{k+1} - 2u_i^{k+1} + u_{i-1}{}^{k+1}) + \Delta t(u_w - u_i^k) \tag{35}$$

<u>Comment</u>: There is only one known on the RHS (u_i^k term), the other terms on the LHS (the $k+1$ time term in superscript are all unknown). So we apply an implicit implementation. Next we recast equation (10) in an easily computable form by doing the following:

(A) Put the unknown term on the LHS,

(B) The first term u_i^k comes from the transient term on the LHS.

(C) Separate out the coefficients of $u_{i-1}{}^{k+1}$, $u_i{}^{k+1}$ and $u_{i+1}{}^{k+1}$.

$$\left(\underbrace{-\frac{D\Delta t^2}{\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k+1)}{\tau}}}_{u_{i+1}^{k+1}}\right)+\left(\underbrace{1+\frac{2D\Delta t^2}{\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k+1)}{\tau}}+\underbrace{\Delta t(-1+2u_i^k)}_{nonlinear\,term}}_{u_i^{k+1}}\right)+\left(\underbrace{-\frac{D\Delta t^2}{\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k+1)}{\tau}}}_{u_{i-1}^{k+1}}\right)=$$

$$\frac{\lambda\Delta t}{2\Delta x}u_{i+1}{}^{k}+\frac{\mu\Delta t}{\Delta x^2}u_{i+1}{}^{k}+u_i{}^{k}-\frac{2\mu\Delta t}{\Delta x^2}u_i{}^{k}-\frac{\lambda\Delta t}{2\Delta x}u_{i-1}{}^{k}+\frac{\mu\Delta t}{\Delta x^2}u_{i-1}{}^{k}+\Delta t(u_w-u_i^k)$$

$$\left(\underbrace{-\frac{D\Delta t^2}{2\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k+1)}{\tau}}}_{u_{i-1}^{k+1}}\right)+\left(\underbrace{1+\frac{2D\Delta t^2}{\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k+1)}{\tau}}+\underbrace{\Delta t(-1+2u_i^k)}_{nonlinear\,term}}_{u_i^{k+1}}\right)+\left(\underbrace{-\frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k+1)}{\tau}}}_{u_{i+1}^{k+1}}\right)=$$

(36)

$$\left(-\frac{\lambda\Delta t}{2\Delta x}+\frac{\mu\Delta t}{\Delta x^2}\right)u_{i-1}{}^{k}+\left(1-\frac{2\mu\Delta t}{\Delta x^2}\right)u_i{}^{k}+\left(\frac{\lambda\Delta t}{2\Delta x}+\frac{\mu\Delta t}{\Delta x^2}\right)u_{i+1}{}^{k}+\underbrace{(u^2)_i^n}_{nonlinear\,term}+\Delta t(u_w-u_i^k)$$

1) $u_{i-1}^{k+1}(coefficient)=A\left(coefficient\ for\,tridiagonal\,value\right)=\left(\underbrace{-\frac{D\Delta t^2}{2\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k+1)}{\tau}}}_{u_{i-1}^{k+1}}\right)$

2) $u_i^{k+1}(coefficient)=B\left(coefficient\ for\,tridigonal\,value\right)=\left(\underbrace{1+\frac{2D\Delta t^2}{\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k)}{\tau}}+\underbrace{\Delta t(-1+2u_i^k)}_{nonlinear\,term}}_{u_i^{k+1}}\right)$

3) $u_{i+1}^{k+1}(coefficient)=C\left(coefficient\ for\,tridiagonal\,value\right)=\left(\underbrace{-\frac{D\Delta t^2}{2\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k+1)}{\tau}}}_{u_{i+1}^{k+1}}\right)$

$$D=\left(-\frac{\lambda\Delta t}{2\Delta x}+\frac{\mu\Delta t}{\Delta x^2}\right)u_{i-1}{}^{k}+\left(1-\frac{2\mu\Delta t}{\Delta x^2}\right)u_i{}^{k}+\left(\frac{\lambda\Delta t}{2\Delta x}+\frac{\mu\Delta t}{\Delta x^2}\right)u_{i+1}{}^{k}+\underbrace{(u^2)_i^n}_{nonlinear\,term}+\Delta t(u_w-u_i^k)$$

(D) Identify the constant terms with letters.

$$let \quad r=\frac{D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k+1)}{\tau}} \qquad r_1=\frac{\lambda\Delta t}{\Delta x^2}, \quad and \quad r_2=\frac{u\Delta t}{2\Delta x}$$

Then,

$$-ru_{i+1}{}^{k+1}+(1+r+\underbrace{\Delta t(-1+2u_i^k)}_{nonlinear\,term})u_i{}^{k+1}-ru_{i-1}{}^{k+1}=\left(r_1+r_2\right)u_{i+1}{}^{k}+\left(1-2r_2\right)u_i{}^{k}+\left(-r_1+r_2\right)u_{i-1}{}^{k}+\underbrace{\Delta t.(u^2)_i^n}_{nonlinear\,term}$$

$$\begin{cases} A=-r \\ B=1+2r+\underbrace{\Delta t(-1+2u_i^k)}_{nonlinear\,term} \\ C=-r \\ D=(r_1+r_2)u_{i+1}{}^{k}+(1-2r_2)u_i{}^{k}+(-r_1+r_2)u_{i-1}{}^{k}+\underbrace{\Delta t.(u^2)_i^n}_{nonlinear\,term}+\Delta t(u_w-u_i^k) \end{cases}$$

3.34 Discretizing the Integro-partial differential equation by Crank-Nicolson method

In calculus we learned that the integrals are (signed) areas and can be approximated by sums of the smaller areas, such as the areas of rectangles. As depicted the picture below the integration of a function f(x) b/n a and b is represented by the area under the function curve and the point a from the left and point b from the right and also the x-axis from below.

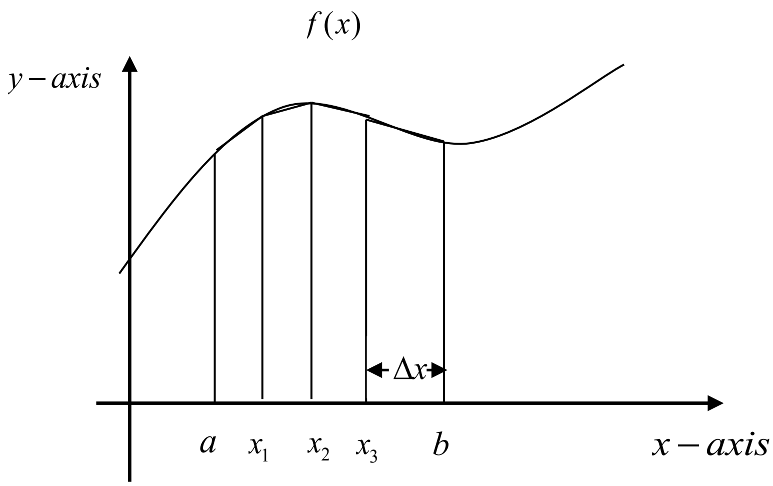


Fig 3.3 A graph shows how the function is segmented and considered

To find the approximated area under the curve we can split the curve as it is show in the figure in equal parts. If we considered the first portion (i.e. b/n a and x_1) and the area the picture can be approximated to a trapezoid. The width of the trapezoid $x_1 - a$, and the length of the two parallel sides are $f(x_1) \& f(a)$. Calculating the area under the curve bounded under (a, x_1) (for a segment) is given by:

$$\int_a^{x_1} f(x)dx = \frac{1}{2}(f(a) + f(x_1)).(x_1 - a) \tag{37}$$

Then, equation (24a) can generally be written for the total area under the curve.

$$\int_a^b f(x)dx = \frac{1}{2}\sum_{i=1}^{N}\{(f(x_i)+f(x_{i+1}))\} \qquad (38)$$

Extending by choosing points $\{x_1\}$ that subdivided (a,b) into n partition in the following manner:

$$a = x_0 < x_1 < \ldots < x_{n-1} < x_n = b \qquad (39)$$

The subinterval (x_{i-1}, x_i) determine the width Δx_i where $\Delta x = \frac{b-a}{n}$ since we are creating equal subspaces. For the height, the function $f(x_{i-1})$ and $f(x_i)$ are considered.

The general approximation of the area under the interval (a,b) is given by:-

$$\int_{x_0}^{x_n} f(x)dx = \frac{\Delta t}{2}\sum_{i=1}^{n}(f(x_{i-1})+f(x_i)) \qquad (40)$$

Using this formula for the given integral in the example will give us the following form (k supposed to be from 0 to N, but **MATLAB** starts array index from 1). So accordingly k is run from 1 to N + 1 implied in k and k + 1 instead of k – 1 and k.

$$\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds = \frac{\Delta t}{2}\sum_{k=1}^{N}\left(e^{-\frac{t(N+1)-t(k)}{\tau}} u_{xx}(x,t_k) + e^{-\frac{t(N+1)-t(k+1)}{\tau}} u_{xx}(x,t_{k+1})\right) \qquad (41)$$

Now we shall see how this can be done with respect to time. The following discretization show how this can be done.

Using the equation (26) and equation (27) in equation (1):

$$u_t = u(1-u) + \lambda u_x + \mu u_{xx} + \frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds + \Delta t(u_w - u_i^k) \tag{42}$$

Note: The nonlinear term i.e $u(1-u)$ is done implicitly in Appendix A (See appendix A).

Then, how do we relate spatial index to the time index in order to resolve the time discretization of the memory?

- The Trapezoidal rule discretization of the memory term.

<u>Note</u>:

$$T_p = \frac{1}{2}\sum\left(f(x_i) + f(x_{i+1})\right)\Delta x \tag{43}$$

Then, using this concept we can define the memory term as follows:

$$\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds = \frac{\Delta t}{2}\sum_{k=1}^{N}\left(e^{-\frac{t(N+1)-t(k)}{\tau}} u_{xx}(x,t_k) + e^{-\frac{t(N+1)-t(k+1)}{\tau}} u_{xx}(x,t_{k+1})\right) \tag{44}$$

<u>Note that</u>: k is an index for time (see eqn. 19) where as i is an index for space (see eqn. 19) and t (k+1) is the upper limit of the time in time scale as reflected by the Trapezoidal rule of equation (19).

Thus**,**

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = \lambda\left(\frac{u_{i+1}^k - u_{i-1}^k}{2\Delta x}\right) + \mu\left(\frac{u_{i+1}^k - 2u_i^k + u_{i-1}^k}{\Delta x^2}\right)$$
$$+\frac{D}{\tau}\left(\frac{\Delta t}{2}\sum_{k=1}^{N}\left(e^{-\frac{t(N+1)-t(k)}{\tau}} u_{xx}(x,t_k) + e^{-\frac{t(N+1)-t(k+1)}{\tau}} u_{xx}(x,t_{k+1})\right)\right) + \Delta t(u_w - u_i^k) \tag{45}$$

$$u_i^{k+1} - u_i^k = \Delta t.\lambda\left(\frac{u_{i+1}{}^k - u_{i-1}{}^k}{2\Delta x}\right) + \Delta t.\mu\left(\frac{u_{i+1}{}^k - 2u_i^k + u_{i-1}{}^k}{2\Delta x}\right) + \Delta t(u_w - u_i^k)$$

$$+\frac{D\Delta t^2}{2\tau}\left(\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}} \frac{u_{i+1}{}^k - 2u_i^k + u_{i-1}{}^k}{\Delta x^2} + \sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}} \frac{u_{i+1}{}^k - 2u_i^k + u_{i-1}{}^k}{\Delta x^2}\right)$$

$$u_i^{k+1} - u_i^k = \frac{\lambda\Delta t}{2\Delta x}\left(u_{i+1}{}^k - u_{i-1}{}^k\right) + \frac{\mu\Delta t}{\Delta x^2}\left(u_{i+1}{}^k - 2u_i^k + u_{i-1}{}^k\right) + \Delta t(u_w - u_i^k)$$

$$+\frac{D\Delta t}{2\tau\Delta x^2}\left(\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}}\left(u_{i-1}{}^k - u_i^k + u_{i+1}{}^k\right) + \sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}}\left(u_{i-1}{}^{k+1} - u_i^{k+1} + u_{i+1}{}^{k+1}\right)\right) \quad (46)$$

$$u_i^{k+1} = u_i^k + \frac{\lambda\Delta t}{2\Delta x}u_{i+1}{}^k - \frac{\lambda\Delta t}{2\Delta x}u_{i-1}{}^k + \frac{\mu\Delta t}{\Delta x^2}u_{i+1}{}^k - \frac{2\mu\Delta t}{\Delta x^2}u_i^k + \frac{\mu\Delta t}{\Delta x^2}u_{i-1}{}^k + \Delta t(u_w - u_i^k)$$

$$+\frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}}\left(u_{i-1}{}^k - u_i^k + u_{i+1}{}^k\right) + \frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}}\left(u_{i-1}{}^{k+1} - u_i^{k+1} + u_{i+1}{}^{k+1}\right)$$

<u>**Comment**</u>**:** There are three unknowns on the LHS ($u_{i-1}{}^{k+1}, u_i^{k+1}$ and $u_{i+1}{}^{k+1}$ term), the other terms on the RHS (the k time term in superscript are all known). So we apply an implicit implementation. Next we recast equation (10) in an easily computable form by doing the following:

(A) Put the unknown term on the LHS,

(B) The first term u_i^k comes from the transient term on the LHS.

(C) Separate out the coefficients of $u_{i-1}{}^{k+1}$, u_i^{k+1} and $u_{i+1}{}^{k+1}$.

$$u_i^{k+1} = u_i^k + \frac{\lambda\Delta t}{2\Delta x}u_{i+1}{}^k - \frac{\lambda\Delta t}{2\Delta x}u_{i-1}{}^k + \frac{\mu\Delta t}{\Delta x^2}u_{i+1}{}^k - \frac{2\mu\Delta t}{\Delta x^2}u_i^k + \frac{\mu\Delta t}{\Delta x^2}u_{i-1}{}^k + \Delta t(u_w - u_i^k)$$

$$+\frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}} u_{i+1}{}^k - \frac{D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}} u_i^k + \frac{D\Delta t^2}{2\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}} u_{i-1}{}^k \quad (47)$$

$$+\frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}} u_{i+1}{}^{k+1} - \frac{D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}} u_i^{k+1} + \frac{D\Delta t^2}{2\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}} u_{i-1}{}^{k+1}$$

$$-\frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k+1)}{\tau}}u_{i+1}{}^{k+1}+u_i^{k+1}+\frac{D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k+1)}{\tau}}u_i^{k+1}-\frac{D\Delta t^2}{2\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k+1)}{\tau}}u_{i-1}{}^{k+1}$$

$$=\frac{\lambda\Delta t}{2\Delta x}u_{i+1}{}^{k}+\frac{\mu\Delta t}{\Delta x^2}u_{i+1}{}^{k}+\frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k)}{\tau}}u_{i+1}{}^{k}+u_i^{k}-\frac{2\mu\Delta t}{\Delta x^2}u_i^{k} \tag{48}$$

$$-\frac{D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k)}{\tau}}u_i^{k}-\frac{\lambda\Delta t}{2\Delta x}u_{i-1}{}^{k}+\frac{\mu\Delta t}{\Delta x^2}u_{i-1}{}^{k}+\frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k)}{\tau}}u_{i-1}{}^{k}+\Delta t(u_w-u_i^k)$$

Then, we can collect the like terms together by using the above Description:

$$\left(\underbrace{-\frac{D\Delta t^2}{2\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k+1)}{\tau}}}_{u_{i-1}^{k+1}}\right)+\left(\underbrace{1+\frac{D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k+1)}{\tau}}+\underbrace{\Delta t(-1+2u_i^k)}_{nonlinear\,term}}_{u_i^{k+1}}\right)+\left(\underbrace{-\frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k+1)}{\tau}}}_{u_{i+1}^{k+1}}\right)$$

$$=\left(\underbrace{-\frac{\lambda\Delta t}{2\Delta x}+\frac{\mu\Delta t}{\Delta x^2}+\frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k)}{\tau}}}_{u_{i-1}^{k}}\right)+\left(\underbrace{1-\frac{2\mu\Delta t}{\Delta x^2}-\frac{D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k)}{\tau}}}_{u_i^{k}}\right)$$

$$+\left(\underbrace{\frac{\lambda\Delta t}{2\Delta x}+\frac{\mu\Delta t}{\Delta x^2}+\frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k)}{\tau}}}_{u_{i+1}^{k}}\right)+\underbrace{(u^2)_i^k}_{nonlinear\,term}+\Delta t(u_w-u_i^k) \tag{49}$$

So,

- $$u_{i-1}^{k+1}(coefficient)=\left(\underbrace{-\frac{D\Delta t^2}{2\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k+1)}{\tau}}}_{u_{i-1}^{k+1}}\right),$$

- $$u_i^{k+1}(coefficient)=\left(\underbrace{1+\frac{D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k+1)}{\tau}}+\underbrace{\Delta t(-1+2u_i^k)}_{nonlinear\,term}}_{u_i^{k+1}}\right) \tag{50}$$

- $$u_{i+1}^{k+1}(coefficient)=\left(\underbrace{-\frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k+1)}{\tau}}}_{u_{i-1}^{k+1}}\right)$$

(D) Identify the constant terms with letters.

$$let\quad r=\frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k+1)}{\tau}},\quad r_1=\frac{\mu\Delta t}{\Delta x^2},\quad r_2=\frac{\lambda\Delta t}{2\Delta x},\ r_3=\frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N}e^{-\frac{t(N+1)-t(k)}{\tau}}$$

Then,

$$\begin{aligned}&-ru_{i-1}{}^{k+1}+(1+2r)u_i{}^{k+1}-ru_{i+1}{}^{k+1}=\Delta t.u_i{}^k-\Delta t.(u_i{}^k)^2+\left(r_1-r_2+r_3\right)u_{i-1}{}^k\\&+\left(1-2r_1-2r_3\right)u_i{}^k+\left(r_1+r_2+r_3\right)u_{i+1}{}^k+\underbrace{(u^2)_i^k}_{nonlinear\,term}+\Delta t(u_w-u_i^k)\end{aligned} \tag{51}$$

$$\begin{cases}a=-r\\ b=1+2r+\underbrace{\Delta t(-1+2u_i^k)}_{nonlinear\,term}\\ c=-r\\ d=\Delta t.u_i{}^k-\Delta t.(u_i{}^k)^2+(r_1-r_2+r_3)u_{i-1}{}^k+(1-2r_1-2r_3)u_i{}^k+(r_1+r_2+r_3)u_{i+1}{}^k+\underbrace{(u^2)_i^k}_{nonlinear\,term}+\Delta t(u_w-u_i^k)\end{cases}$$

Table 3.2:- Coefficient of the right end point rule, Left end point rule and Trapezoidal end point rule

		Coefficients of each methods			
No	Name of methods	Coefficients of A(coefficient u_{i-1}^{k+1})	Coefficients of B(coefficient u_i^{k+1})	Coefficients of C(coefficient u_{i+1}^{k+1})	Coefficients of D
1.	Left endpoint rule		1		$\Delta t.u_i^k - \Delta t.(u_i^k)^2 + \underbrace{\left(-\frac{\lambda\Delta t}{2\Delta x}+\frac{\mu\Delta t}{\Delta x^2}+\frac{D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}}\right)}_{u_{i-1}^k} + \underbrace{\left(1-\frac{2\mu\Delta t}{\Delta x^2}-\frac{2D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}}\right)}_{u_i^k}$
2.	Right endpoint rule	$\left(\underbrace{-\frac{D\Delta t^2}{\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}}}_{u_{i-1}^{k+1}}\right)$	$\left(\underbrace{1+\frac{2D\Delta t^2}{\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}}}_{u_i^{k+1}}\right)$	$\left(\underbrace{-\frac{D\Delta t^2}{\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}}}_{u_{i+1}^{k+1}}\right)$	$D = \Delta t.u_i^k - \Delta t.(u_i^k)^2 + \left(-\frac{\lambda\Delta t}{2\Delta x}+\frac{\mu\Delta t}{\Delta x^2}\right)u_{i-1}^k + \left(1-\frac{2\mu\Delta t}{\Delta x^2}\right)u_i^k + \left(\frac{\lambda\Delta t}{2\Delta x}+\frac{\mu\Delta t}{\Delta x^2}\right)u_{i+1}^k$
3.	Trapezoidal rule	$\left(\underbrace{-\frac{D\Delta t^2}{2\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}}}_{u_{i-1}^{k+1}}\right),$	$\left(\underbrace{1+\frac{D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}}}_{u_i^{k+1}}\right)$	$\left(\underbrace{-\frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}}}_{u_{i-1}^{k+1}}\right)$	$D = \Delta t.u_i^k - \Delta t.(u_i^k)^2 + (q_1 - q_2 + q_3)u_{i-1}^k + (1-2q_1-2q_3)u_i^k + (q_1+q_2+q_3)u_{i+1}^k$ Where; $\begin{cases} q = \frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}}, \\ q_1 = \frac{\mu\Delta t}{\Delta x^2}, \\ q_2 = \frac{\lambda\Delta t}{2\Delta x}, \\ q_3 = \frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}} \end{cases}$

CHAPTER FOUR

Numerical solution of the memory term

In this study, we shall see how linear reaction term, nonlinear reaction term, Heaviside function source term and memory term affects fluid flow. The flow is more affected by the nonlinear term than the others. A solution to a differential equation is said to be stable if a slightly different solution that is close to it remains close for nearby values. In this study we looked four different methods of discritizing the governing equation of the generalized Fisher Kolomogrov-Petrovskii-Piskunov (FKPP), equation.

The three different methods are: Left end point rule, Right end point rule and trapezoidal end point rule. Therefore, the analyses of all the three methods are showed in this chapter. In chapter three, we discritized the FKPP equation using the three different methods and convert them to a dimensionless form. In this chapter, we introduce a small disturbance on the flow equations and discuss the extension to account for turbulence. We will present the interpretation of analysis results and compare them with those found in literature.

4.1 What is memory in numerical computation?

The following illustrative examples on different things or activities shows what memory is? It indicates our association to the things and tries to observe what memory is.

Examples:

- Sneezing /cough / being in a group don’t cause diseases immediately,
- Plastic stretching after shrinking,
- Shower water becomes hot slowly being opened after a while, etc.

The following illustrative examples on different things or activities shows what memory (delay) is. It indicates our association to the things and tries to observe what memory is.

Fig4.1 Sneezing /cough / being in a group don't cause diseases immediately

Fig4.2 Plastic stretching after shrinking

Fig4.3 Shower water becomes hot slowly being opened after a while

4.2 Discritization of memory term by Left end point rule

This is used to approximate the shape as a rectangle as depicted in the figure below. The area of a rectangle is computed using height and width. There is only one height and width.

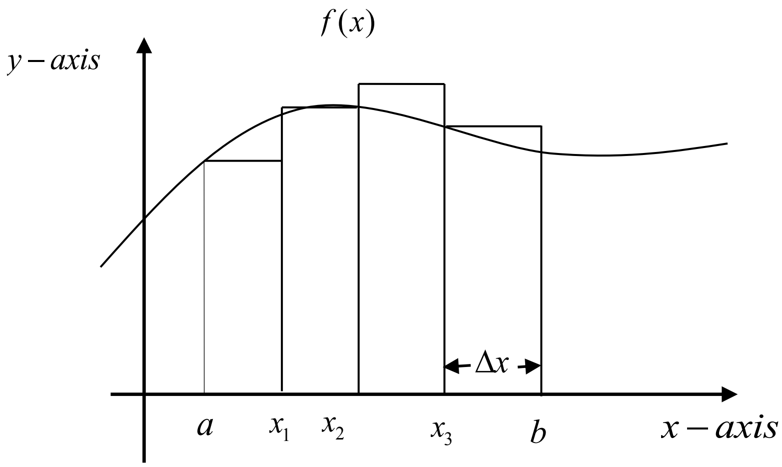


Fig 4.4 A graph shows how the memory term is discredited by LHS

For this rule, the left side of the right from the subinterval is considered to create the rectangle. If the gap is an ascending graph the rectangle will be subscribed by the area under the graph and if the graph is a descending, the rectangle will subscribe the area under the graph.

In both cases the rectangle approximates the area with an error. In an equation form:

$$\int_a^b f(x)dx = f(a).(x_1 - a) \tag{52}$$

When the computation is for the whole graph, it is the sum of individual rectangles and it will be of the following general formula:

$$\int_a^b f(x)dx = \Delta x \sum_{i=1}^{n} f(x_{i-1}) \tag{53}$$

We need to discretize the integral sign according to equation (5).

The following discretization show how this can be done.

$$\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds = \Delta t \sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}} u_{xx}(x,t_k)$$

4.3 Discritization of memory term by Right end point rule

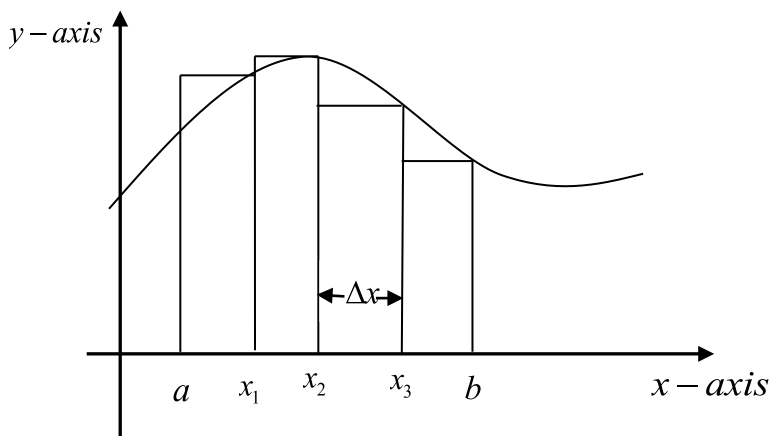


Fig 4.5 A graph shows how the memory term is discredited by RHS

For the right end rule, the right side of the interval functional value is used as a height to approximate the shape into a rectangle. The equation now will have the following form:

$$\int_a^{x_1} f(x)dx = f(x_1).(x_1 - a) \tag{54}$$

The general formula:

$$\int_{x_0}^{x_n} f(x)dx = \Delta x \sum_{i=1}^{n} f(x_{i+1}) \tag{55}$$

Substituting the general formula for the given example:

$$\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds = \Delta t \sum_{k=1}^{N} \left(e^{-\frac{t(N+1)-t(k+1)}{\tau}} u_{xx}(x,t_{k+1}) \right) \tag{56}$$

4.4 Discritization of memory term by Trapezoidal end point rule

In calculus we learned that the integrals are (signed) areas and can be approximated by sums of the smaller areas, such as the areas of rectangles. As depicted the picture below the integration of a function f(x) b/n a and b is represented by the area under the function curve and the point a from the left and point b from the right and also the x-axis from below.

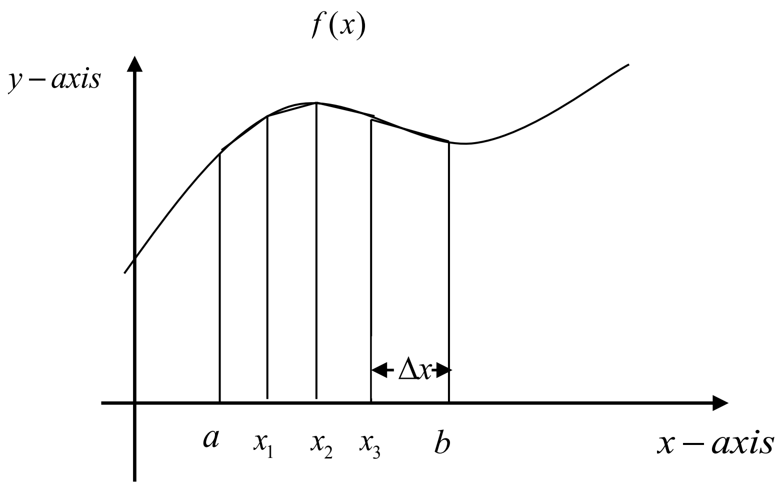


Fig 4.6 A graph shows how the memory term is discritized by Trapezoidal end point rule

To find the approximated area under the curve we can split the curve as it is show in the figure in equal parts. If we considered the first portion (i.e. b/n a and x_1) and the area the picture can be approximated to a trapezoid. The width of the trapezoid $x_1 - a$, and the length of the two parallel sides are $f(x_1)$ & $f(a)$. Calculating the area under the curve bounded under (a, x_1) (for a segment) is given by:

$$\int_a^{x_1} f(x)dx = \frac{1}{2}(f(a) + f(x_1)).(x_1 - a) \tag{57}$$

Then, equation (24a) can generally be written for the total area under the curve.

$$\int_a^b f(x)dx = \frac{1}{2}\sum_{i=1}^{N}\{(f(x_i) + f(x_{i+1}))\} \tag{58}$$

Extending by choosing points $\{x_1\}$ that subdivided (a,b) into n partition in the following manner:

$$a = x_0 < x_1 < \ldots < x_{n-1} < x_n = b \tag{59}$$

The subinterval (x_{i-1}, x_i) determine the width Δx_i where $\Delta x = \frac{b-a}{n}$ since we are creating equal subspaces. For the height, the function $f(x_{i-1})$ and $f(x_i)$ are considered.

The general approximation of the area under the interval (a,b) is given by:-

$$\int_{x_0}^{x_n} f(x)dx = \frac{\Delta t}{2}\sum_{i=1}^{n}(f(x_{i-1}) + f(x_i)) \tag{60}$$

Using this formula for the given integral in the example will give us the following form (k supposed to be from 0 to N, but **MATLAB** starts array index from 1). So accordingly k is run from 1 to N + 1 implied in k and k + 1 instead of k – 1 and k.

$$\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds = \frac{\Delta t}{2}\sum_{k=1}^{N}\left(e^{-\frac{t(N+1)-t(k)}{\tau}} u_{xx}(x,t_k) + e^{-\frac{t(N+1)-t(k+1)}{\tau}} u_{xx}(x,t_{k+1}) \right) \tag{61}$$

CHAPTER FIVE

Numerical solution of an Intrgro-transport PDE (FKPP equation)

5.1 Discritization of convection diffusion transport model

Generic transport equation for a scalar u

$$\boxed{\underbrace{\frac{\partial(\rho u)}{\partial t}}_{Transient}+\underbrace{\vec{\nabla}.(\rho\vec{u}u)}_{Convection}=\underbrace{\vec{\nabla}.(F\vec{\nabla}u)}_{Diffusion}+\underbrace{S_u}_{Source}} \quad (6)$$

This PDE describes transport phenomena such as heat transfer, mass transfer, momentum transfer, etc...

Incompressible flow: $\frac{\partial(u)}{\partial t}+\vec{\nabla}.(\vec{u}u)=D\nabla^2 u+S_u$

<u>Special cases:</u>

Diffusion only: $\frac{\partial(u)}{\partial t}=D\nabla^2 u$ (unsteady heat conduction)

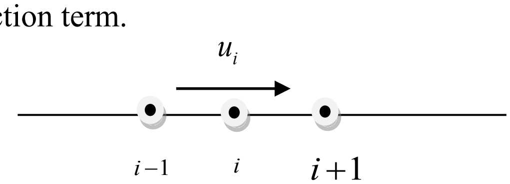
Convection only: $\frac{\partial(u)}{\partial t}+\vec{\nabla}.(\vec{u}u)=0$

In 1D, $\frac{\partial(u)}{\partial t}+\frac{\partial}{\partial x}(uu)=0$

If u=const, we have a wave equation (first order)

However, in convection related problems, $\vec{u}$ is a function of time and space, in this case the upwind scheme needs to used for treating the convection term.

Upwind Scheme: for $\frac{\partial}{\partial x}(u)=0$ at x_i

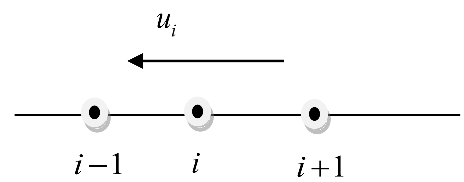


If $u_i>0$, convective flow moves from left to right, in this case the convection at x_i should be more influenced by what happen at upstream x_i and less influenced by downstream x_{i+1}

To reflect this, we use $\frac{\partial}{\partial x}(u)_i=\frac{(u)_i-(u)_{i-1}}{\Delta x}$

If $u < 0,$ flows moves from right to left,

We use $\quad \frac{\partial}{\partial x}(u\Phi)_i = \frac{(u\Phi)_{i+1} - (u\Phi)_i}{\Delta x}$



If we define a sign function for u_i,

$$\begin{cases} 1 & if\ u_i > 0 \\ -1 & if\ u_i < 0 \\ 0 & if\ u_i = 0 \end{cases}$$

Then we have a compact form for the convection term:

$$\frac{\partial}{\partial x}(u\Phi)_i = \frac{1}{2}(1+e_u)\frac{(u\Phi)_i - (u\Phi)_{i-1}}{\Delta x} + \frac{1}{2}(1-e_u)\frac{(u\Phi)_{i+1} - (u\Phi)_i}{\Delta x}$$
$$\frac{\partial}{\partial x}(u\Phi)_i = \frac{(1-e_u)(u\Phi)_{i+1} + 2e_u(u\Phi)_i - (1+e_u)(u\Phi)_{i-1}}{\Delta x}$$

Hence, an upwind scheme for convection only PDE follows:

$$\frac{\Phi_i^{k+1} - \Phi_i^k}{\Delta t} + \frac{(1-e_u)(u\Phi)_{i+1}^k + 2e_u(u\Phi)_i^k - (1+e_u)(u\Phi)_{i-1}^n}{\Delta x} = 0$$

Where e_u is a sign function of u_i,

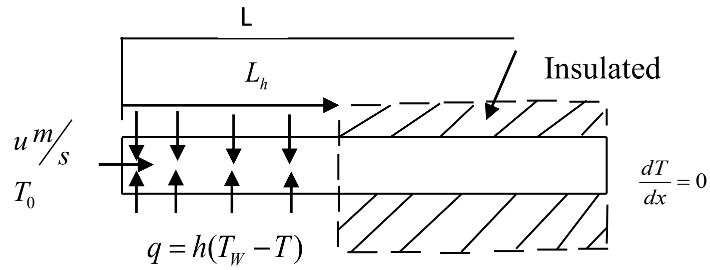
The above FDE is an explicit scheme; an implicit upwind scheme can also be obtained

The Governing equation is given as follows:

$$\frac{\partial \phi}{\partial t} + \frac{\partial}{\partial x}(u\phi) = D\frac{\partial^2 T}{\partial x^2} + S(\phi) \tag{6}$$

Let us consider heat transfer t fluid flowing in tube with a small diameter.

The first L_h meters of tube are heated by raising the temperature outside the tube to T_W, which results in a Flux of $q = h(T_w - T)$ into the fluid. (h is a heat transfer coefficient), the rest of ($L-L_h$) is insulated.



I.c. $T(x, t = 0) = T_0$ everywhere

An accurate model for this problem is a 2D transport model with T varies along both x-direction and radial direction. However, for a small diameter tube, the variation in radial can be neglected, and the heat flux q can be treated as a source term.

1D Generic Equation Model:

$$\frac{\partial T}{\partial t} + \frac{\partial}{\partial x}(uT) = \alpha\frac{\partial^2 T}{\partial x^2} + \lambda(T_W - T)H(x) \tag{7}$$

Where;

$\alpha -$ fluid diffusivity,
$\lambda -$ a coefficient related to heat flux into the fluid,

$$H(x) = \begin{cases} 1, & L_h > x \geq 0 \\ 0 & otehwise \end{cases}$$

In the simulation we set:

$$L = 1m, \quad L_h = 0.5m, \quad T_0 = 20^0{}_c, \quad T_w = 80^0{}_c \quad \alpha = 0.01\,m^2/s, \quad \lambda = 1\,m/s$$

Initial Condition:

$$T(x,t=0)=T_0;$$

Boundary Condition:

At inlet, $T(x=0,t)=T_0;$

At outlet, we set: $\dfrac{\partial T}{\partial x}(x=L,t)=0;$ the thermal conductivity is neglected.

The FDE: Using an implicit FTCS scheme.

$$\frac{T_i^{n+1}-T_i^n}{\Delta t}+u\frac{T_{i+1}{}^{n+1}-T_{i-1}{}^{n+1}}{2\Delta x}=\alpha\frac{T_{i+1}{}^{n+1}-2T_i^{n+1}+T_{i-1}{}^{n+1}}{\Delta x^2}+\lambda(T_W-T_i^n)H(x_i) \quad (8)$$

Since u = constant, we also can use the upwind scheme as follows:

$$\frac{T_i^{n+1}-T_i^n}{\Delta t}+u\frac{T_i^{n+1}-T_{i-1}{}^{n+1}}{\Delta x}=\alpha\frac{T_{i+1}{}^{n+1}-2T_i^{n+1}+T_{i-1}{}^{n+1}}{\Delta x^2}+\lambda(T_W-T_i^n)H(x_i) \quad (9)$$

The source term is treated explicitly.

Rearrange the upwind FDE:

$$-\left[\frac{\alpha\Delta t}{\Delta x^2}+\frac{u\Delta t}{\Delta x}\right]T_{i-1}{}^{n+1}+\left[1+\frac{u\Delta t}{\Delta x}+2\left(\frac{\alpha\Delta t}{\Delta x^2}\right)\right]T_i^{n+1}-\left(\frac{\alpha\Delta t}{\Delta x^2}\right)T_{i+1}{}^{n+1}=T_i^n+\lambda\Delta t(T_W-T_i^n)H(x_i) \quad (10)$$

For the boundary node: $i=0,\quad T_0^{n+1}=T_0$

$i=nx,\quad T_{nx}{}^{n+1}=T_{nx-1}{}^{n+1}$

In the simulation we set:

The above set of linear equations can be solved by TDMA.

Where; $A=-\left(\dfrac{\alpha\Delta t}{\Delta x^2}+\dfrac{u\Delta t}{\Delta x}\right),\quad B=\left(1+\dfrac{u\Delta t}{\Delta x}+2\left(\dfrac{\alpha\Delta t}{\Delta x^2}\right)\right)$

$C=-\left(\dfrac{\alpha\Delta t}{\Delta x^2}\right),$ and $D=T_i^n+\lambda\Delta t(T_W-T_i^n)H(x_i)$

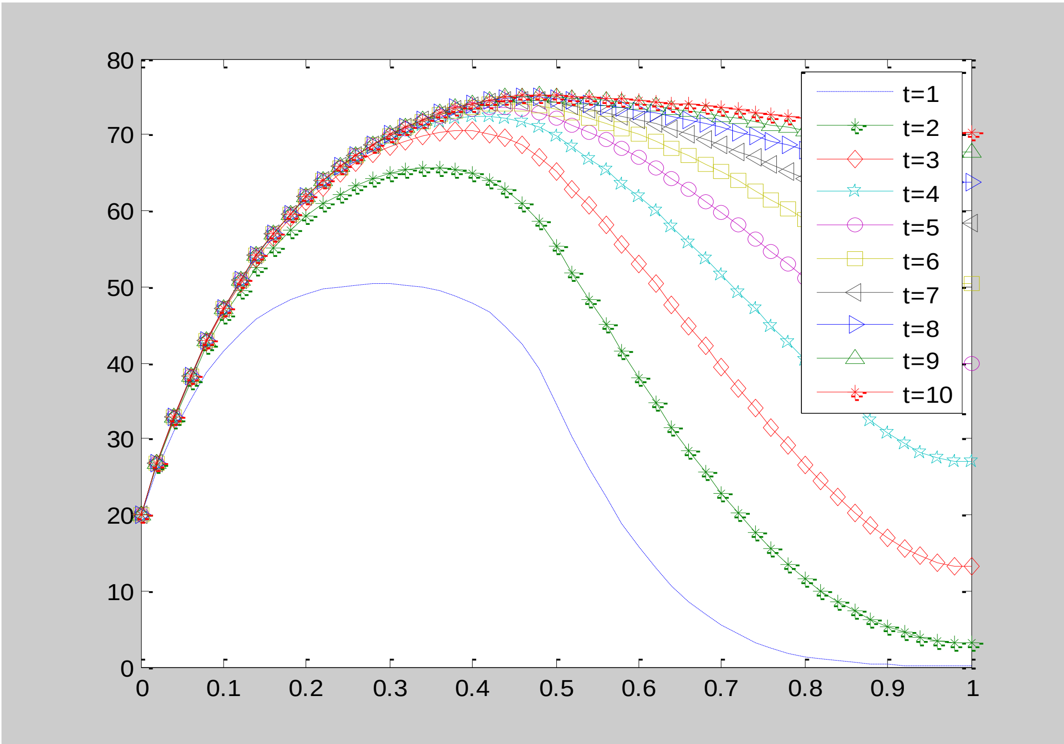


Fig 5.1 1D Generic transport equation with Heaviside function

Note: The above graph is Dirichlet on the LHS, Neumann on the RHS and the effect of Heaviside function is shown blow by the following figure.

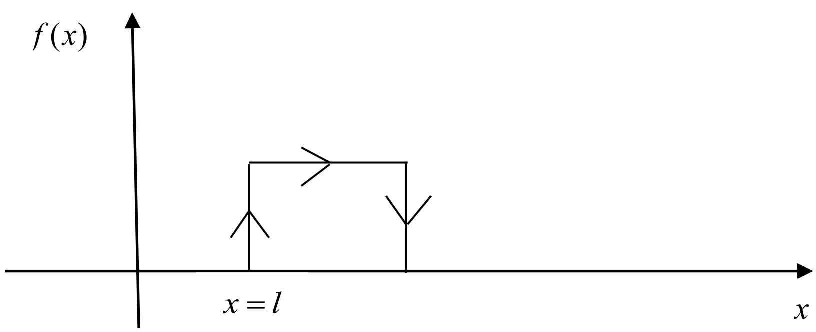


Fig.5.2 Graph of Heaviside function

The above graph shows what a Heaviside function is. For example when we put on the light is come and when we put off the light goes, so this can be a best example to explain Heaviside function.

5.2 Consideration for nonlinear reaction, convection, diffusion and memory term

The following example will used to demonstrate the graph of the solution.

The Governing equation is:

$$u_t = u(1-u) + \lambda u_x + \mu u_{xx} + \frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds \tag{62}$$

Where $(x,t) \in (0,10)x(0,1]$ subject to the initial condition

$$u(x,0) = \begin{cases} 1, & 0 \le x \le 5, \\ 0, & 5 \le x \le 10 \end{cases}$$

and the boundary conditions

$$u(x,0) = 1,\ u(10,t) = 0,\ where\ t \in (0,1].$$

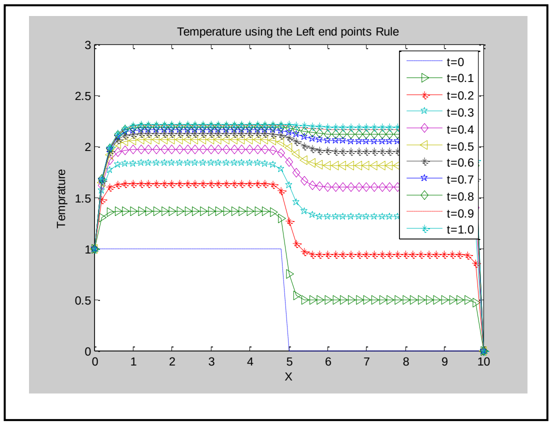


Fig 5.3 Graph of 2D nonlinear reaction, convection, diffusion and memory term

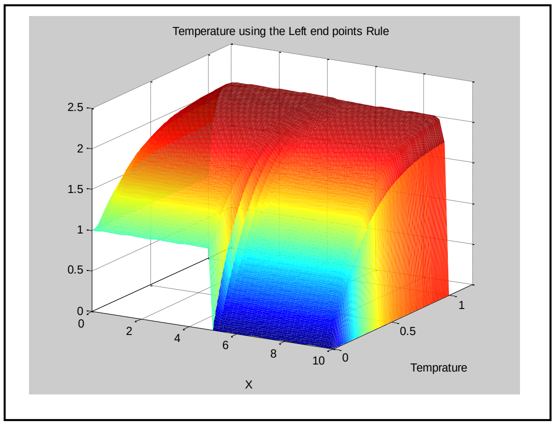


Fig 5.4 Graph of 3D nonlinear reaction, convection, diffusion and memory term

Analysis of Left end point rule

The outputs of the left end point rule give rise to the same diagram both in 2 dimensional and 3 dimensional diagrams when we add memory term, convection source term, and linear reaction term, quadratic reaction term and fisher reaction term.

5.3 Consideration for convection, diffusion and memory term

The governing equation is:

$$u_t = \lambda u_x + \mu u_{xx} + \frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds \tag{64}$$

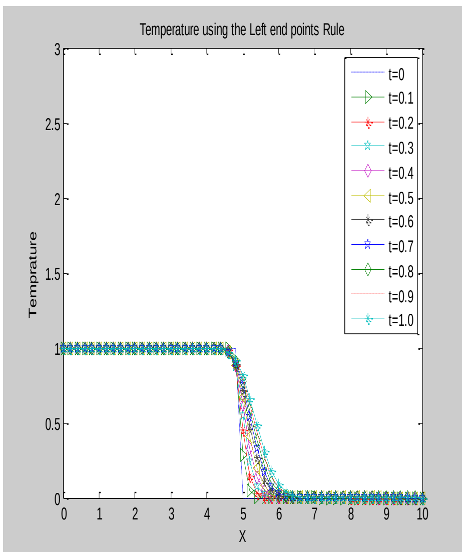


Fig 5.5 Graph of 2D convection, diffusion and memory term

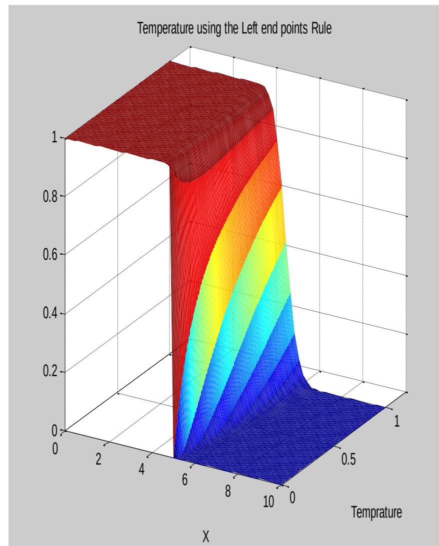


Fig 5.6 Graph of 3D convection, diffusion and memory term

Analysis of right end point rule

The outputs of the right end point rule give rise to the same diagram both in 2 dimensional and 3 dimensional diagrams when we add memory term, convection source term, and linear reaction term, quadratic reaction term and fisher reaction term.

5.4 Consideration for linear reaction, convection, diffusion, memory and convection source term

The governing equation is:

$$u_t = u + \lambda u_x + \mu u_{xx} + \frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds + \lambda(T_w - T) \tag{65}$$

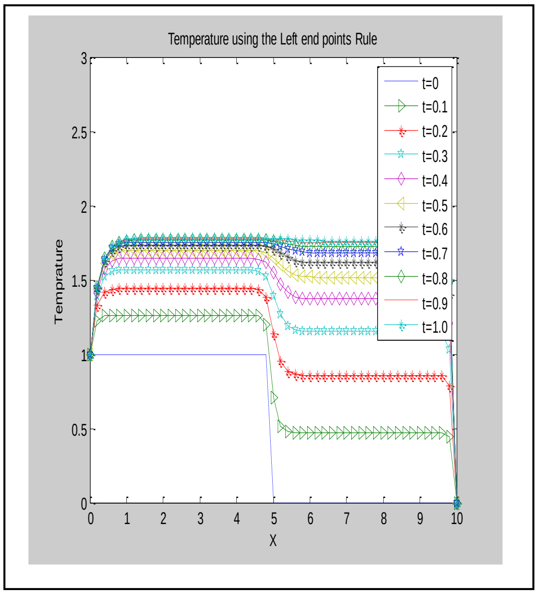


Fig 5.7 Graph of 2D linear reaction, convection, diffusion and memory term

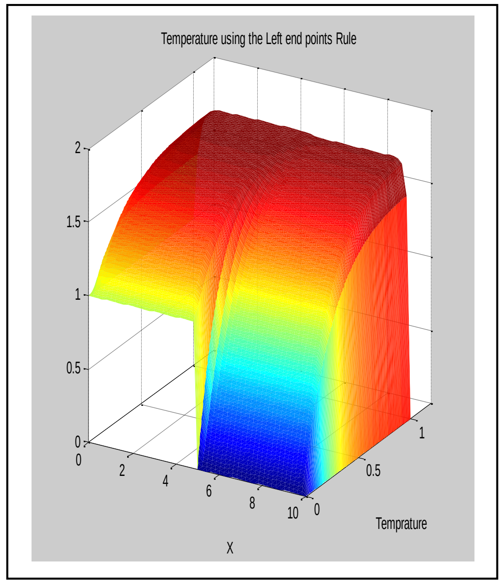


Fig 5.8 Graph of 3D linear reaction, convection, diffusion and memory term

The outputs of the Trapezoidal end point rule give rise to the same diagram both in 2 dimensional and 3 dimensional diagrams when we add memory term, convection source term, linear reaction term, quadratic reaction term and fisher reaction term.

5.5 Consideration for nonlinear reaction, convection, diffusion, memory and convection source term

The governing equation is:

$$u_t = u^2 + \lambda u_x + \mu u_{xx} + \frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds + \lambda(T_w - T) \tag{66}$$

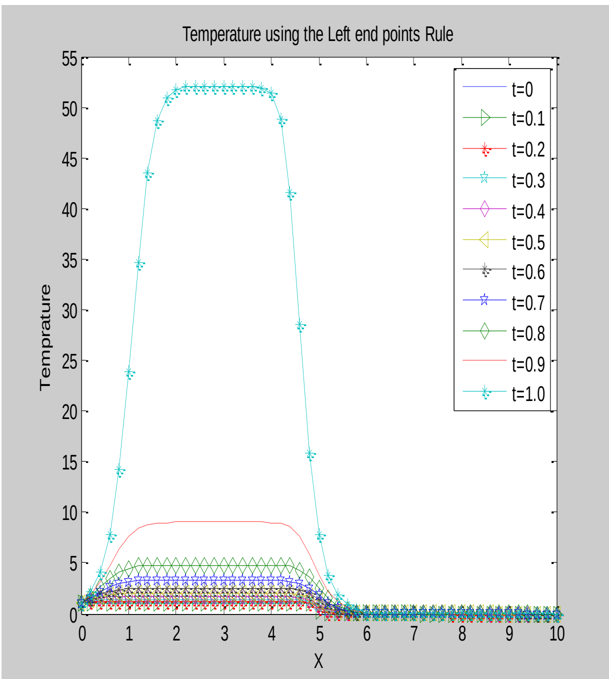


Fig 5.9 Graph of 2D quadratic reaction, convection, diffusion and memory term

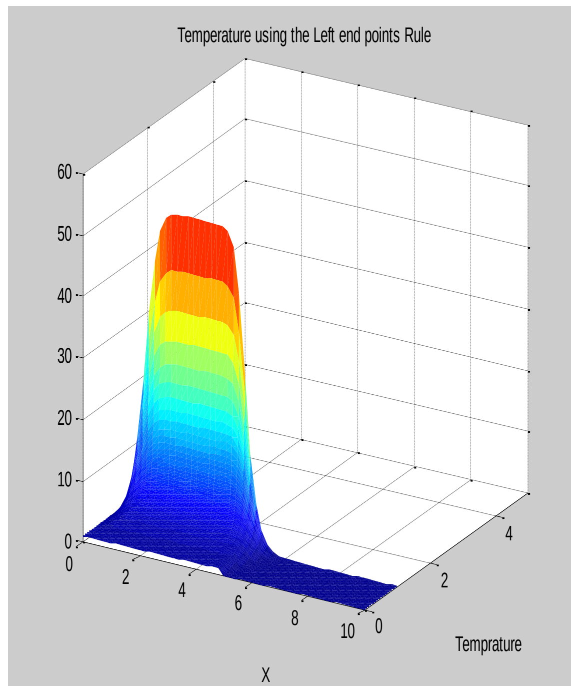


Fig 5.10 Graph of 3D quadratic reaction, convection, diffusion and memory term

The outputs of the left end point rule give rise to the same diagram both in 2 dimensional and 3 dimensional diagrams when we add memory term, convection source term, linear reaction term, quadratic reaction term and fisher reaction term, but when we add the non-linear (quadratic) term there is a big change in 2 dimensional and 3 dimensional diagram.

CHAPTER SIX

SUMMARY, CONCLUSION AND RECOMMENDATIONS

6.1 Summary

A study of an incompressible one dimensional flow in a channel with one porous wall is presented in this research. We presents one dimensional FKPP with linear, memory, non linear and convection source term driven, incompressible laminar flow in one dimensional channel flow with upper permeable wall and no slip condition while the lower wall is impermeable. The governing continuity and momentum equations together with the associated boundary conditions are first reduced to a set of self similar non linear coupled ordinary differential equations using similarity transformations. Then we solved the ordinary differential equation by perturbation series method and the boundary value problem. Some important features of the flow in terms of normal and stream wise velocities, skin friction and pressure distributions for different values of the governing parameters are analyzed, discussed and presented through graphs. I solve the FKPP by Left, Right and Trapezoidal end point rule.

The three methods (Trapezoidal, Left endpoints, and Right endpoints) give the rise to the same diagram both in 2 dimensional and 3 dimensional diagrams. Among the three methods, when I add the non-linear term to all the three methods listed above, the flow is more affected than the others (unstable). The findings show that, although there is change when we add linear reaction, convection source and memory term, but the change is not that much similar to the non-linear (quadratic) reaction term. Generally, what we can conclude from this is that, the fluid flow along one dimensional channel is affected by linear reaction term, quadratic reaction term, convection source term and memory term, but it is most affected by non-linear (quadratic) reaction term.

6.2 Conclusion

In this study we have developed a mathematical model which can be used to model among other things, the subsurface drip Irrigation. This is one of the best methods for irrigation as it helps to have an optimal use of water as it helps to retain a lot of water which may be lost through evaporation if other methods are used. It also helps to prevent diseases which can occur due to water contamination with leaves, and also pesticides and insecticides can be added in the irrigation pipes. I believe that, the use of mathematical models for irrigation systems can help to improve irrigation efficiency not only where water is scarce but also where irrigation consumes too much labor and gives extra burden to farmers. As indicated in the study, the flow of water in a pipe is affected by the linear, quadratic, convection source and memory term, although it is more affected by quadratic reaction term. So if irrigation engineer can have proper flow, it is more affected by quadratic reaction source term. If, let say there is a leakage somewhere in the pipes it will be easier for them to see changes through computer simulations and make the necessary renovations.

6.3 Recommendations

Future researchers may extend this work by performing real field experiments to show practically how the quadratic reaction term, linear reaction term, convection source term and memory term affects the fluid flow. Pressure gradient and slip parameter may be used to control the flow of water in the subsurface irrigation pipes. We believe that the application of this model will not only simplify the work of the irrigation engineers but also improving agricultural production while maintaining the optimal use of water.

References

[1] A.F. Soliman, A.M.A. El-Asyed, M.S. El-Azab; On the Numerical Solution of Partial integro-differential equations; An International Journal; Vol. 1 (p 71-80);Natural Sciences Publishing Cor.; 2012

[2] A.R. Sanderson, M.D. Meyer, R.M. Kirby, C.R. Johnson, A framework for exploring Numerical solutions of advection–reaction–diffusion equations using a GPU-based Approach, Comput. Vis. Sci. 12 (2009) 155–170.

[3] Ber- Asher,J and C.J Phene (1993)," Analysis of Surface and Subsurface Drip Irrigation using a numerical model. M.C. Cross, P.C. Hohenberg, Pattern formation outside of equilibrium, Rev. Mod. Phys. 65 (1993) 851–1112.

[4] Ber- Asher,J and C.J Phene (1993)," Subsurface Drip Irrigation Theory, Practices and Application". Volume 2 pp: 185-200, Frensio Calfornia State University.

[5] B. Block, P. Virnau, T. Preis, Multi-GPU accelerated multi-spin Monte Carlo simulations of the 2D Isingmodel, Comput. Phys. Commun. 181 (2010) 1549–1556.

[6] Cox, B.J;J.M.Hill(2011)," Flow through a circular tube with a permeable Navier slip boundary". Nanoscale research letters.

[7] Cox,S.M (1990)," Two-dimensional flow of viscous fluid in a channel with porous walls.J.Fluid Mech." Volume 227:pp 1-33.

[8] (Chow (1959), Cunge et al. (1980), Stelling & Booij (1994)). (Suharsono and Wiwatanapataphee, 2011).

[9] D.P. Playne, K.A. Hawick, Data parallel three-dimensional Cahn-Hilliard field equation simulation on GPUs with CUDA, Technical Report CSTN-073 (2009).

[10] E.O. Holzbecher (1998). Modeling Density-Driven Flow in Porous Media. Springer-Verlag, Berlin.

[11] F. Treves [(Plenum Press, NY) (1980)].

[12] I. Lagzi, D. Ueyama, Pattern transition between periodic Liesegang pattern and crystal growth regime in reaction–diffusion systems, Chem. Phys. Lett. 468 (2009) 188–192.

[13] I. Senocak, J. Thibault, M. Caylor, Rapid-response urban CFD simulations using a GPU computing paradigm on desktop supercomputers, Eighth Symposium on the Urban Environment, Phoenix Arizona, 2009, p. J19.2.

[14] In the thesis “Spectral Theory of Ordinary and Partial Linear Differential Operators on Finite Intervals” by Smith, D.A.), [University of Reading, U.K.) (June, 2011)].

[15] I.R. Epstein, K. Showalter, Nonlinear chemical dynamics: oscillations, patterns, and chaos, J. Phys. Chem. 100 (1996) 13132–13147.

[16] J. Bear and A. Verruijt (1987). Modelling Groundwater Flow and Pollution. D. Reidel Publishing Company, Dordrecht.

[17] J.D. Owens, M. Houston, D. Luebke, S. Green, J.E. Stone, J.C. Phillips, GPU computing, Proc. IEEE 96 (2008) 879–899.

[18] J. Horváth, I. Szalai, P. De Kepper, An experimental design method leading to chemical Turing patterns, Science 324 (2009) 772–775.

[19] J.W. Elder (1967). Transient convection in a porous medium. Journ. Fluid Mech., 27(3):609-623.

[20] Lax and Friedrichs in the paper “Boundary Value Problems for First Order Operators”(1965).

[21] Makinde and Osalusi (2005), Makinde (1998), Berman (1953) Suharsono, et al (2011), Cox Hill,2011)

[22] M. Januszewski, M. Kostur, Accelerating numerical solution of stochastic differential equations with CUDA, Comput. Phys. Commun. 181 (2010) 183–188.

[23] M. Garland, S. Le Grand, J. Nickolls, J. Anderson, J. Hardwick, S. Morton, E. Phillips, Y. Zhang, V. Volkov, Parallel computing experiences with CUDA, Micro IEEE 28 (2008) 13–27.

[24] O. Kolditz et al. (1998). Coupled groundwater flow and transport: 1. Verification of variable density flow and transport models. Advances in Water Resources, 21(1):27-46.

[25] P. Frolkoviˇc (1998). Consistent Velocity Approximation for Density Driven Flow and Transport. In: R. Van Keer et al. (eds.), ACOMEN'98, 603-611. Shaker Publishing, Maastricht.

[26] P. Frolkoviˇc (1998). Maximum principle and local mass balance for numerical solutions of transport equation coupled with variable density flow. Acta Mathematica Universitatis Comenianae, 67(1):137-157.

[27] P. Frolkoviˇc (1999). Discretization. In: E. Fein (ed.), d3f - A Simulator for Density-Driven Flow. Gesellschaft fuer Anlagen- und Reaktorsicherheit, Braunschweig.

[28] P. Frolkoviˇc (1998). Maximum principle and local mass balance for numerical solutions of transport equation coupled with variable density flow. Acta Mathematica Universitatis Comenianae, 67(1):137-157.

[29] Ph. Ackerer, A. Younes and R. Mose (1999). Modelling variable density flow and solute transport in porous media: 1. Numerical model and verification. Transport in Porous Media, 35(3):345-373.

[30] P. Micikevicius, 3D finite difference computation on GPUs using CUDA, Technical Report NVIDIA (2009).

[31] R.G. Belleman, J. Bédorf, S.F. Portegies Zwart, High performance direct gravitational N-body simulations on graphics processing units II: an implementation in CUDA, 13, New Astron, 2008, pp. 103–112

[32] S. Fedotov; travelling wave in a reaction–diffusion system: Diffusion with finite velocity and Kolmogorov–Petrovskii–Piskunov kinetics; Phys. Rev. E 58 (1998) 5143–5145.

[33] S. Melchionna, M. Bernaschi, S. Succi, E. Kaxiras, F.J. Rybicki, D. Mitsouras, A.U. Coskun, C.L. Feldman, Hydrokinetic approach to large-scale cardiovascular blood flow, Comput. Phys. Commun. 181 (2010) 462–472.

[34] S. Che, M. Boyer, J. Meng, D. Tarjan, J.W. Sheaffer, K. Skadron, A performance study of general purpose applications on graphics processors using CUDA, J. Parallel Distrib. Comput. 68 (2008) 1370–1380.

[35] (Schuurmans, W. et al. Model Predictive Control on Open Water Systems P.J van Overloop (1999), Overloop et al. (2001), Mareels et al. (2003), Overloop (2003a), Roos (2003)), Steenis et al. (2003), Huising (2004), Montazar & Overloop (2005), Overloop (2005a));

[36] [Van(1973), Camp (1998); Zachman Thomas (1973); Ben-Asher Phene (1993)]; but most of them are not current.

[37] V. Simek, R. Dvorak, F. Zboril, J. Kunovsky, Towards accelerated computation of Atmospheric equations using CUDA, Proceedings of the UKSim 2009: 11th International Conference on Computer Modelling and Simulation, 2009,pp. 449–454.

APPENDIX A: IMPLICIT TAYLOR SERIES EXPANSION FOR NONLINEARITY

Taylor series expansion for nonlinearity

$$[f(u)]_i^{n+1} = [f(u)]_i^n + \Delta t.[\frac{\partial}{\partial t} f(u)]_i^n \tag{1}$$

$$= [f(u)]_i^n + \Delta t.\left[\frac{\partial}{\partial u} f(u)\frac{\Delta u}{\Delta t}\right]_i^n \tag{2}$$

$$= [f(u)]_i^n + \left[\frac{\partial}{\partial u} f(u)\right]_i^n \left(u_i^{n+1} - u_i^n\right) \tag{3}$$

Let

$$\boxed{f(u) = u(1-u) = u - u^2} \tag{4}$$

Following the procedure from equations (1)-(3):

$$[u-u^2]_i^{n+1} = (u-u^2)_i^n + \left\lfloor \frac{\partial}{\partial u}(u_i^n - (u^2)_i^n) \right\rfloor (u_i^{n+1} - u_i^n) \tag{5}$$

$$= (u-u^2)_i^n + (1-2u_i^n)(u_i^{n+1} - u_i^n) \tag{6}$$

$$= u_i^n - (u^2)_i^n + (u_i^{n+1} - u_i^n - 2u_i^n u_i^{n+1} + 2(u^2)_i^n \tag{7}$$

$$= u_i^n - (u^2)_i^n + (2u^2)_i^n - u_i^n + u_i^{n+1} - 2u_i^n u_i^{n+1} \tag{8}$$

Then, equation (8) can be simplified as follows:

$$\boxed{[u-u^2]_i^{n+1} = (u^2)_i^n + (u)_i^{n+1} - (2u)_i^n (u)_i^{n+1}} \tag{9}$$

Remember to change the signs accordingly

The Governing Differential equation is:

$$u_t = u(1-u) + \lambda u_x + \mu u_{xx} + \frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds \tag{10}$$

Remember: The tridiagonal coefficient is:

$$A_i U_{i-1}^{n+1} + B_i U_i^{n+1} + C_i u_{i+1}^{n+1} = D_i u_i^n \tag{11}$$

Note: Nonlinear term is on the right hand side (RHS) of equation (10), but the coefficients of the tridiagonal matrix are on the left hand (LHS) of equation (11) except the D coefficient.

Implication: Any of the terms of the linearized reaction term (see equation 9) that belongs to $A, B\, or\, C$ coefficient must have the sign changed.

Note: The term of equation (9) that belongs to the B coefficient, that is $B_i U_i^{n+1}$ are:

$$dt.\left[(u)_i^{n+1} - (2u)_i^n (u)_i^{n+1}\right] \tag{12}$$

Thus, the coefficients of B are $dt.\left[1-(2u)_i^n\right]$, when moved to the LHS (see equation (11)) they become $dt.\left[-1+(2u)_i^n.\right]$

Note: On the other hand, the remaining term of equation (9) is $(u^2)_i^n$, belongs to the D coefficient both are on the RHS, thus the sign will not change. As always the linearized terms belong to coefficients B and D, but remember to change signs when including terms for B.

Implicit Discritizations and inclusion of nonlinear terms

Table1: The nonlinear discritization for the right end point rule with explicit and implicit method.

Before (for B)	After (for B)
$1+\frac{2D\Delta t^2}{\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}}$	$1+\frac{2D\Delta t^2}{\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}}$ $+dt.(-1+2*u(2:M,k-1))$
Before (for D)	**After (for D)**
$D=\Delta t.u_i^{\,k}-\Delta t.(u_i^{\,k})^2+\left(-\frac{\lambda\Delta t}{2\Delta x}+\frac{\mu\Delta t}{\Delta x^2}\right)u_{i-1}{}^{k}$ $+\left(1-\frac{2\mu\Delta t}{\Delta x^2}\right)u_i^{\,k}+\left(\frac{\lambda\Delta t}{2\Delta x}+\frac{\mu\Delta t}{\Delta x^2}\right)u_{i+1}{}^{k}$	$D=\left(-\frac{\lambda\Delta t}{2\Delta x}+\frac{\mu\Delta t}{\Delta x^2}\right)u_{i-1}{}^{k}$ $+\left(1-\frac{2\mu\Delta t}{\Delta x^2}\right)u_i^{\,k}+\left(\frac{\lambda\Delta t}{2\Delta x}+\frac{\mu\Delta t}{\Delta x^2}\right)u_{i+1}{}^{k}$ $+dt.u(2:M,k-1)).\wedge 2$

Table2: The nonlinear discritization for the trapezoidal end point rule with explicit and implicit method.

Before (for B)	After (for B)
$1+\frac{D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}}$	$1+\frac{D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k+1)}{\tau}}+dt.(-1+2*u(2:M,k-1))$
Before (for D)	**After (for D)**
$D=\Delta t.u_i^{\,k}-\Delta t.(u_i^{\,k})^2+\left(\underbrace{-\frac{\lambda\Delta t}{2\Delta x}+\frac{\mu\Delta t}{\Delta x^2}+\frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}}}_{u_{i-1}^k}\right)$ $+\left(\underbrace{1-\frac{2\mu\Delta t}{\Delta x^2}-\frac{D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}}}_{u_i^k}\right)$ $+\left(\underbrace{\frac{\lambda\Delta t}{2\Delta x}+\frac{\mu\Delta t}{\Delta x^2}+\frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}}}_{u_{i+1}^k}\right)$	$D=\left(\underbrace{-\frac{\lambda\Delta t}{2\Delta x}+\frac{\mu\Delta t}{\Delta x^2}+\frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}}}_{u_{i-1}^k}\right)$ $+\left(\underbrace{1-\frac{2\mu\Delta t}{\Delta x^2}-\frac{D\Delta t}{\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}}}_{u_i^k}\right)$ $+\left(\underbrace{\frac{\lambda\Delta t}{2\Delta x}+\frac{\mu\Delta t}{\Delta x^2}+\frac{D\Delta t}{2\tau\Delta x^2}\sum_{k=1}^{N} e^{-\frac{t(N+1)-t(k)}{\tau}}}_{u_{i+1}^k}\right)+dt.u(2:M,k-1)).\wedge 2$

Again let consider the Taylor series;

$$[f(u)]_i^{n+1} = [f(u)]_i^n + \Delta t.[\frac{\partial}{\partial t} f(u)]_i^n \tag{13}$$

$$= [f(u)]_i^n + \Delta t.\left[\frac{\partial}{\partial u} f(u)\frac{\Delta u}{\Delta t}\right]_i^n \tag{14}$$

$$= [f(u)]_i^n + \left[\frac{\partial}{\partial u} f(u)\right]_i^n \left(u_i^{n+1} - u_i^n\right) \tag{15}$$

So let,

$$f(u) = u^2 \tag{16}$$

Following the procedure from equations (9)-(11):

$$[f(u)]_i^{n+1} = [f(u)]_i^n + \left[\frac{\partial}{\partial t} f(u)\right]_i^n \tag{17}$$

$$= [f(u)]_i^n + \Delta t.\left[\frac{\partial}{\partial u} f(u)\frac{\Delta u}{\Delta t}\right]_i^n \tag{18}$$

$$= (u^2)_i^n + (2u_i^n)(u_i^{n+1} - u_i^n) \tag{19}$$

$$= (u^2)_i^n + 2u_i^n u_i^{n+1} - 2(u^2)_i^n \tag{20}$$

Then, equation (14) can be simplified as follows:

$$[u^2]_i^{n+1} = -(u^2)_i^n + 2u_i^n u_i^{n+1} \tag{21}$$

APENDEX B: ANALYSIS OF LEFT, RIGHT AND TRAPEZOIDAL END POINT RULE

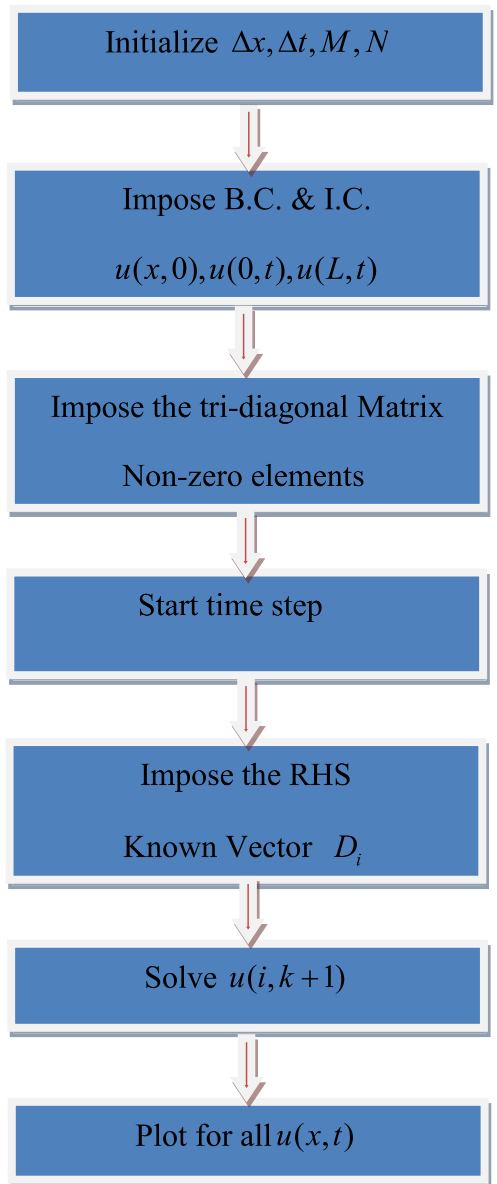


Fig 4.1: FTCS Explicit, FTCS Implicit and FTCS Crank-Nicholson coding Block Diagram

4.2 Graphical Analysis for Left end point rule (Explicit method)

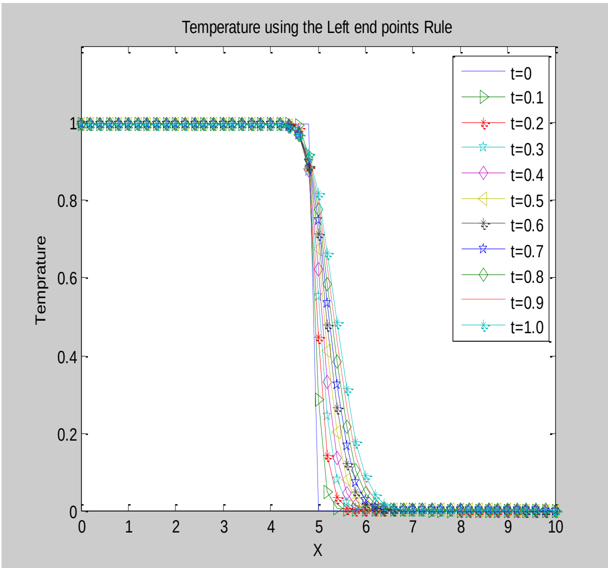


Fig 4.1 A 2-D Left memory term only

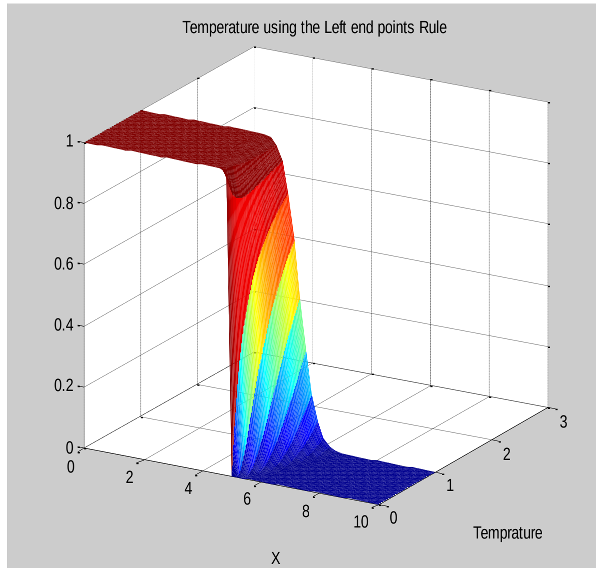


Fig 4.2 A 3-D Left memory term only

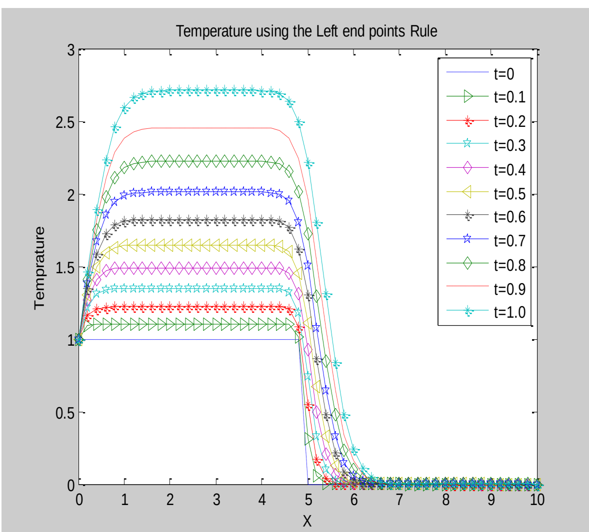


Fig 4.3 A 2-D Left memory term and linear reaction term

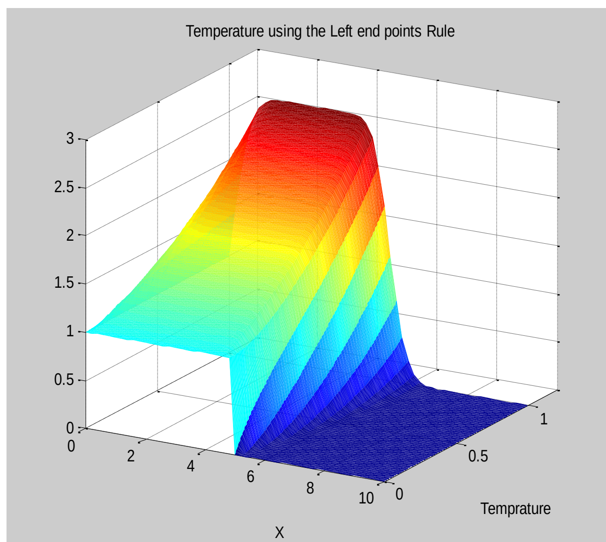
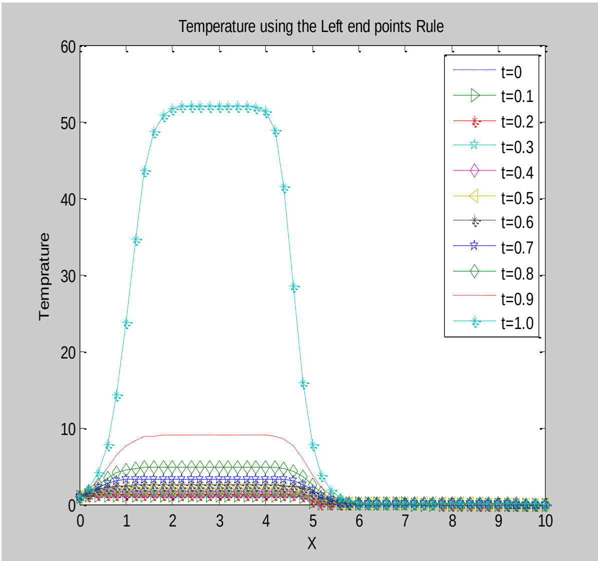


Fig 4.4 A 3-D Left memory term and linear reaction term



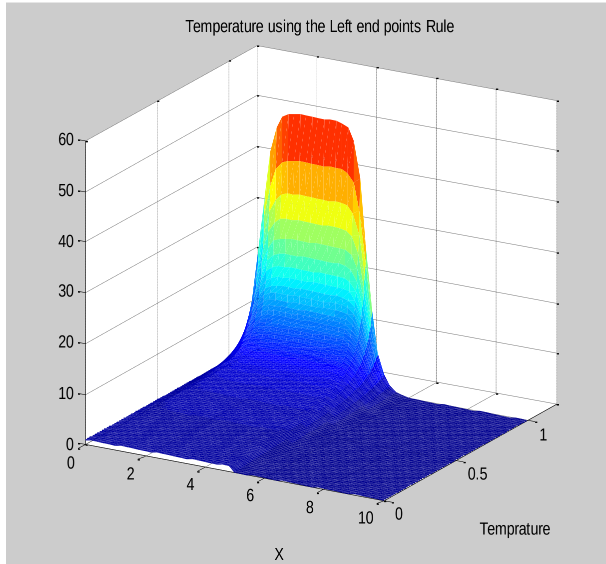
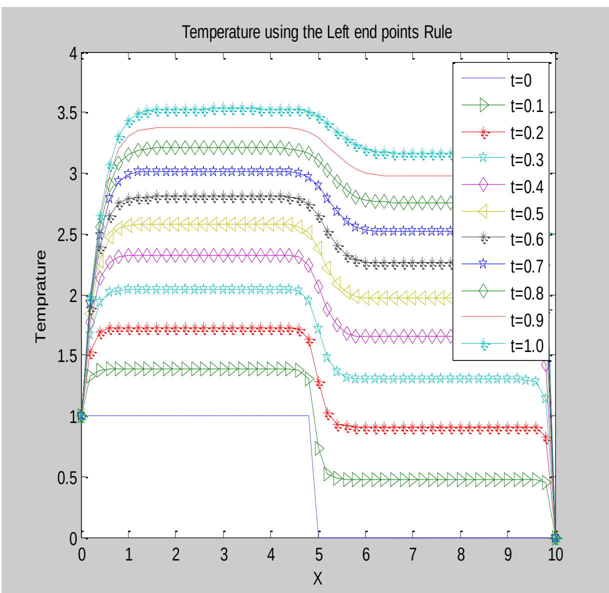


Fig 4.5 A 2-D Left memory term and quadratic reaction term

Fig 4.6 A 3-D Left memory term and quadratic reaction term



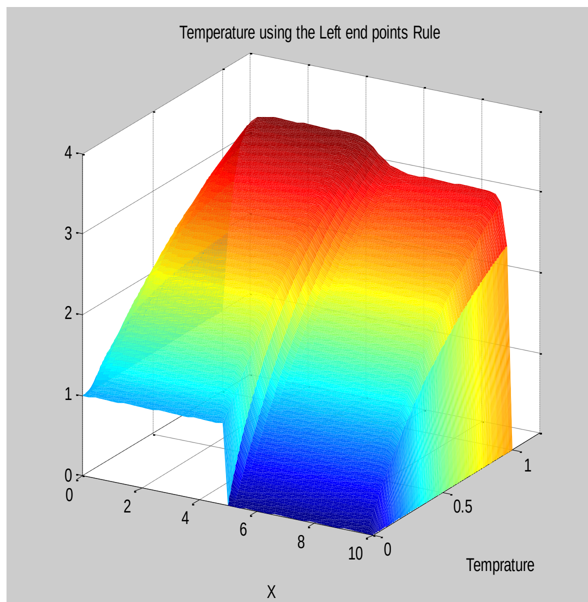


Fig 4.7 A 2-D Left memory term and convection source term

Fig 4.8 A 3-D Left memory term and convection source term

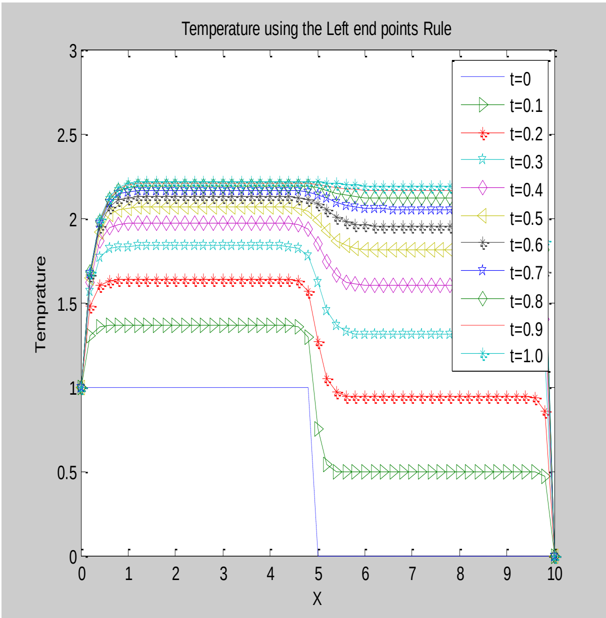


Fig 4.9 A 2-D Left memory term, convection source term and Fisher rection term

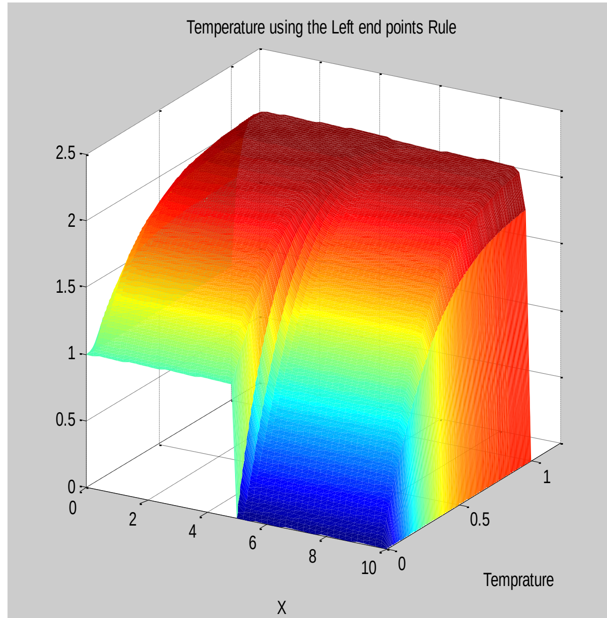


Fig 4.10 A 3-D Left memory term, convection source term and Fisher rection term

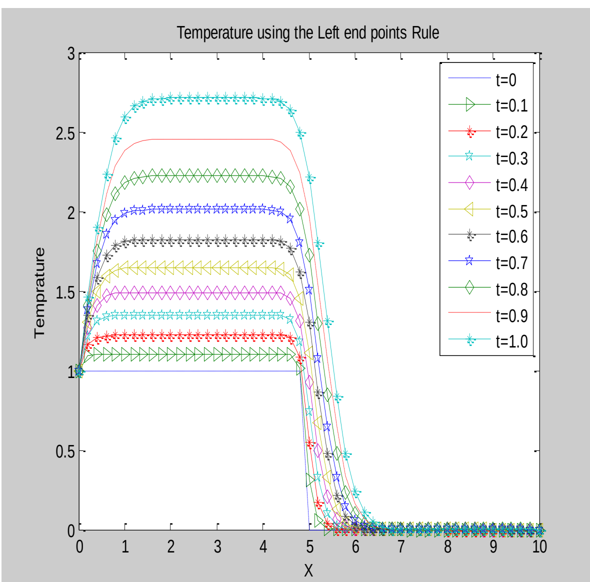


Fig 4.11 A 2-D Left Linear reaction term without memory term

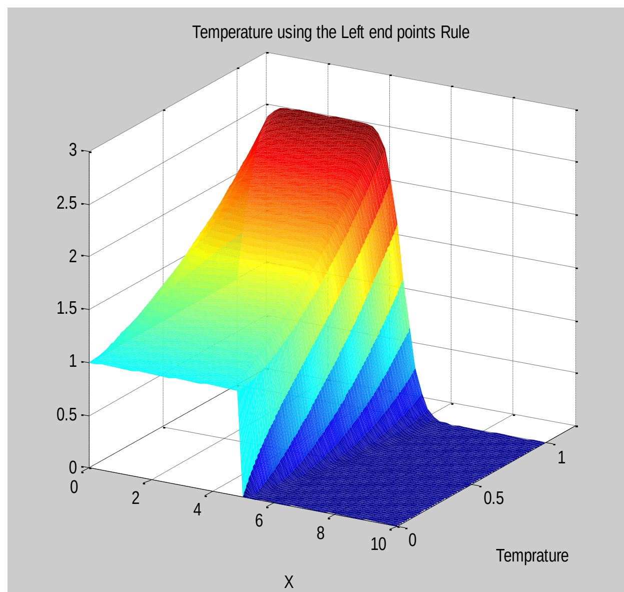


Fig 4.12 A 2-D Left Linear reaction term without memory term

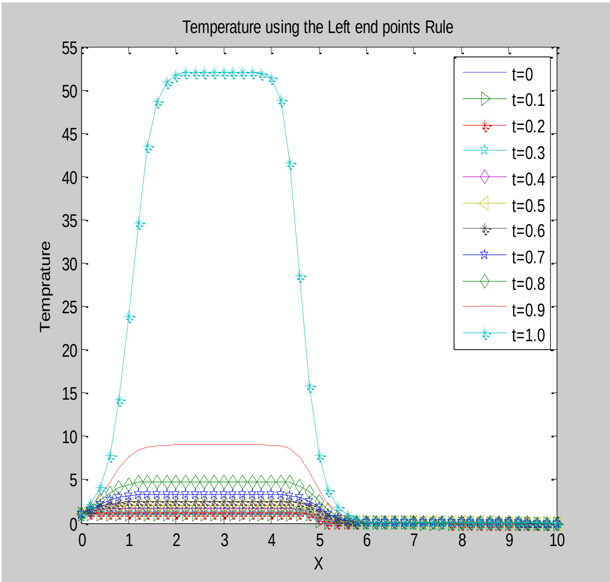


Fig 4.13 A 2-D Left Quadratic reaction term without memory term

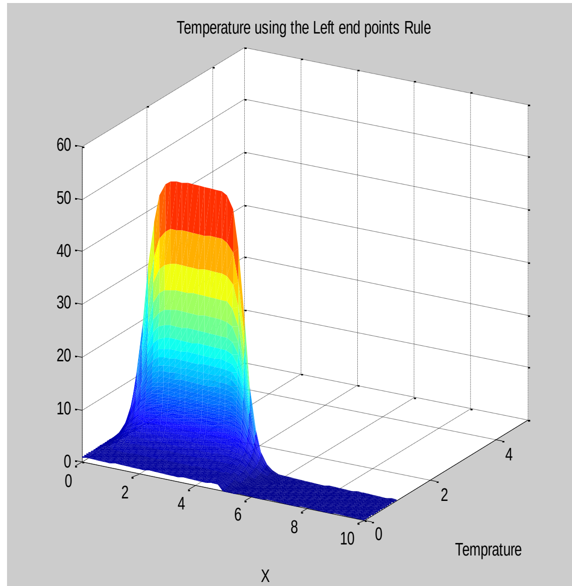


Fig 4.14 A 3-D Left Quadratic reaction term without memory term

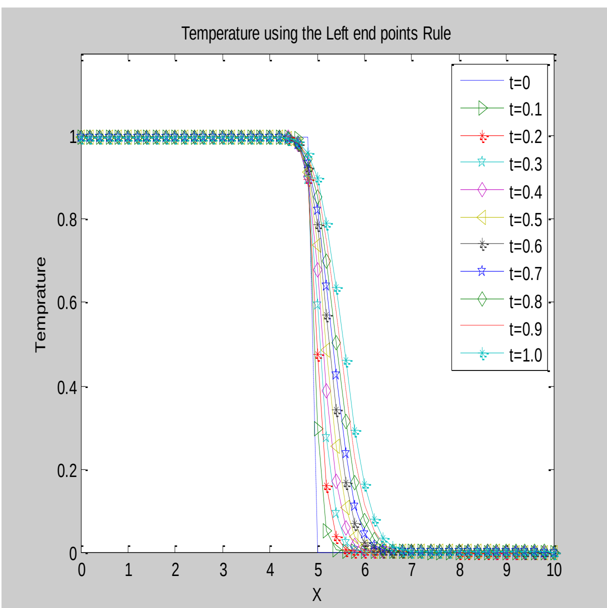


Fig 4.15 A 2-D Left convection source term without memory term

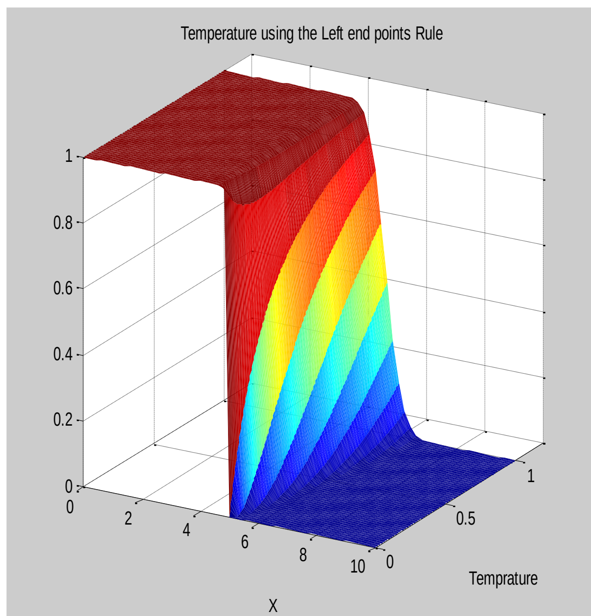


Fig 4.16 A 2-D Left convection source term without memory term

Analysis of the left end point rule

The outputs of the left end point rule give rise to the same diagram both in 2 dimensional and 3 dimensional diagrams when we add memory term, convection source term, linear reaction term, quadratic reaction term and fisher reaction term, but when we add the non-linear (quadratic) term there is a big change in 2 dimensional and 3 dimensional diagram. Therefore, the fluid flow is more disturbed (affected) when we add the quadratic (non-linear) term than the others. The physical meaning of adding non-linear term could be applying a new force to the fluid flow through the channel. When we add the memory term the fluid flow is more stable than the others and finally it approaches to steady state (no more changes as we increase time). When we add the linear reaction term the fluid flow is simple from top to bottom, which means the applied force is not more affect the fluid flow when we add linear reaction term.

Generally when we add convection source term, linear reaction term, quadratic reaction term and fisher reaction term to the left end point rule, the output give rise to the same diagram both in 2 dimensions and 3 dimensions although we couldn't deny the change in shape of the diagram. But this is not a proof that all the methods gave the same output.

4.3 Graphical Analysis for right end point rule (Implicit method)

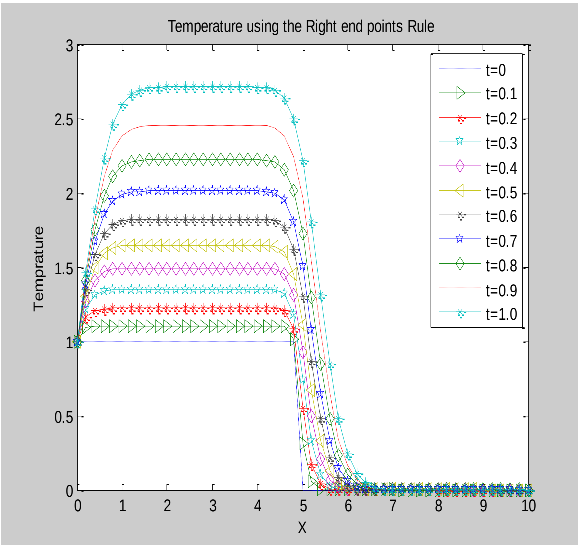


Fig 4.17 A 2-D Right memory and linear reaction term

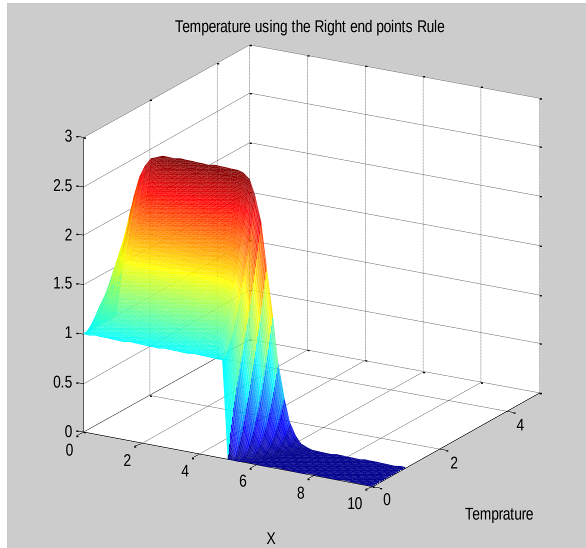


Fig 4.18 A 3-D Right memory and linear reaction term

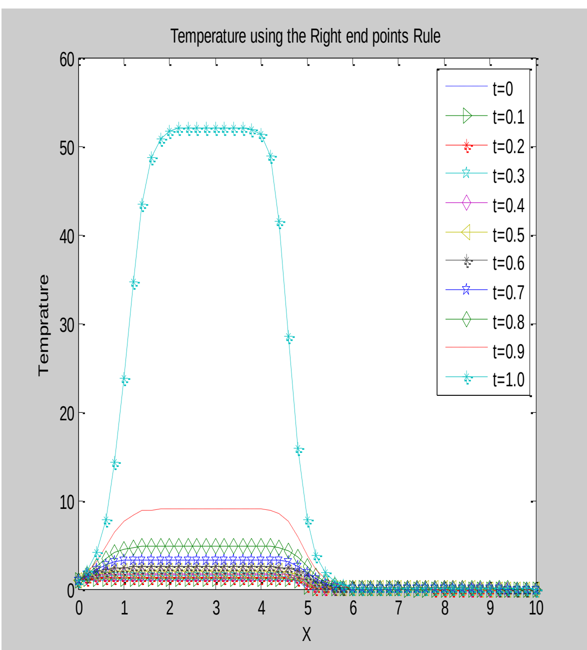


Fig 4.19 A 2-D Right memory and quadratic reaction term

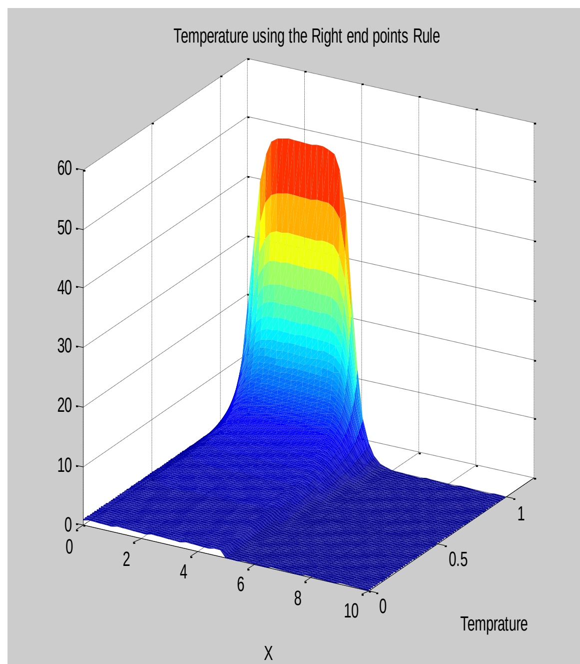


Fig 4.20 A 3-D Right memory and quadratic reaction term

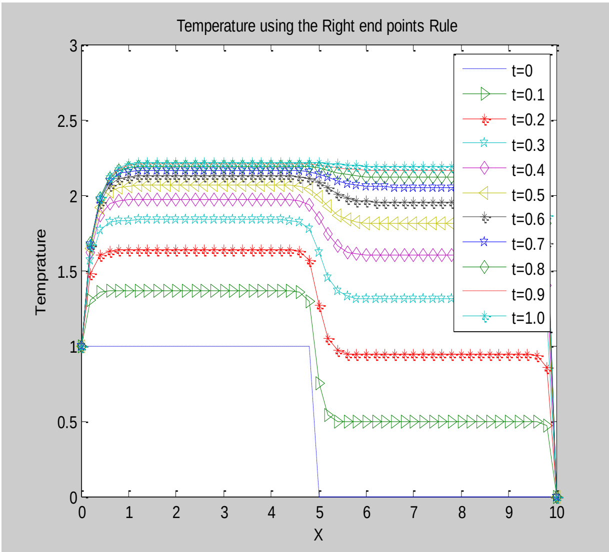


Fig 4.21 A 2-D Right memory, fisher reaction term and convection source term

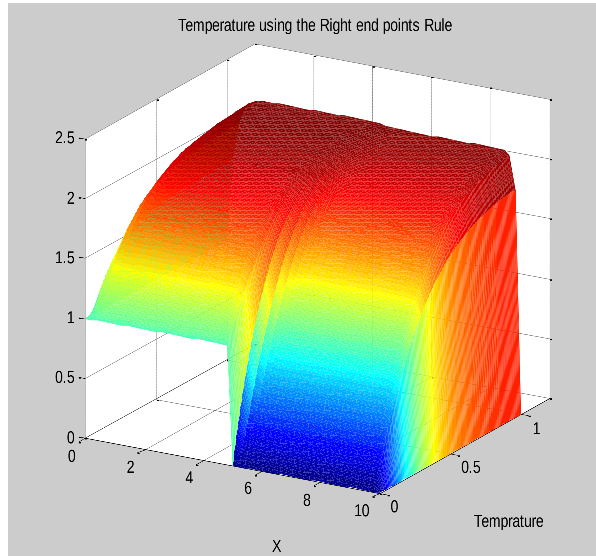


Fig 4.22 A 3-D Right memory, fisher reaction term and convection source term

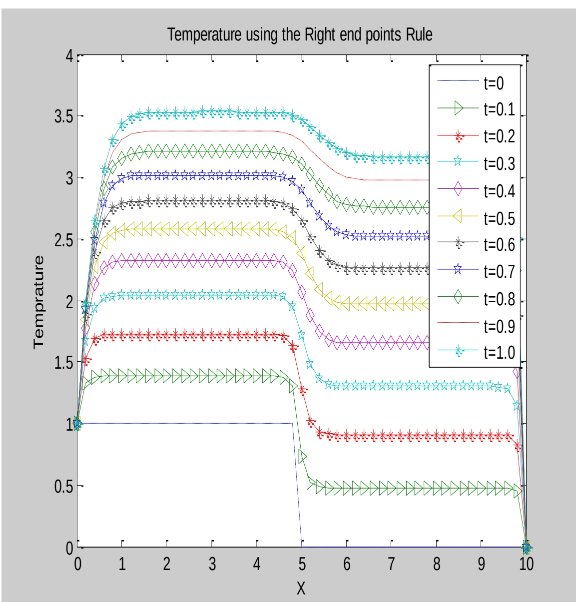


Fig 4.23 A 2-D Right memory, fisher reaction term and convection source term

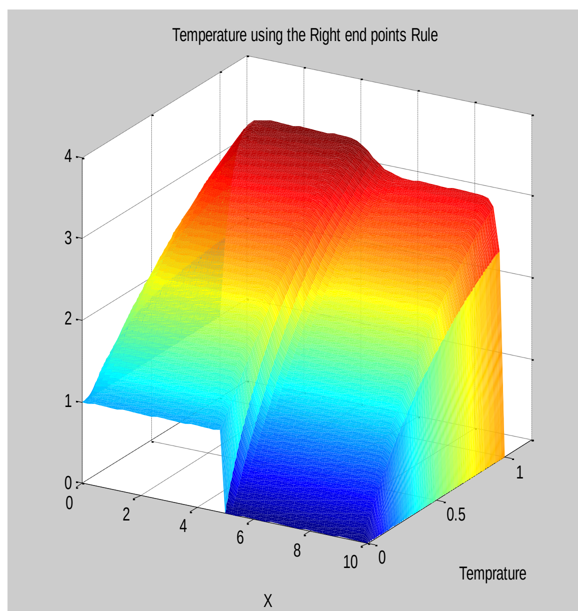


Fig 4.24 A 2-D Right memory, fisher reaction term and convection source term

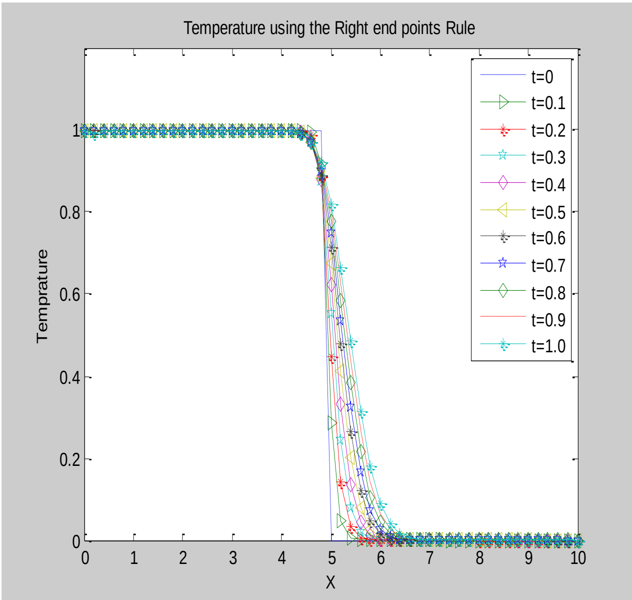


Fig 4.25 A 2-D Right memory term only

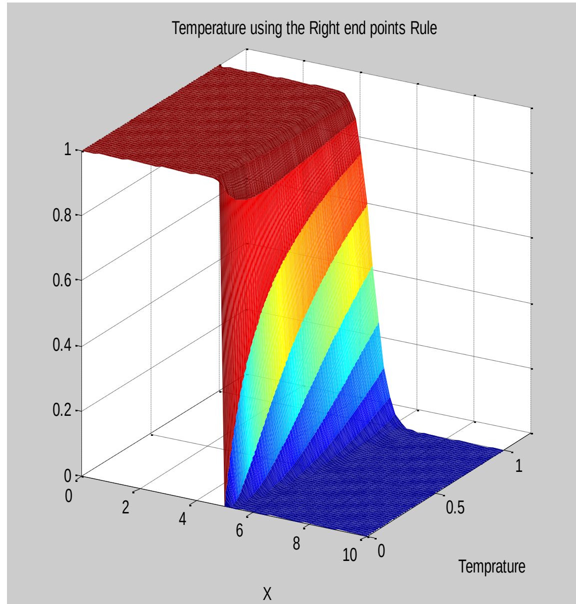


Fig 4.26 A 3-D Right memory term only

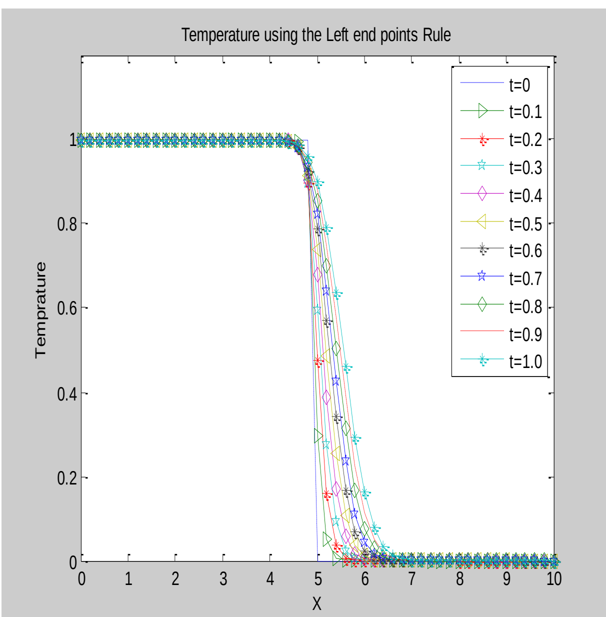


Fig 4.27 A 2-D Left convection source term without memory term

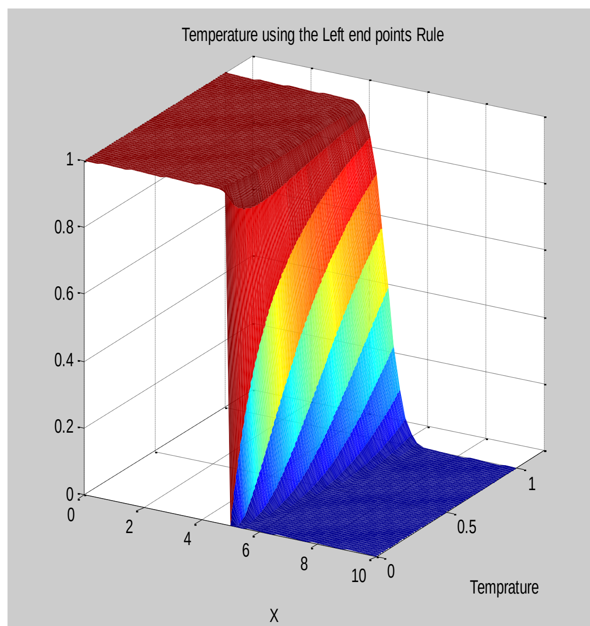


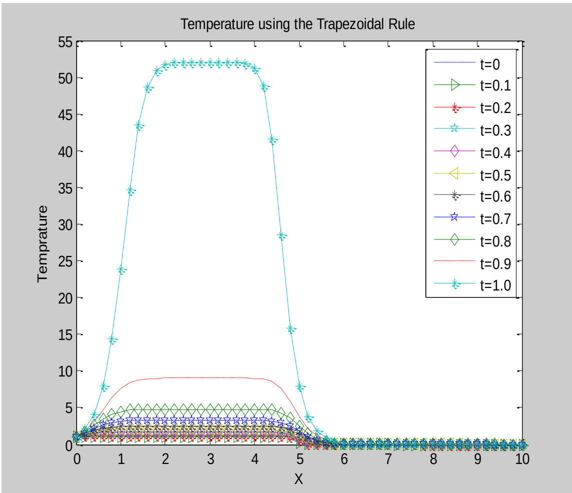
Fig 4.28 A 3-D Left convection source term without memory term

Analysis of the right end point rule

When we add memory term, convection source term, linear reaction term, quadratic reaction term and fisher reaction term, the diagram tends to the same shape for all, but when we add the non-linear (quadratic) term there is a big change in 2 dimensional and 3 dimensional diagram. Therefore, the fluid flow is more disturbed (affected) when we add the quadratic (non-linear) term than the others. The physical meaning of adding non-linear term could be applying a new force to the fluid flow through the channel or adding the source term. But when we add the memory term the fluid flow is more stable than the others and final it approaches to steady state (no more changes as we increase time). When we add the linear reaction term the fluid flow is flowing from top to bottom, which means we apply a less force during a linear reaction term.

Generally when we add convection source term, linear reaction term, quadratic reaction term and fisher reaction term to the right end point rule, the output give rise to the same diagram both in 2 dimensions and 3 dimensions although we couldn’t deny the change in shape of the diagram.

4.4 Graphical Analysis for Trapezoidal end point rule (Both Explicit and Implicit method)



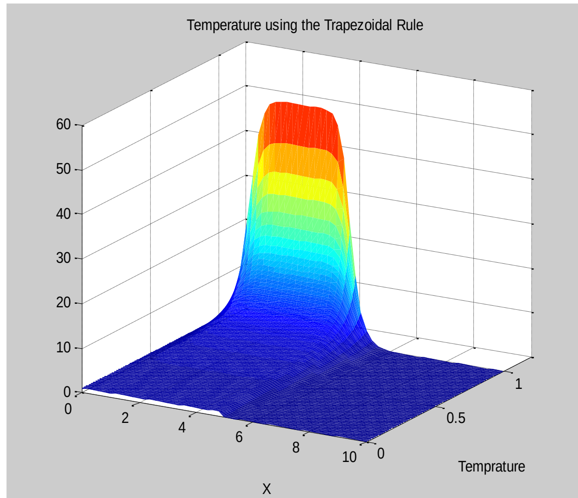
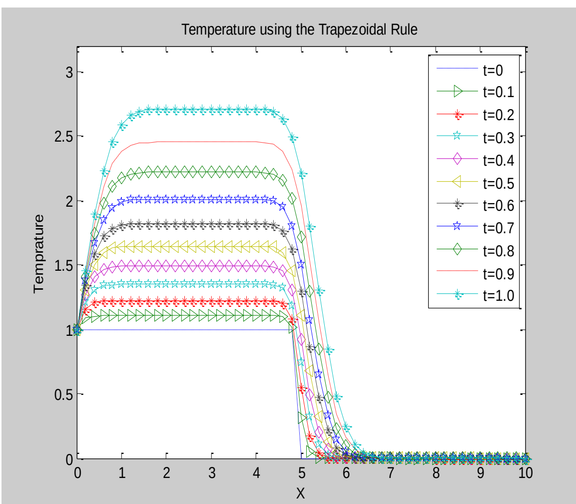


Fig 4.29 A 2-D Trapezoidal Memory and Quadratic linear

Fig 4.30 A 3-D Trapezoidal Memory and Quadratic linear



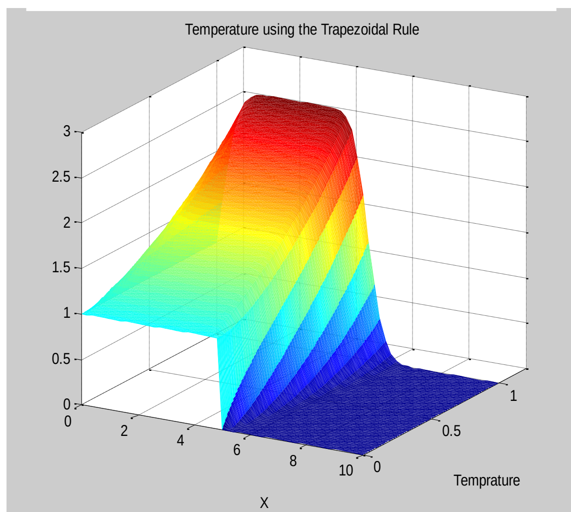


Fig4.31 A 2-D Trapezoidal Memory and linear

Fig 4.32 A 3-D Trapezoidal Memory and linear

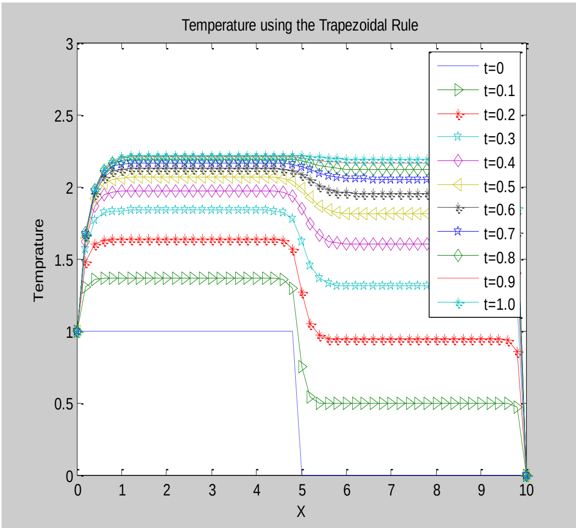


Fig 4.33 A 2-D Trapezoidal Memory, Fisher reaction and convection source

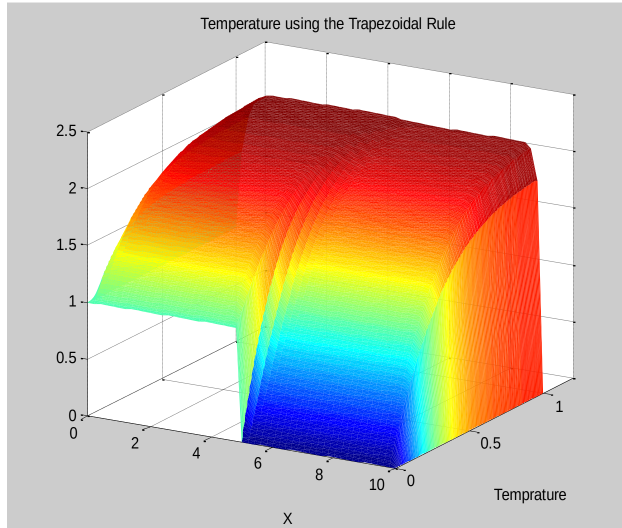


Fig 4.34 A 2-D Trapezoidal Memory, Fisher reaction and convection source term

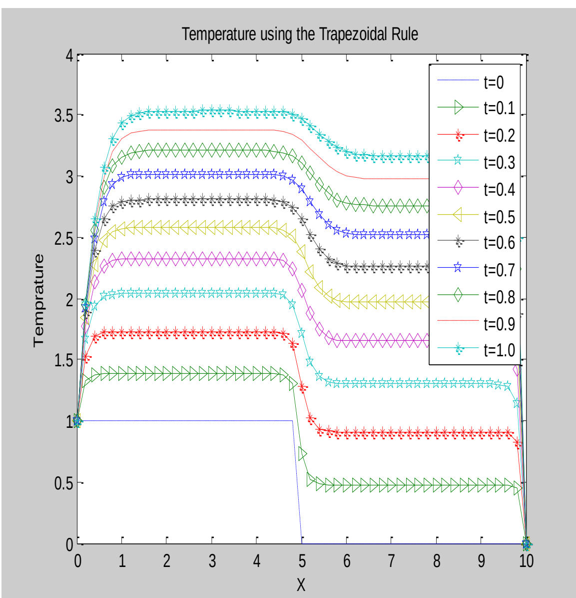


Fig 4.35 A 2-D Trapezoidal Memory and convection source term

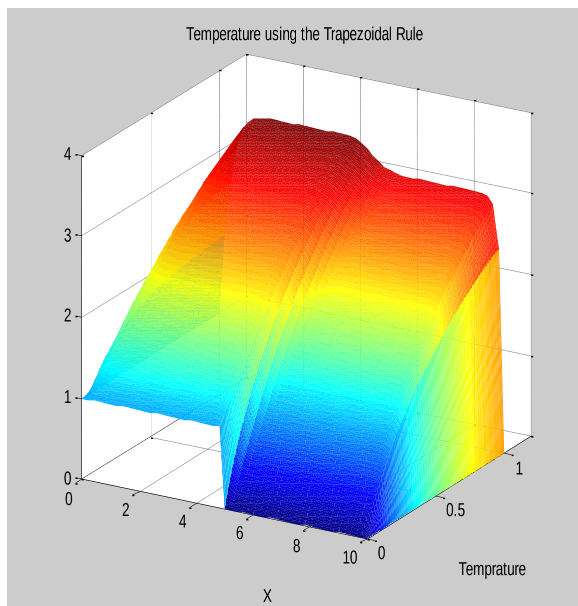


Fig 4.36 A 3-D Trapezoidal Memory and convection source term

Analysis of the trapezoidal end point rule

When we add memory term, convection source term, linear reaction term, quadratic reaction term and fisher reaction term, the diagram tends to the same shape for all, but when we add the non-linear (quadratic) term there is a big change in 2 dimensional and 3 dimensional diagram. Therefore, the fluid flow is more disturbed (affected) when we add the quadratic (non-linear) term than the others. The physical meaning of adding non-linear term could be applying a new force to the fluid flow through the channel or adding the source term. But when we add the memory term the fluid flow is more stable than the others and final it approaches to steady state (no more changes as we increase time). When we add the linear reaction term the fluid flow is flowing from top to bottom, which means we apply a less force during a linear reaction term.

Generally when we add convection source term, linear reaction term, quadratic reaction term and fisher reaction term to the right end point rule, the output give rise to the same diagram both in 2 dimensions and 3 dimensions although we couldn't deny the change in shape of the diagram.

A further study must be conducted to conclude that. Comparing them to an analytical solution will be one of the methods. For this specific problem the analytical solution is not provided, so it is difficult to prove the claim. But the paper can be used as an entry point for upcoming studies related to this topic.

APENDEX C: A COMPUTER PROGRAM WHICH SOLVES A LEFT END POINT RULE, RIGHT END POINT RULE AND TRAPEZOIDAL END POINT RULE

A computer program which solves a left end point rule script (Explicit Method)

```
%==========================================================================
% A Left end points Rule Script to solve the example
% ut=u(1-u)-.5u_x+.1u_xx+10integral(exp(-(t-s)/.01)u_xx(x,s))ds over
% 0 to t, where 0<=x<=10; 0<=t<=1;
% Intial condition: u(x,0)=1, 0<=x<=5; u(x,0)=0, 5<=x<=10
% Boundary Condition: u(0,t)=1, u(10, t)=0
% M=Number of subintervals along x axis
% N=Number of subintervals along t axis
% Within the explicit Method
% dx=.2, dt=.005

%
%
```

$$u_t = u(1-u) + \lambda u_x + \mu u_{xx} + \frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds + (Tw-u)$$

```
%--------------------------------------------------------------------------
% Prepared by: Habtamu Bekele Nora
% Submitted to: PROF. OKAY OSELOKA ONYEJEKWE
% Date: oct 20, 2015
%--------------------------------------------------------------------------

clear
close all
xf=10; % Final Space
T=1; % Final Time
M=50; % Number of subinterval for Space
N=200; % Number of subinterval for Time
% Declaring the Constants
lamda=-0.5;
micro=0.1;
tao=0.01;
D=0.1;
dx=xf/M; % Change of Space
dt=T/N; % Change of Time
x=linspace(0, 10, M+1); % Discretized Space
t=linspace(0, 1, N+1); % Discretized Time
% Defining these expressions in the discretized equation
r1=(micro*dt)/(dx^2);
r2=(lamda*dt)/(2*dx);
% Initial Condition
lb=5; % Initial of the Interval
nb=round(lb/dx);
u(1:nb, 1)=1;
u(nb+1:M+1, 1)=0;
% Dirichlet Boundary Conditions
u(1, 1:N+1)=1;
u(M+1, 1:N+1)=0;
for k=2:N+1 % Time Loop
```

```
% Defining these expressions in the discretized equation
r3=((D*dt^2)/(2*tao*dx^2))*exp(-(t(N+1)-t(k-1))/tao);
% Computing the value using direct substitution
u(2:M,k)=(r1-r2+r3)*u(1:M-1, k-1)+(1-2*r1-2*r3)*u(2:M,k-
1)+(r1+r2+r3)*u(3:M+1, k-1)+dt*u(2:M,k-1)-dt*u(2:M,k-1).^2;
end
% 2-D Graph for different time
figure(1)
plot(x, u(:,1), '-',x, u(:,21), '-',x, u(:,41), '-',x, u(:,61), '-',x,
u(:,81), '-',x, u(:,101), '-',x, u(:,121), '-',x, u(:,141), '-',x,
u(:,161),'-',x, u(:,181), '-',x, u(:,201), '-')
legend('t=0', 't=0.1', 't=0.2','t=0.3', 't=0.4', 't=0.5','t=0.6',
't=0.7','t=0.8', 't=0.9', 't=1.0')
title('Temperature using the Left end points Rule')
xlabel('X')
ylabel('Temprature')
axis([0 10 0 1.2])
% 3-D graph of the Approximated solution by the Implicit Method
figure(2)
set(gcf,'renderer','painters')
surf(x, t, u');
shading interp;
title('Temperature using the Left end points Rule')
xlabel('X')
ylabel('Temprature')
axis([0 10 0 1.2])
view([30 20])
==============================================================================
```

A computer program which solves a right end point rule script (Implicit Method)

```
%=============================================================================
% A Right end points Rule Script to solve the example
% ut=u(1-u)-.5u_x+.1u_xx+10integral(exp(-(t-s)/.01)u_xx(x,s))ds over
% 0 to t, where 0<=x<=10; 0<=t<=1;
% Initial condition: u(x,0)=1, 0<=x<=5; u(x,0)=0, 5<=x<=10
% Boundary Condition: u(0,t)=1, u(10, t)=0
% M=Number of subintervals along x axis
% N=Number of subintervals along t axis
% Within the explicit Method
% dx=.2, dt=.005
%----------------------------------------------------------------------------
% The Governing equation is:
```

$$u_t = u(1-u) + \lambda u_x + \mu u_{xx} + \frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds$$

```
% Prepared by: Habtamu Bekele Nora
% Submitted to: PROF. OKAY OSELOKA ONYEJEKWE
% Date: Oct 20, 2015
%-------------------------------------------------------------------------
```

```
clear
close all
xf=10; % Final Space
T=1; % Final Time
M=50; % Number of subinterval for Space
N=200; % Number of subinterval for Time
% Declaring the Constants
lamda=-0.5;
micro=0.1;
tao=0.01;
D=0.1;
dx=xf/M; % Change of Space
dt=T/N; % Change of Time
x=linspace(0, 10, M+1); % Discretized Space
t=linspace(0, 1, N+1); % Discretized Time
% Defining these expressions in the discretized equation
r1=(micro*dt)/(dx^2);
r2=(lamda*dt)/(2*dx);
% Initial Condition
lb=5; % Initial of the Interval
nb=round(lb/dx);
u(1:nb, 1)=1;
u(nb+1:M+1, 1)=0;
% Dirichlet Boundary Conditions
u(1, 1:N+1)=1;
u(M+1, 1:N+1)=0;
for k=2:N+1 % Time Loop


% defining these expressions in the discretized equation
r=((D*dt^2)/(2*tao*dx^2))*exp(-(t(N+1)-t(k))/tao);
```

```
% setting the diagonal parameters value
a(1:M-1)=-r;
b(1:M-1)=1+2*r-dt+dt*(2*(u(2:M,k-1)));
c(1:M-1)=-r;
d(1:M-1)=(r1-r2)*u(1:M-1, k-1)+(1-2*r1)*u(2:M,k-1)+(r1+r2)*u(3:M+1, k-
1)+dt*u(2:M,k-1)+dt*u(2:M,k-1).^2;
% Solving the values of u by the tri-diagonal algorithm
u(2:M,k)=tridiagonal(M-1,a, b, c, d);
end
% 2-D Graph for different time
figure(1)
plot(x, u(:,1), '-',x, u(:,21), '-',x, u(:,41), '-',x, u(:,61), '-',x,
u(:,81), '-',x, u(:,101), '-',x, u(:,121), '-',x, u(:,141), '-',x,
u(:,161),'-',x, u(:,181), '-',x, u(:,201), '-')
legend('t=0', 't=0.1', 't=0.2','t=0.3', 't=0.4', 't=0.5','t=0.6',
't=0.7','t=0.8', 't=0.9', 't=1.0')
title('Temperature using the Right end points Rule')
xlabel('X')
ylabel('Temprature')
axis([0 10 0 1.2])
% 3-D graph of the Approximated solution by the Implicit Method
figure(2)
set(gcf,'renderer','painters')
surf(x, t, u');
shading interp;
title('Temperature using the Right end points Rule')
xlabel('X')
ylabel('Temprature')
axis([0 10 0 1.2])
view([30 20])
```

A computer program which solves a right end point rule with convection source term

```
%==========================================================================
% A Right end points Rule Script to solve the example
% ut=u(1-u)-.5u_x+.1u_xx+10integral(exp(-(t-s)/.01)u_xx(x,s))ds over
% 0 to t, where 0<=x<=10; 0<=t<=1;
% Initial condition: u(x,0)=1, 0<=x<=5; u(x,0)=0, 5<=x<=10
% Boundary Condition: u(0,t)=1, u(10, t)=0
% M=Number of subintervals along x axis
% N=Number of subintervals along t axis
% Within the explicit Method
% dx=.2, dt=.005
%--------------------------------------------------------------------------
% The Governing equation is:
```

% $$u_t = u(1-u) + \lambda u_x + \mu u_{xx} + \frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds + (Tw - u)$$

```
% Prepared by: Habtamu Bekele Nora
% Submitted to: PROF. OKAY OSELOKA ONYEJEKWE
% Date: Oct 20, 2015
%---------------------------------------------------------------------------

clear
close all
xf=10; % Final Space
T=1; % Final Time
M=50; % Number of subinterval for Space
N=200; % Number of subinterval for Time
% Declaring the Constants
lamda=-0.5;
micro=0.1;
tao=0.01;
D=0.1;
dx=xf/M; % Change of Space
dt=T/N; % Change of Time
x=linspace(0, 10, M+1); % Discretized Space
t=linspace(0, 1, N+1); % Discretized Time
% Defining these expressions in the discretized equation
r1=(micro*dt)/(dx^2);
r2=(lamda*dt)/(2*dx);
% Initial Condition
lb=5; % Initial of the Interval
nb=round(lb/dx);
u(1:nb, 1)=1;
u(nb+1:M+1, 1)=0;
Tw=5;
% Dirichlet Boundary Conditions
u(1, 1:N+1)=1;
u(M+1, 1:N+1)=0;
for k=2:N+1 % Time Loop
```

```
% Defining these expressions in the discretized equation
r=((D*dt^2)/(2*tao*dx^2))*exp(-(t(N+1)-t(k))/tao);
% Setting the diagonal parameters value
a(1:M-1)=-r;
b(1:M-1)=1+2*r;
c(1:M-1)=-r;
d(1:M-1)=(r1-r2)*u(1:M-1, k-1)+(1-2*r1)*u(2:M,k-1)+(r1+r2)*u(3:M+1, k-
1)+dt*(Tw-u(2:M,k-1));
% Solving the values of u by the tri-diagonal algorithm
u(2:M,k)=tridiagonal(M-1,a, b, c, d);
end
% 2-D Graph for different time
figure(1)
plot(x, u(:,1), '--',x, u(:,21), '>-',x, u(:,41), '*-',x, u(:,61), 'p-',x,
u(:,81), 'd-',x, u(:,101), '<-',x, u(:,121), '*-',x, u(:,141), 'p-',x,
u(:,161),'d-',x, u(:,181), '--',x, u(:,201), '*-')
legend('t=0', 't=0.1', 't=0.2','t=0.3', 't=0.4', 't=0.5','t=0.6',
't=0.7','t=0.8', 't=0.9', 't=1.0')
title('Temperature using the Right end points Rule')
xlabel('X')
ylabel('Temprature')
axis([0 10 0 4])
% 3-D graph of the Approximated solution by the Implicit Method
figure(2)
set(gcf,'renderer','painters')
surf(x, t, u');
shading interp;
title('Temperature using the Right end points Rule')
xlabel('X')
ylabel('Temprature')
axis([0 10 0 1.2])
view([30 20])
============================================================================
```

A computer program which solves a trapezoidal end point rule script (Both Implicit and explicit Method)

```
%========================================================================
% A Trapezoidal Rule Script to solve the example
% ut=u(1-u)-.5u_x+.1u_xx+10integral(exp(-(t-s)/.01)u_xx(x,s))ds over
% 0 to t, where 0<=x<=10; 0<=t<=1;
% Intial condition: u(x,0)=1, 0<=x<=5; u(x,0)=0, 5<=x<=10
% Boundary Condition: u(0,t)=1, u(10, t)=0
% M=Number of subintervals along x axis
% N=Number of subintervals along t axis
% Within the Implicit Method
% dx=.2, dt=.005
```

%
$$u_t = u(1-u) + \lambda u_x + \mu u_{xx} + \frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds + (u_w - u)$$

```
%========================================================================
% Prepared by: Habtamu Bekele Nora
% Submitted to: PROF. OKAY OSELOKA ONYEJEKWE
% Date: June 18, 2015 %
%------------------------------------------------------------------------
```

```
clear
close all
xf=10; % Final Space
T=1; % Final Time
M=50; % Number of subinterval for Space
N=200; % Number of subinterval for Time
% Declaring the Constants
lamda=-0.5;
micro=0.1;
tao=0.01;
D=0.1;
dx=xf/M; % Change of Space
dt=T/N; % Change of Time
x=linspace(0, 10, M+1); % Discretized Space
t=linspace(0, 1, N+1); % Discretized Time
% Defining these expressions in the discretized equation
r1=(micro*dt)/(dx^2);
r2=(lamda*dt)/(2*dx);
% Initial Condition
lb=5; % Initial of the Interval
nb=round(lb/dx);
u(1:nb, 1)=1;
u(nb+1:M+1, 1)=0;
% Dirichlet Boundary Conditions
u(1, 1:N+1)=1;
u(M+1, 1:N+1)=0;
for k=2:N+1 % Time Loop
```

```
% Defining these expressions in the discretized equation
r3=((D*dt^2)/(2*tao*dx^2))*exp(-(t(N+1)-t(k-1))/tao);
r=((D*dt^2)/(2*tao*dx^2))*exp(-(t(N+1)-t(k))/tao);
% setting the diagonal parameters value
a(1:M-1)=-r;
b(1:M-1)=1+2*r-dt+dt*(2*(u(2:M,k-1)));
c(1:M-1)=-r;
d(1:M-1)=(r1-r2+r3)*u(1:M-1, k-1)+(1-2*r1-2*r3)*u(2:M,k-
1)+(r1+r2+r3)*u(3:M+1, k-1)+dt*u(2:M,k-1)-dt*u(2:M,k-1).^2+dt*(u(2:M,k-1)).^2
;
% solving the values of u by the tridiagonal algorithm
u(2:M,k)=tridiagonal(M-1,a, b, c, d);
end
% 2-D Graph for different time
figure(1)
plot(x, u(:,1), '-',x, u(:,21), '-',x, u(:,41), '-',x, u(:,61), '-',x,
u(:,81), '-',x, u(:,101), '-',x, u(:,121), '-',x, u(:,141), '-',x,
u(:,161),'-',x, u(:,181), '-',x, u(:,201), '-')
legend('t=0', 't=0.1', 't=0.2','t=0.3', 't=0.4', 't=0.5','t=0.6',
't=0.7','t=0.8', 't=0.9','t=1.0')
title('Temperature using the Trapezoidal Rule')
xlabel('X')
ylabel('Temprature')
axis([0 10 0 1.2])
% 3-D graph of the Approximated solution by the Implicit Method
figure(2)
set(gcf,'renderer','painters')
surf(x, t, u');
shading interp;
title('Temperature using the Trapezoidal Rule')
xlabel('X')
ylabel('Temprature')
axis([0 10 0 1.2])
view([30 20])
```

A computer program which solves a trapezoidal end point rule script (Both Implicit and explicit Method)

```
%==========================================================================
% A Trapezoidal Rule Script to solve the example
% ut=u(1-u)-.5u_x+.1u_xx+10integral(exp(-(t-s)/.01)u_xx(x,s))ds over
% 0 to t, where 0<=x<=10; 0<=t<=1;
% Intial condition: u(x,0)=1, 0<=x<=5; u(x,0)=0, 5<=x<=10
% Boundary Condition: u(0,t)=1, u(10, t)=0
% M=Number of subintervals along x axis
% N=Number of subintervals along t axis
% Within the Implicit Method
% dx=.2, dt=.005
%
```

$$u_t = u(1-u) + \lambda u_x + \mu u_{xx} + \frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds + (u_w - u)$$

```
%==========================================================================
% Prepared by: Habtamu Bekele
% Submitted to: PROF. OKAY OSELOKA ONYEJEKWE
% Date: June 18, 2015 %
%--------------------------------------------------------------------------
```

```
clear
close all
xf=10; % Final Space
T=1; % Final Time
M=50; % Number of subinterval for Space
N=200; % Number of subinterval for Time
% Declaring the Constants
lamda=-0.5;
micro=0.1;
tao=0.01;
D=0.1;
dx=xf/M; % Change of Space
dt=T/N; % Change of Time
x=linspace(0, 10, M+1); % Discretized Space
t=linspace(0, 1, N+1); % Discretized Time
% Defining these expressions in the discretized equation
r1=(micro*dt)/(dx^2);
r2=(lamda*dt)/(2*dx);
% Initial Condition
lb=5; % Initial of the Interval
nb=round(lb/dx);
u(1:nb, 1)=1;
u(nb+1:M+1, 1)=0;
% Dirichlet Boundary Conditions
u(1, 1:N+1)=1;
u(M+1, 1:N+1)=0;
for k=2:N+1 % Time Loop
% Defining these expressions in the discretized equation
r3=((D*dt^2)/(2*tao*dx^2))*exp(-(t(N+1)-t(k-1))/tao);
r=((D*dt^2)/(2*tao*dx^2))*exp(-(t(N+1)-t(k))/tao);
```

```
% Setting the diagonal parameters value
a(1:M-1)=-r;
b(1:M-1)=1+2*r-dt+dt*(2*(u(2:M,k-1)));
c(1:M-1)=-r;
d(1:M-1)=(r1-r2+r3)*u(1:M-1, k-1)+(1-2*r1-2*r3)*u(2:M,k-
1)+(r1+r2+r3)*u(3:M+1, k-1)+dt*u(2:M,k-1).^2;
% Solving the values of u by the tridiagonal algorithm
u(2:M,k)=tridiagonal(M-1,a, b, c, d);
end
% 2-D Graph for different time
figure(1)
plot(x, u(:,1), '--',x, u(:,21), '>-',x, u(:,41), '*-',x, u(:,61), 'p-',x,
u(:,81), 'd-',x, u(:,101), '<-',x, u(:,121), '*-',x, u(:,141), 'p-',x,
u(:,161),'d-',x, u(:,181), '--',x, u(:,201), '*-')
legend('t=0', 't=0.1', 't=0.2','t=0.3', 't=0.4', 't=0.5','t=0.6',
't=0.7','t=0.8', 't=0.9','t=1.0')
title('Temperature using the Trapezoidal Rule')
xlabel('X')
ylabel('Temprature')
axis([0 10 0 55])
% 3-D graph of the Approximated solution by the Implicit Method
figure(2)
set(gcf,'renderer','painters')
surf(x, t, u');
shading interp;
title('Temperature using the Trapezoidal Rule')
xlabel('X')
ylabel('Temprature')
axis([0 10 0 1.2])
view([30 20])
```

==

A computer program which solves a Simpson's end point rule script

```
% A Simpson's Rule Script to solve the example
% ut=u(1-u)-.5u_x+.1u_xx+10integral(exp(-(t-s)/.01)u_xx(x,s))ds over %
% 0 to t, where 0<=x<=10; 0<=t<=1; %
% Intial condition: u(x,0)=1, 0<=x<=5; u(x,0)=0, 5<=x<=10 %
% Boundary Condition: u(0,t)=1, u(10, t)=0 %
% M=Number of subintervals along x axis %
% N=Number of subintervals along t axis %
% within the Implicit Method %
% dx=.2, dt=.005
```

% $$u_t = u(1-u) + \lambda u_x + \mu u_{xx} + \frac{D}{\tau}\int_0^t e^{-\frac{t-s}{\tau}} u_{xx}(x,s)ds + \Delta t(u_w - u_i^k)$$

```
%##########################################################################%
% Prepared by: Habtamu Bekele
%-----------------------------------------------------------------------
% Submitted to: PROF. OKAY OSELOKA ONYEJEKWE %
% Date: June 18, 2015 %
%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%
%========================================================================
```

```
clear
close all
xf=10; % Final Space
T=1; % Final Time
M=50; % Number of subinterval for Space
N=200; % Number of subinterval for Time
% Declaring the Constants
lamda=-0.5;
micro=0.1;
tao=0.01;
D=0.1;
dx=xf/M; % Change of Space
dt=T/N; % Change of Time
x=linspace(0, 10, M+1); % Discretized Space
t=linspace(0, 1, N+1); % Discretized Time
% Defining these expressions in the discretized equation
r3=(micro*dt)/(dx^2);
r2=(lamda*dt)/(2*dx);
% Initial Condition
lb=5; % Initial of the Interval
nb=round(lb/dx);
u(1:nb, 1)=1;
u(nb+1:M+1, 1)=0;
% Dirichlet Boundary Conditions
u(1, 1:N+1)=1;
u(M+1, 1:N+1)=0;
% Euler method to compute the value next to the initial condition
u(1:M+1, 2)=u(1:M+1, 1)+dt*u(1:M+1, 1);
for k=2:N % Time Loop
```

```
% Defining these expressions in the discretized equation
r=((D*dt^2)/(3*tao*dx^2))*exp(-(t(N+1)-t(k+1))/tao);
r1=((D*dt^2)/(3*tao*dx^2))*exp(-(t(N+1)-t(k-1))/tao);
r4=((4*D*dt^2)/(3*tao*dx^2))*exp(-(t(N+1)-t(k))/tao);
% Setting the diagonal parameters value
a(1:M-1)=-r;
b(1:M-1)=1+2*r;
c(1:M-1)=-r;
d(1:M-1)=(r3-r2+r4)*u(1:M-1, k)+(1-2*r3-2*r4)*u(2:M,k)+(r2+r3+r4)*u(3:M+1,
k)+dt*u(2:M,k)-dt*u(2:M,k).^2+r1*u(1:M-1, k-1)-2*r1*u(2:M, k-1)+r1*u(3:M+1,
k-1);
% Solving the values of u by the tridiagonal algorithm
u(2:M,k+1)=tridiagonal(M-1,a, b, c, d);
end
% 2-D Graph for different time
figure(1)
plot(x, u(:,1), '-',x, u(:,21), '-',x, u(:,41), '-',x, u(:,61), '-',x,
u(:,81), '-',x, u(:,101), '-',x, u(:,121), '-',x, u(:,141), '-',x,
u(:,161),'-',x, u(:,181), '-',x, u(:,201), '-')
legend('t=0', 't=0.1', 't=0.2','t=0.3', 't=0.4', 't=0.5','t=0.6',
't=0.7','t=0.8', 't=0.9', 't=1.0')
title('Temperature using the Simpsons Rule')
xlabel('X')
ylabel('Temprature')
axis([0 10 0 1.2])
% 3-D graph of the Approximated solution by the Implicit Method
figure(2)
set(gcf,'renderer','painters')
surf(x, t, u');
shading interp;
title('Temperature using the Simpsons Rule')
xlabel('X')
ylabel('Temprature')
axis([0 10 0 1.2])
view([30 20])
==============================================================================
```