An-Najah National University

Faculty of Graduate Studies

Study of Higher Order Numerical Methods for Solving Parabolic Partial Differential Equations with Applications

By

Ahmad M.O Kassef

Supervisor

Dr. Samir Matar

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

2019

Study of Higher Order Numerical Methods for Solving Parabolic Partial Differential Equations with Applications

By

Ahmad M.O Kassef

This thesis was defended successfully on 24/06/2019 and approved by:

Defense Committee Members	**Signatures**
– **Dr. Samir Matar /Supervisor**	…………………… S. Matar
– **Dr. Abdelhalim Ziqan / External Examiner**	…………………… Halimziqan
– **Dr. Mohamad Najib Assa'd / Internal Examiner**	……………………

Dedication

I dedicate this thesis to my mother, father, my brothers and my family. Without their support, love, patience and understanding, nothing will be complete.

Thanks all.

Acknowledgment

First and before all, thanks and praises to Allah for blessing me much more I deserve.

Thanks to my father who have always loved me unconditionally and who good examples have taught me to work hard for the things that I aspire to achieve.

Thanks to my supervisor Dr. Samir Matar for his understanding and support, I really appreciate his advices and assistance. Finally, much of love and thanks for my School family for their support and love.

الإقرار

أنا الموقع أدناه مقدم الرسالة التي تحمل عنوان :

Study of Higher Order Numerical Methods for Solving Parabolic Partial Differential Equations with Applications

أقر بأن ما اشتملت عليه هذه الرسالة إنما هي نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وأن هذه الرسالة ككل، أو أي جزء منها لم يقدم من قبل لنيل أية درجة علمية أو بحث علمي أو بحثي لدى أية مؤسسة تعليمية أو بحثية أخرى.

Declaration

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

Student's Name: **اسم الطالب:**

Signature: **التوقيع:**

Date: **التاريخ:**

Table of contents

List of Figures

List of Tables

High order numerical methods for Solving Parabolic Partial Differential Equations with Application

By
Ahmad kassef
Supervisor
Dr. Samir Matar

Abstract

Parabolic Partial Differential equations have clearly emerged in many branches of science, for example technology, engineering, physics and many others. So based on the importance of parabolic equations, new efficient and more accurate numerical methods were discussed, studied and analyzed.

These numerical methods are Explicit Method, Implicit Method, Crank-Nicolson Method, Finite Difference Method, Method of Line, And Pade Approximation.

Each method has been studied and implemented with examples and A MATLAB code was written for each method to obtain very accurate results. The Numerical results were compared to determine the best method which is the most accurate.

Chapter One

Introduction

1.1. Introduction

Mathematics in general is the body of knowledge centered on such concepts as quantity, structure, space, and change, and also the academic discipline that studies them. Benjamin Pierce called it " the science that draws necessary conclusions”. On the other hand, Mathematics is the basis of all science, no science can do itself without the existence of mathematics; it is the language of communication in the world that any specialist can understand.

Applied mathematics, the application of mathematics to such fields, inspires and makes use of new mathematical discoveries and sometimes leads to the development of entirely new disciplines. [13]

1.2. Partial Differential Equations

A mathematical formulation of many important scientific and engineering problems involving rates of change with respect to two or more independent variables can often be expressed as partial differential equations(PDEs) [14].

The majority of the problems of physics, science and engineering fall naturally into one of the following three physical categories: equilibrium problems, eigenvalue problems and propagation problems[18]. Also, most physical phenomena, whether in the field of fluid flow or the spread of heat, can be generally described by partial differential equations. Often, the problems that one would like to solve exceed the capacity of even the most powerful computers. On the other hand, the time required is too large to all inclusion of advanced mathematical models in the design process.

To solve partial differential equations there are many methods that are practically applied. One of the methods are to transform the partial differential equation into an ordinary differential equation. The partial differential equations can then be solved in two ways; first by analytical methods (such as the method of separating variables, which allows the partial differential equation to be converted into a normal differential equation), and second by numerical methods that convert the partial differential equation to a set of difference equations or system of equations, which can then be solved using computer programs.

In general, there are many numerical methods for finding solutions to these types of equations, both numerically or analytically. But, in general, each method works well only for a specific class of problems, often solving some examples. But generally existence of solution for PDE is a hard problem and one of open problem in mathematics.

1.3. Why Numerical Methods?

Numerical methods are used in the modern world in order to give an approximation of the solution by using computers in a fraction of a second's time. For example, to locate a moving hostile object you need to locate its position within microseconds. (Elapsed time is 0.0015701 seconds for example). If there is a heat flow, to know the progress of the heat within a short time is important. [6][7][8][9] [10].

1.4. Order of Convergence

The order of convergence is one of the primary ways to estimate the actual rate of convergence, the speed at which the errors go to zero. Typically, the order of convergence measures the asymptotic behavior of convergence. For example, Newton's method is said to have quadratic convergence, so the method has order 2. However, the true rate of convergence depends on the problem, the initial value taken, etc., and is typically impossible to quantify exactly. Numerical methods have different orders of convergence. Developing high order convergence is needed.

Consider a case of a sequence of vectors, $\{y_i\}_{i=0}^{\infty}$, $y_i \in R^n$ is said to converge to a vector L with order *p* if there exists a constant $0<K\leq 1$ such that

$$\|y_{i+1} - L\| \leq K \|y_i - L\|^p$$

The above inequality is some type of a contraction mapping in a metric space.

Knowledge of calculus, partial differential equations, numerical methods for solving partial and ordinary differential equations is of at most importance. [1]

Chapter Two

Parabolic Partial Differential Equation

2.1. Introduction

The solutions of PDEs are important in many fields of science and engineering, because they describe the behavior of electric, gravitational and fluid potential. Most of the PDEs that arise in mathematical models of physical phenomena are difficult (if not impossible) to solve analytically, so we have used numerical methods to approximate the solution.

The general two dimensional second order linear (PDE) for the function U(x, y) can be written as: [12]

$$A\frac{\partial^2 U}{\partial x^2}+B\frac{\partial^2 U}{\partial x\partial y}+C\frac{\partial^2 U}{\partial y^2}+D\frac{\partial U}{\partial x}+E\frac{\partial U}{\partial y}+FU=G\ldots \quad (1)$$

where A, B, C, D, E, F and G may be continuous functions of the independent variables x and y.

The solution the function U(x, y) on its domain.

Equation (1) can be classified as:

1) Hyperbolic if $B^2-4AC>0$

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

3) Elliptic if $B^2-4AC<0$

PDEs, which belong to one of the most important parts of mathematical analysis, are closely related to the physical world. We may come across the wave equation or the heat equation, and the names of Euler, Poisson, Laplace, etc., are quite familiar to scientists. One may encounter PDEs not only in physics, mechanics and engineering, but in other fields as well, such as biology, finance and in image analysis.

Many methods were used to solve PDEs. It replaces differential operators by finite differences and the PDE becomes a finite system of linear equations, or it can become a finite system of ordinary differential equations (ODEs). An Algorithm can be established then a computer implementation is used to solve the PDEs defined on regular or not regular geometries. [12]

2.2. A simple Experiment on Heat Diffusion

On the assumption that we have a simple issue that we will divide into three steps:

Step 1: Start with a metal rod (copper, for example) with certain and suitable dimensions (for example, length and diameter are known, m=L and diameter =2 cm) under a condition, side surface only (without the two bases) must be coated with an insulated material. That means that the leakage of heat will only be from the two bases, not from the side surface.

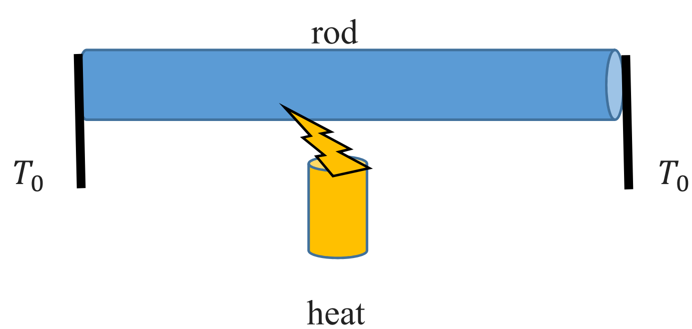


Figure2.1: Start of experiment

Step 2: Now place the rod into a constant temperature environment (T_0 Celsius) for enough to make the temperature of the rod as equal as the temperature of that environment(T_0 Celsius). Let's assume that the constant temperature is 10 Celsius.

Step 3: Now we will monitor the changes of heat in the rod in a certain period of time and we will observe the changes of the rod temperature. That is called the solution of parabolic partial differential equations. [13]

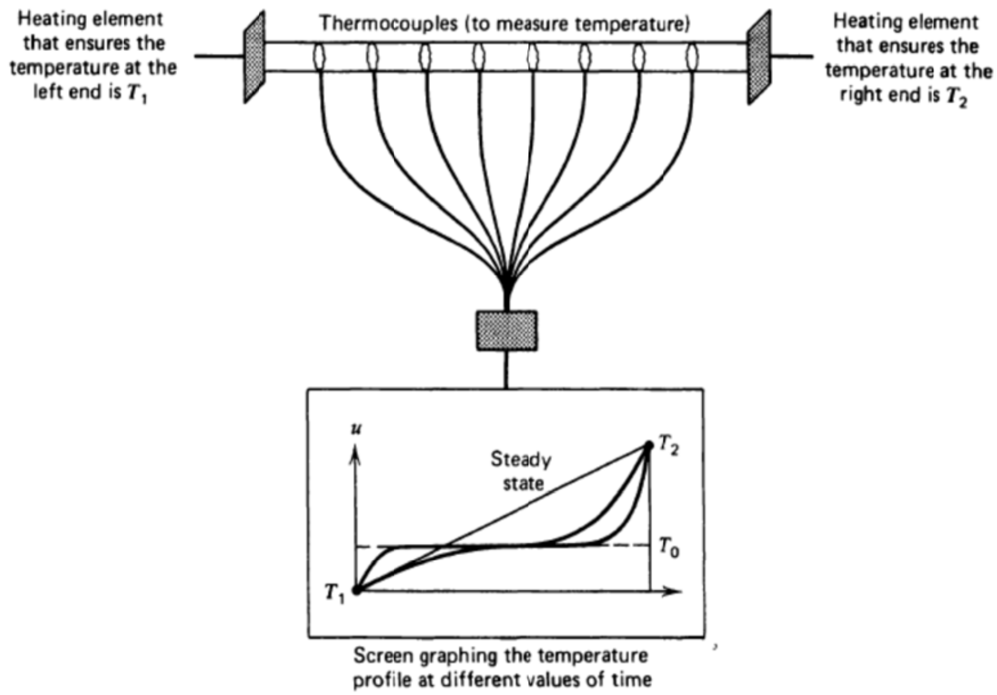


Figure2.2: Schematic diagram of the experiment.

2.3. A Model Problem

Many problems in physics, engineering and mathematics are modelled by some special cases of the linear or nonlinear parabolic equation for the unknown U (x, y).

The heat equation describes the evolution of temperature U (x, y):

$$\frac{\partial U}{\partial t} = K \frac{\partial^2 U}{\partial x^2} \qquad (2.3.1)$$

$u(0,t) = u(L,t) = 0$ *boundary conditions*

$u(x,0) = g(x)$ *initial condition* $\quad t > 0$ and $0 < x < L$

$f(x,t)$, g(x) , and L are given

Step 1. First introduce the points where we will compute the solution General solutions of the heat equation can be found by more than one method, but if we use the separation of variables. Some examples appear in the heat equation article. They are examples of Fourier series for periodic f and Fourier transforms for non-periodic f. Using the Fourier transform, a general solution of the heat equation has the form: [16]

$$u(t,x) = \sum_{n=1}^{\infty} B_n \, e^{-kt\left(\frac{n\pi}{L}\right)^2} \sin\left(x\frac{n\pi}{L}\right) \qquad (2.3.2)$$

where the arbitrary constants B_n, $n \geq 1$, are as yet undetermined. And to determine $B_n, n \geq 1$, we substitute t=0 in (2.3.2) and by using the initial condition we find

$$f(x) = \sum_{n=1}^{\infty} B_n \sin\left(x\frac{n\pi}{L}\right) \qquad (2.3.3)$$

The constants B_n can be determined in this case by using Fourier coefficients given by the formula

$$B_n = \left(\frac{2}{L}\right)\int_0^L f(x) \times \sin\left(\frac{n\pi}{L}x\right)dx \qquad (2.3.4)$$

Having determined the constants B_n, the particular solution $u(t,x)$ follows immediately.

On the other hand, if the initial condition $f(x)$ is given in terms of $\sin\left(\frac{n\pi}{L}x\right)$, $n \geq 1$, , the constants B_n can be completely determined by expanding (2.3.2), using the initial condition, and by equating the coefficients of like terms on both sides. The initial condition in the first two examples will be trigonometric functions.

To give a clear overview of the method of separation of variables, we have selected several examples to illustrate the analysis presented above.

The description of this physical situation requires three types of equations:

1. Partial differential equation describing the problem under study.
2. Boundary conditions: the physical situation of the problem state.
3. Initial conditions: that describe the physical phenomenon at the beginning of the movement.

2.4. Conditions of Parabolic Partial Differential Equation:

The parabolic partial equation contains two types of conditions:

1- Initial Condition (IC) when t=0: $u(x,0)=g(x)$.for $0 \leq x \leq L$.

2- Boundary Conditions(BC's): (have 3-types)

   i- Dirichlet Boundary Conditions [Value of solution at boundary] , for$0 < x < L$,$u(0,\mathrm{t}) = p(t)$, $u(L,\mathrm{t}) = q(\mathrm{t})$.

   ii- Neumen Boundary Conditions [Rate of change of the solution at the boundaries is given], $u_x(0,\mathrm{t}) = p(\mathrm{t})$, $u_x(L,\mathrm{t}) = q(\mathrm{t})$.

   iii- Robin Boundary Conditions [Mixed of both (i) and (ii)].

2.5. Some Examples of Parabolic PDEs

1) The Diffusion equation that gives the concentration of monomers C (r, t) during heating of a ceramic mold is given by:

$$\frac{\partial C}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left(r D \frac{\partial C}{\partial r}\right) + \overset{*}{Q} \qquad 0 < r < r_0 \ , \ t > 0$$

r: is the radius of the cylinder, D is the diffusion coefficient, and Q^* is the rate of production of monomer.[8]

2) Lateral leakage equation Heat:

$$\frac{\partial U}{\partial t} = \alpha^2 \frac{\partial^2 U}{\partial x^2} - \beta(u - u_0) \qquad , \beta > 0$$

The heat side leakage is directly proportional to the temperature difference as the equation above describes the heat flow in a rod as well as the leakage of heat from (or to) the sides of the rod. If $\beta > 0$ then the leakage from the inside to the outside, but If $\beta < 0$ then the leakage from the outside to the inside. Moreover, in both cases the proportionality is in direct proportion to the difference between the temperature of the rod U (x, *y)* and the temperature of the u_0 .[9][7]

3) The Convection–Diffusion equation

Assume that some pollution running into a flow with velocity V, the concentration U (x, y) is changing as a function of two variables, Distance (in the positive direction of the flow) and Time. The rate of change u_t can be calculated according to the following:

$$\frac{\partial U}{\partial t} = \alpha^2 \frac{\partial^2 U}{\partial x^2} - v \frac{\partial U}{\partial x}$$

$\alpha^2 \frac{\partial^2 U}{\partial x^2}$: represents the Diffusion , $v \frac{\partial U}{\partial x}$:represents the convection carrier.

An example about Convection-Diffusion equation, when smoke rises from chimneys, the smoke particles are being carried upward by the hot air, in the same time, particles are being diffused in the air due to air currents, which is explained by the previous equation. [6]

Chapter Three

Finite Difference Methods

3.1. Taylor's Theorem

3.1.1. Taylor's Theorem for Function of One Variables

A function that is continuous and differentiable up to (n+1) derivatives can be approximated by using a finite number of terms of its Taylor series. Taylor's theorem gives quantitative estimates on the error introduced by the use of such an approximation.

The Taylor series method is of general applicability and it is the standard to which we can compare the accuracy of various other numerical methods for solving differential equations problems. It can be used to have any specified degree of accuracy. Taylor series is a series expansion of a function about a point x_0. [1]

If $f(x)$, $f^{/}(x)$, $f^{//}(x)$, $f^{///}(x)$,............,$f^{(n+1)}(x)$ are continuous at x_0:

$$f(x)=f(x_0)+(x-x_0)\ f^{/}(x)+\frac{(x-x_0)^2}{2!}f^{//}(x)+.........+\frac{(x-x_0)^n}{n!}f^{(n)}(x)+\frac{(x-x_0)^{n+1}}{(n+1)!}f^{(n+1)}(\text{c})$$

where $c\in[x,x_0]$.

In another form if f(x) has derivatives of all orders at x_0
then the infinite Taylor series expansion of f(x) about x_0 is:

$$f(x)=\sum_{k=0}^{\infty}\frac{(x-x_0)^k}{k}\ f^{(k)}(\text{x}_0) \qquad (3.1.1.1)$$

The equation (3.1.1.1) is called –One dimensional Taylor theorem – In this method when the number of terms increases then the error is decreased to a certain extent.

Using Taylor's Theorem, we can derive some finite difference formulas that approximate the derivatives

1) Forward-difference formula for $f^{/}(x)$ is

$$f^{/}(x) = \frac{f(x+h)-f(x)}{h} - \frac{h}{2!} f^{//}(c) \qquad x < c < x+h$$

2) Backward-difference formula for $f^{/}(x)$

$$f^{/}(x) = \frac{f(x)-f(x-h)}{h} - \frac{h}{2!} f^{//}(c) \qquad x-h < c < x$$

3) Central-difference formula for $f^{/}(x)$

x-h c1 x c2 x + h

$$\frac{f(x+h)-f(x-h)}{2h} = f^{/}(x) + \frac{h^2}{3}\left[f^{///}(c1)+f^{///}(c2)\right]$$

4) Central-difference formula for $f^{//}(x)$

x-h c1 x c2 x + h

$$f(x+h)+f(x-h) = 2\,f(x) + 2\,\frac{h^2}{2!}\,f^{//}(x) + \frac{h^4}{4!}\left[f^{(4)}(c1)+f^{(4)}(c2)\right]$$

$\frac{h^4}{4!}\left[f^{(4)}(c1)+f^{(4)}(c2)\right]$: this is error term 'second order' = O(h^2)

Drop the error term then:

$$f^{//}(x) = \frac{f(x+h)-2f(x)+f(x-h)}{h^2}-\frac{h^2}{6}\left[f^{(4)}(c_1)+f^{(4)}(c_2)\right]$$

in this way, need x – h and x + h to approximate $f^{//}(x)$.

3.1.2. Taylor's Theorem for Function of Two Variables

Suppose that f (x, y) and its partial derivatives of all order less than or equal to n+1 are continuous on $D = \{(x, y): a \le x \le b \ and \ c \le y \le d\}$ and let $(x_0, y_0) \in D$

for every $(x, y) \in D, \exists \xi \ between \ x \ and \ x_0, \eta \ between \ y \ and \ y_0$ such that :[12][18]

$$f(x,y) = f_0 + \left(\frac{\partial f}{\partial x}\right)_0 (x - x_0) + \left(\frac{\partial f}{\partial y}\right)_0 (y - y_0) + \frac{1}{2!}\left\{\left(\frac{\partial^2 f}{\partial x^2}\right)_0 (x - x_0)^2\right\} + 2\left(\frac{\partial^2 f}{\partial x\, \partial y}\right)_0 (x - x_0)(y - y_0) + \left(\frac{\partial^2 f}{\partial y^2}\right)_0 (y - y_0)^2\bigg\} + \cdots + \frac{1}{n!}\left\{\sum_{j=0}^{n} \binom{n}{j} \left(\frac{\partial^n f}{\partial x^{n-j}\, \partial y^j}\right)(x - x_0)^{n-j}(y - y_0)^j{}_0\right\} + R_n(x,y)$$

Where a subscript zero on f and its derivatives denotes evaluation at (x_0, y_0) and R_n is the remainder,

$$R_n(x,y) = \frac{1}{(n+1)!}\left\{\sum_{j=0}^{n+1} \binom{n+1}{j} \left(\frac{\partial^{n+1} f}{\partial x^{n+1-j}\, \partial y^j}\right)_{(\xi,\eta)} (x - x_0)^{n+1-j}(y - y_0)^j\right\}$$

For example, the Taylor series for f (x, y) about (a, b) is:

$$f(x,y) = f(a,b) + f_x\ (x - a) + f_y\ (y - b) + \frac{1}{2!}\{f_{xx}(x - a)^2 + f_{xy}(x - a)(y - b) + f_{yy}(y - b)^2\}.$$

Where all the derivatives are evaluated at (a, b).

Now, consider a Taylor expansion of an analytical function u(x) and $x_i = x_0 + ih,\ t_j = t_0 + jh$, for $i = 0,1, \dots, N, j = 0,1, \dots, M$

$$u(x + h, t) = u(x) + \sum_{n=1}^{\infty} \frac{h^n}{n!} \frac{\partial^n u}{\partial x^n} \qquad (3.1.2.1)$$

$$u(x+h,t) = u(x) + h\frac{\partial u}{\partial x} + \frac{h^2}{2}\frac{\partial^2 u}{\partial x^2} + \frac{h^3}{3!}\frac{\partial^3 u}{\partial x^3} + \frac{h^4}{4!}\frac{\partial^4 u}{\partial x^4} + \cdots \quad (3.1.2.2)$$

Then for the first derivative one obtains:

$$\frac{\partial u}{\partial x} = \frac{u(x+h) - u(x)}{h} + O(h)$$

Drop the error term then:

$$\frac{\partial u}{\partial x}(x_i, y_j) \approx \frac{u\left(x_{i+1}, y_j\right) - u(x_i, y_j)}{h} \quad (3.1.2.3)$$

Is called a forward difference. And now

$$u(x-h) = u(x) + \sum_{n=1}^{\infty}(-1)^n \frac{h^n}{n!}\frac{\partial^n u}{\partial x^n} \quad (3.1.2.4)$$

$$u(x-h,t) = u(x) - h\frac{\partial u}{\partial x} + \frac{h^2}{2}\frac{\partial^2 u}{\partial x^2} - \frac{h^3}{3!}\frac{\partial^3 u}{\partial x^3} + \frac{h^4}{4!}\frac{\partial^4 u}{\partial x^4} + \cdots \quad (3.1.2.5)$$

So for the first derivative one obtains:

$$\frac{\partial u}{\partial x} = \frac{u(x) - u(x+h)}{h} + O(h)$$

Drop the error term then:

$$\frac{\partial u}{\partial x} \approx \frac{u(x_i, y_j) - u\left(x_{i+1}, y_j\right)}{h} \quad (3.1.2.6)$$

From (3.1.2.2) and (3.1.2.5)

$$\frac{\partial u}{\partial x} = \frac{u(x+h,t) - u(x-h,t)}{2h} + O(h^2)$$

Drop the error term then:

$$\frac{\partial u}{\partial x}\left(x_i, y_j\right) \approx \frac{u\left(x_{i+1}, y_j\right) - u\left(x_{i-1}, y_j\right)}{2h} \tag{3.1.2.7}$$

The second derivative can be found in the same way using the linear combination of different Taylor expansions. For instance, consider

$$u(x + 2h, t) = u(x) + 2h\frac{\partial u}{\partial x} + \frac{(2h)^2}{2}\frac{\partial^2 u}{\partial x^2} + \frac{(2h)^3}{3!}\frac{\partial^3 u}{\partial x^3} + \cdots \tag{3.1.2.8}$$

Subtracting from the last equation (3.1.2.2) multiplied by 2,one get the following equation

$$u(x + 2h, t) - 2\ u(x + 2h) = -u(x) + h^2\ \frac{\partial^2 u}{\partial x^2} + h^3\ \frac{\partial^3 u}{\partial x^3} + \tag{3.1.2.9}$$

Hence one can approximate the second derivative as:

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x + 2h) - 2u(x + h) + u(x)}{h^2} + O(h)$$

Drop the error term then:

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u(x + 2h) - 2u(x + h) + u(x)}{h^2} \tag{3.1.2.10}$$

Similarly

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u(x - 2h) - 2u(x - h) + u(x)}{h^2} \tag{3.1.2.11}$$

Finally, the second derivative reads

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j) \approx \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{h^2} \tag{3.1.2.11}$$

Like wise

$$\frac{\partial u}{\partial y}(x_i, y_j) \approx \frac{u(x_i, y_{j+1}) - u(x_i, y_{j-1})}{2h} \tag{3.1.2.12}$$

$$\frac{\partial^2 u}{\partial y^2}(x_i, y_j) \approx \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1})}{h^2} \tag{3.1.2.13}$$

3.2. Finite Difference Methods

The finite difference method is based on to the approximations that permit replacing differential equation by finite difference equation. There, finite difference approximations are algebraic in form, and the solutions are related to grid points. Thus, a finite difference solution basically involves three steps: -

1) Dividing the unknown solution u(x,t) into grids of nodes of the domain.
2) Approximating the given differential equation by finite difference equivalence that relates the solutions to grid points.
3) Solving the difference equations subject to the prescribed boundary conditions and/or initial conditions.

3.3. Forward-Difference (Explicit) Method

3.3.1. Approximate the Solution Numerically

Consider the simple diffusion problem:

$$\frac{\partial U}{\partial t} = f(x, t) + K\frac{\partial^2 U}{\partial x^2} \quad 0 < x < L, t > 0 \tag{3.3.1.1}$$

$u(0,t) = u(L,t) = 0$ *boundary conditions*

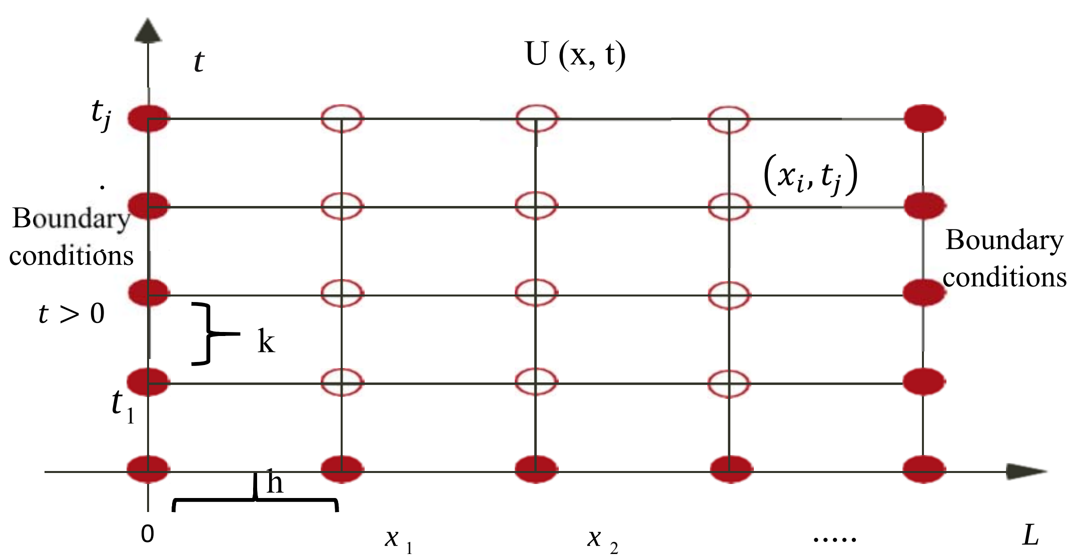
$u(x,0) = g(x)$ *initial condition* $t > 0$ and $0 < x < L$

$f(x,t)$, g(x) , and L are given

Step 1. First introduce the points where we will compute the solution. Unlike the situation with either IVPs or BVPs, the region over which we solve the problem is two-dimensional. So our points have the form (x_i, t_j) .

The step sizes are h = (L – 0)/ (N + 1) where N+1 is the number of subintervals in the interval [0, L] and k = T /M where M is the number of time steps.

Step 2. Evaluate the differential equation at the grid point (x, t) = (x_i, t_j) to obtain



X

Figure 3.1: The grid system used to find the numerical solution of the heat equation.

$$x_i = 0 + ih, i = 0,1,2,......,N, h = \frac{L}{N+1}$$

$$t_j = jk, j = 0,1,2,......,M, k = \frac{T}{M}$$

The above solid grid points are either initial or boundary points where the solution is given. Initial, it is given by $g(x_i)$. On the other boundaries the solution in this example (3.3.1.1) is zero unless the boundary conditions are not homogeneous. The other grid points are the points at which the solution is to be approximated. We need to evaluate the boundary and initial conditions at their respective grid points, but this will be done later once we have taken care of the equation.

Step 3. Evaluate the equation at each interior point $(\mathrm{x}_i, \mathrm{t}_j)$

Where use the first order forward-difference approximation for the time derivative $u_t(x_i, t_j)$ and 2nd order centered difference approximation

$$u_t(x_i,t_j) = f(x_i,t_j) + u_{xx}(x_i,t_j) \ \ where \ \ i = 1,2,..N, j = 1,2,..\mathrm{M} \qquad (3.3.1.2)$$

for the space derivative $u_{xx}(x_i, t_j)$

$$u_t(x_i,t_j) = \frac{u(x_i,t_{j+1}) - u(x_i,t_j)}{k} - \frac{k}{2} \ u_{tt}(x_i,\beta_j) \ , \beta_j \in \left[t_j \ , \ t_{j+1}\right]$$

$$u_{xx}(x_i,t_j) = \frac{u(x_{i+1},t_j) - 2 \ u(x_i,t_j) + u(x_{i-1},t_j)}{h^2} - \frac{h^2}{12} \ u_{xxxx}(\mu_i,\mathrm{t}_j) \ , \ \mu_i \in \left[x_{i-1} \ , \ x_{i+1}\right]$$

Then we replace $u_t(x_i, t_j)$ and $u_{xx}(x_i, t_j)$ in (3.3.1.2), then equation (3.3.1.2) become :

$$\frac{u(x_i,t_{j+1}) - u(x_i,t_j)}{k} - \frac{k}{2} \ u_{tt}(x_i,\beta_j) = f(x_i,t_j) + \frac{u(x_{i+1},t_j) - 2 \ u(x_i,t_j) + u(x_{i-1},t_j)}{h^2} - \frac{h^2}{12} \ u_{xxxx}(\mu_i,\mathrm{t}_j)$$

The truncation error $\tau_{i,j} = \dfrac{k}{2} \ u_{tt}(x_i, \beta_j) - \dfrac{h^2}{12} \ u_{xxxx}(\mu_i, \mathrm{t}_j) = \mathrm{O(k)} + \mathrm{O(h^2)}$

After that we drop the truncation error term , multiply by k , let $\lambda = \frac{k}{h^2}$ and let $u_{i,j}$ be the value that satisfies the new difference equation :

$$u_{i,j+1} = \lambda \ u_{i+1,j} + (1-2\lambda) \ u_{i,j} + \lambda \ u_{i-1,j} - k \ f_{i,j} \qquad (3.3.1.3)$$

$\forall \ i = 1,2,......,N$

$\forall \ j = 0,1,2,.....,M-1$

$u_{i,0} = g(\mathrm{x}_i)$ *initial comdition*

$u_{\mathrm{N+1,j}} = u_{\mathrm{0,j}} = 0$ *boundary conditions*

The derivation of the finite difference approximation of the heat equation problem is now complete. As written in (3.3.1.3) the method is explicit in the sense that $u_{i,j+1}$ is given explicitly in terms of known quantities.

3.3.2. The Stencil and Matrix Form for Explicit Forward-Difference Method is:

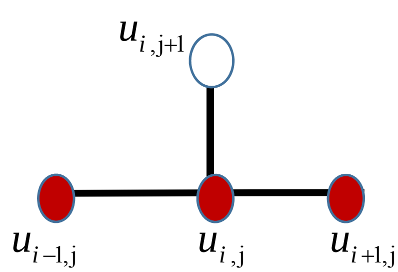


Figure 3.2: The Stencil for Explicit Forward-Difference Method

We can write this equation (3.3.1.3) in matrix system to facilitate numerical analysis:

vector of the unknowns at the time level j is

$u_j = \left[u_{1,j} \ , \ u_{2,j} \ , \ .... \ , \ u_{\mathrm{N},j} \right]^T$ and

$f_j = \left[f_{1,j} \ , \ f_{2,j} \ , \ .... \ , \ f_{\mathrm{N},j} \right]^T$

and the left hand side of equation (3.3.1.3) u_{j+1},then equation depends on the three values $u_{i-1,j}$,$u_{i,j}$,$u_{i+1,j}$ in the vector u_j from time level j .Therefore , for each time level $j = 0,1, \ldots\ldots M-1$ we get the system

$$u_{j+1} = A \ u_j - k \ f_j$$

Where $A = \begin{bmatrix} 1-2\lambda & \lambda & 0 & 0 & 0 & 0 & 0 \\ \lambda & 1-2\lambda & \lambda & 0 & 0 & 0 & 0 \\ 0 & \lambda & 1-2\lambda & \lambda & 0 & 0 & 0 \\ 0 & 0 & \lambda & 1-2\lambda & . & 0 & 0 \\ 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & . & . & \lambda \\ 0 & 0 & 0 & 0 & 0 & \lambda & 1-2\lambda \end{bmatrix}$ ، $\lambda = \dfrac{k}{h^2}$

The system is:

$$\begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ u_{3,j+1} \\ u_{4,j+1} \\ . \\ . \\ u_{N,j+1} \end{bmatrix} = \begin{bmatrix} 1-2\lambda & \lambda & 0 & 0 & 0 & 0 & 0 \\ \lambda & 1-2\lambda & \lambda & 0 & 0 & 0 & 0 \\ 0 & \lambda & 1-2\lambda & \lambda & 0 & 0 & 0 \\ 0 & 0 & \lambda & 1-2\lambda & . & 0 & 0 \\ 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & . & . & \lambda \\ 0 & 0 & 0 & 0 & 0 & \lambda & 1-2\lambda \end{bmatrix} \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ u_{3,j} \\ u_{4,j} \\ . \\ . \\ u_{N,j} \end{bmatrix} - k \begin{bmatrix} f_{1,j} \\ f_{2,j} \\ f_{3,j} \\ f_{4,j} \\ . \\ . \\ f_{N,j} \end{bmatrix}$$

Note that: The first and last equation $u_{0,j} = u_{N+1,j} = 0$ (boundary points).The time level j = 0 is used to approximate the solution at time level j = 1. Then time level j = 1 is used to approximate the solution on time level j = 2 and so on until the solution is approximated on all time levels.

3.3.3. Stability Analysis for Explicit Forward-Difference Method (Fourier Stability Analysis):

Consider the simple diffusion problem:

$$\frac{\partial U}{\partial t} = f(x,t) + \frac{\partial^2 U}{\partial x^2} \qquad 0 < x < L \ , t > 0$$

$$u(0,t) = u(L,t) = 0 \quad boundary \quad conditions$$

$$u(x,0) = g(x) \qquad initial \quad condition$$

g(x) , and L are given

In the method of separation variable one finds that the solution is

$$u(x,t) = \sum_{n=1}^{\infty} A_n e^{-\lambda_n^2 t} \sin(\lambda_n x) \qquad (3.3.3.1)$$

And

$$A_n = 2\int_0^1 g(x)\sin(\lambda_n x)\,dx \qquad (3.3.3.2)$$

To analyze the magnitude of the solution as time increases:

equation that arises for radioactive decay. The approach used for partial differential equations is different, and instead we investigate how the method does with a test solution. The problem considered is the homogeneous version of the differential equation. So, in (3.3.1.1) we set f (x, t) = 0. To explain how the test solution is selected, one sees in (3.3.3.1) that the separation of variables solution consists of the superposition of functions that are oscillatory in x. In deciding whether a numerical method is stable we determine how well the method does with such solutions. In particular, the start-off assumption to decide on stability is that the solution of the finite difference equation has the form [12] [11]

Let $W_J = e^{rx_iI}, I = \sqrt{-1}$, $x_i = i \quad h$, $r > 0$

The form of forward is:

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

Equation implies is :

$$W_{j+1} \ e^{r \ x_i I} = W_j \left[\lambda \ e^{r \ x_{i-1} I} + (1-2\lambda) \ e^{r \ x_i I} + \lambda \ e^{r \ x_{i+1} I} \right]$$

$$W_{j+1} \ e^{r \ x_i I} = W_j \left[\lambda \ e^{r \ (x_i - \mathrm{h}) I} + (1-2\lambda) \ e^{r \ x_i I} + \lambda \ e^{r \ (x_i + \mathrm{h}) I} \right]$$

$$W_{j+1} \ e^{r \ x_i I} = W_j \left[\lambda \ e^{r \ x_i \ I} \ e^{-r \ h \ I} + (1-2\lambda) \ e^{r \ x_i I} + \lambda \ e^{r \ x_i \ I} \ e^{r \ h \ I} \right]$$

$$W_{j+1} \ e^{r \ x_i I} = W_j \ e^{r \ x_i \ I} \left[\lambda \ e^{-r \ h \ I} + (1-2\lambda) + \lambda \ e^{r \ h \ I} \right]$$

$$W_{j+1} = W_j \left[\lambda \ e^{-r \ h \ I} + (1-2\lambda) + \lambda \ e^{r \ h \ I} \right]$$

$$W_{j+1} = W_j \left[\lambda \ (e^{-r \ h \ I} + e^{r \ h \ I}) + 1 - 2\lambda \right]$$

Note that:

$$e^{\theta I} = \cos \ \theta + I \ \sin \ \theta$$

$$e^{-\theta I} = \cos \ \theta - I \ \sin \ \theta$$

$$\cos \ \theta = \frac{e^{-\theta I} + e^{\theta I}}{2}$$

$$W_{j+1} = W_j \left[2 \ \lambda \ \cos(rh) + 1 - 2\lambda \right]$$

$$2\lambda \ \cos(r \ h) = 2\lambda(1 - 2 \ \sin^2(\frac{r \ h}{2}))$$

$$2\lambda \ \cos(r \ h) = 2\lambda - 4\lambda \ \sin^2(\frac{r \ h}{2})$$

$$W_{j+1} = W_j \left[2\lambda - 4\lambda \ \sin^2(\frac{r \ h}{2}) + 1 - 2\lambda \right]$$

$$W_{j+1} = W_j \left[1 - 4\lambda \ \sin^2(\frac{r \ h}{2}) \right]$$

$$\eta = \left[1 - 4\lambda \ \sin^2(\frac{r \ h}{2}) \right] \text{ then } W_{j+1} = W_j \ \eta$$

The factor η is called the amplification factor since it is responsible for 'amplifying' the solution as j increases (i.e. as time increases).
If $|\eta| \succ 1$, then $\eta^j \to \infty \ as \ j \to \infty$

For stability, require this factor to be a fraction or one:

$$|\eta| \le 1$$

For the forward difference method, this means:

$$\left| \left[1 - 4\lambda \ \sin^2(\frac{r \ h}{2}) \right] \right| \le 1$$

$$-1 \le \left| \left[1 - 4\lambda \ \sin^2(\frac{r \ h}{2}) \right] \right| \le 1$$

$$-2 \le -4\lambda \ \sin^2(\frac{r \ h}{2}) \le 0$$

$$0 \le 4\lambda \ \sin^2(\frac{r \ h}{2}) \le 2$$

The left inequality ($0 \le$) is always true

To right inequality is true regardless of r if we choose
$4\lambda \le 2$ since $\sin^2(\frac{r \ h}{2}) \le 1$

Therefore, the method is stable under the condition:

$$\lambda \le \frac{1}{2}$$

Where $\lambda = \frac{k}{h^2}$ which implies the condition

$$2k \le h^2$$

The explicit forward difference is conditionally stable with stability condition

$$\lambda = \frac{k}{h^2} \le \frac{1}{2}$$

So, for stability of this method, the time and space step size must satisfy the relation $2k \le h^2$.

3.3.4. Error Analysis for Explicit Forward-Difference Method:

Error analysis provides a widely applicable framework for analyzing the accuracy of difference method. This type of analysis can also be used if the discrete equations are written in difference form. The result of the analysis is an asymptotic estimate of the error in the method.

Knowing the exact solution gives understanding of the accuracy of the method. From examining the symbolic expressions of the truncation error we can add correction terms to the differential equations in order to increase the numerical accuracy. [12]

$U(x_i, t_j)$: be the exact solution at (x_i, t_j).

$u_{i,j}$:be the exact solution of the difference equation.

$\overline{u_{i,j}}$:be the computed solution of the difference equation using Matlab or any computer software.

Then the actual error is:

$$e_{i,j} = |exact - computed|$$

$$e_{i,j} = |U(x_i, t_j) - \overline{u_{i,j}}|$$

$$e_{i,j} = |U(x_i, t_j) - u_{i,j} + u_{i,j} - \overline{u_{i,j}}|$$

use **triangle inequality** $|a+b| \le |a| + |b|$

$$e_{i,j} \le |U(x_i, t_j) - u_{i,j}| + |u_{i,j} - \overline{u_{i,j}}|$$

$e_{i,j} \le (trancation \quad error) + (rounding \quad error)$

$E_{i,j} = \left| U(x_i, t_j) - u_{i,j} \right|$

$E_{i,j}$: is the error due to truncating error terms in approximating derivatives, assuming exact calculations which is the truncation error.

$\left| u_{i,j} - \overline{u_{i,j}} \right|$:is the error in evaluating the difference equation due to using finite machine which is the rounding error.

This means there are two sources of errors in approximation solution (truncation and rounding) error.

A key issue is how accurate the numerical solution is. The ultimate way of addressing this issue would be to compute the error $\overline{u_{i,j}}$ at the mesh points. This is usually extremely demanding. In very simplified problem settings we may, however, manage to derive formulas for the numerical solution u, and therefore closed form expressions for the error $\overline{u_{i,j}}$. Such special cases can provide considerable insight regarding accuracy and stability, but the results are established for special problems. [12]

The explicit forward-difference Method implies:

$$E_{i,j+1} = \lambda E_{i+1,j} + (1-2\lambda)E_{i,j} + \lambda E_{i-1,j} + k\,\tau_{i,j}$$

$$\tau_{i,j} = \frac{k}{2}\ u_{tt}(x_i, \beta_j) - \frac{h^2}{12}\ u_{xxxx}(\mu_i, t_j) = O(k) + O(h^2)$$

$$E_{0,j} = E_{N+1,j} = E_{i,j} = 0 \quad (initially \quad and \quad boundaries \quad conditions)$$

Assuming stability $\lambda = \frac{k}{h^2} \le \frac{1}{2}$ and $E_j = \max_{0 \le i \le N+1} \left| E_{i,j} \right|$, where E_j is maximum error at the j^{th} time level.

$$\tau_\infty = \max_{i,j} \left| \tau_{i,j} \right|$$

$$\left| E_{i,j+1} \right| = \lambda \left| E_{i+1,j} \right| + (1-2\lambda) \left| E_{i,j} \right| + \lambda \left| E_{i-1,j} \right| + k\,\tau_\infty$$

$$E_{i,j+1} \le \lambda E_j + E_j - 2\lambda E_j + \lambda E_j + k\,\tau_\infty$$

The result is the recursive formula

$$\left|E_{i,j+1}\right| \le E_j + k\,\tau_\infty$$

$$\left|E_{i,j+1}\right| \le E_{j-1} + 2k\,\tau_\infty$$

$$\left|E_{i,j+1}\right| \le E_{j-2} + 3k\,\tau_\infty$$

$$\left|E_{i,j+1}\right| \le E_{j-3} + 4k\,\tau_\infty$$

. . .

. . .

$$\left|E_{i,j+1}\right| \quad \le E_0 + (\mathrm{j}+1)k\,\tau_\infty$$

But (j+1) k = $t_{j+1} \le T \quad and \quad E_0 = 0 \;\;,\;\; 0 \le t \le T$ .Therefore $\left|E_{i,j+1}\right| \le T\,\tau_\infty$

$Now\,,\;\; \tau_\infty = O(k) + O(h^2)$

$E_j = O(k) + O(h^2)$

$Hence\,,\;\; E_{i,j} = u(\mathrm{x}_i, \mathrm{t}_j) - \mathrm{u}_{i,j} = O(k) + O(h^2)$

Means $E_{i,j} \to 0 \;\; as \;\; h \to 0 \;\; and \;\; k \to 0$

Which all of them means the above method is consistent.

3.4. Backward-difference (implicit) Method

3.4.1 Approximate the Solution Numerically

To solve 3.3.1.1 by this method, use 1st order backward-difference approximation for the time derivative $u_t(x_i, t_j)$ and 2nd order centered difference approximation for the space derivative $u_{xx}(x_i, t_j)$:

$$u_t(x_i,t_j) = \frac{u(x_i,t_j) - u(x_i,t_{j-1})}{k} - \frac{k}{2}\; u_{tt}(x_i,\beta_j) \;\;,\beta_j \in \left[t_j \;\;,\;\; t_{j+1}\right]$$

$$u_{xx}(x_i,t_j)=\frac{u(x_{i+1},t_j)-2\ u(x_i,t_j)+u(x_{i-1},t_j)}{h^2}-\frac{h^2}{12}\ u_{xxxx}(\mu_i,t_j)\ ,\ \mu_i\in\left[x_{i-1}\ ,\ x_{i+1}\right]$$

We can replace $u_t(x_i,t_j)$ and $u_{xx}(x_i,t_j)$ in (3.3.1.2), then equation (3.3.1.2) becomes :

$$\frac{u(x_i,t_j)-u(x_i,t_{j-1})}{k}-\frac{k}{2}\ u_{tt}(x_i,\beta_j)=f(x_i,t_j)+\frac{u(x_{i+1},t_j)-2\ u(x_i,t_j)+u(x_{i-1},t_j)}{h^2}-\frac{h^2}{12}\ u_{xxxx}(\mu_i,t_j)$$

The truncation error $\tau_{i,j}=\frac{k}{2}\ u_{tt}(x_i,\beta_j)-\frac{h^2}{12}\ u_{xxxx}(\mu_i,t_j)=O(k)+O(h^2)$

We then drop the truncation error term , multiply by k , let $\lambda=\frac{k}{h^2}$ and let $u_{i,j}$ be the value that satisfies the new difference equation :

$$-u_{i,j-1}=\lambda\ u_{i-1,j}-(1+2\lambda)\ u_{i,j}+\lambda\ u_{i+1,j}-k\ \ f_{i,j} \qquad (3.4.1.1)$$

$\forall\ \ i=1,2,......,N$

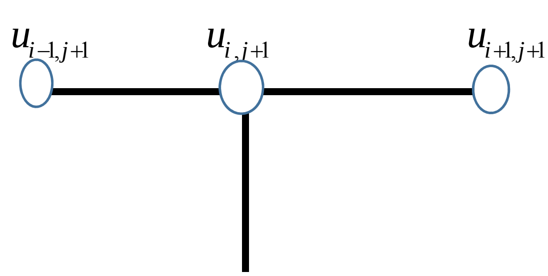
$\forall\ \ j=0,1,2,.....,M$

$u_{i,0}=g(x_i)$ *initial comdition*

$u_{N+1,j}=u_{0,j}=0$ *boundary conditions*

The derivation of the finite difference approximation of the heat equation problem is now complete. As written in (3.4.1.1) the method is explicit in the sense that $u_{i,j+1}$ is given explicitly in terms of known quantities.

3.4.2. The Stencil and Matrix Form for Implicit Backward-Difference Method:



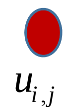


Figure 3.3: The Stencil for Implicit Backward-Difference Method

For each interior point we get a linear equation which contains three unknowns except near the boundaries and initial points $u_{i,j-1}$ is known.

Can write this equation derived above (3.4.1.1) in matrix system to facilitate numerical analysis:

vector of the unknowns at the time level j is

$$u_j = \left[u_{1,j}\ ,\ u_{2,j}\ ,\ \ldots\ldots\ldots\ldots\ ,\ u_{N,j}\right]^T,\ f_j = \left[f_{1,j}\ ,\ f_{2,j}\ ,\ \ldots\ldots\ldots\ldots\ ,\ f_{N,j}\right]^T$$

$$Au_j = u_{j-1} - k\ f_j \quad for \quad j = 1,2,..,M$$

Where $u_0 = g(x)$ and A is the N×N symmetric tridiagonal matrix given as

$$\text{Where } A = \begin{bmatrix} 1+2\lambda & -\lambda & 0 & 0 & 0 & 0 & 0 \\ -\lambda & 1+2\lambda & -\lambda & 0 & 0 & 0 & 0 \\ 0 & -\lambda & 1+2\lambda & -\lambda & 0 & 0 & 0 \\ 0 & 0 & \lambda & 1+2\lambda & . & 0 & 0 \\ 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & . & . & -\lambda \\ 0 & 0 & 0 & 0 & 0 & -\lambda & 1+2\lambda \end{bmatrix}$$

The system is:

$$\begin{bmatrix} 1-2\lambda & \lambda & 0 & 0 & 0 & 0 & 0 \\ \lambda & 1-2\lambda & \lambda & 0 & 0 & 0 & 0 \\ 0 & \lambda & 1-2\lambda & \lambda & 0 & 0 & 0 \\ 0 & 0 & \lambda & 1-2\lambda & . & 0 & 0 \\ 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & . & . & \lambda \\ 0 & 0 & 0 & 0 & 0 & \lambda & 1-2\lambda \end{bmatrix} \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ u_{3,j} \\ u_{4,j} \\ . \\ . \\ u_{N,j} \end{bmatrix} = \begin{bmatrix} u_{1,j-1} \\ u_{2,j-1} \\ u_{3,j-1} \\ u_{4,j-1} \\ . \\ . \\ u_{N,j-1} \end{bmatrix} - k \begin{bmatrix} f_{1,j} \\ f_{2,j} \\ f_{3,j} \\ f_{4,j} \\ . \\ . \\ f_{N,j} \end{bmatrix}$$

3.4.3. Stability Analysis for Explicit Forward-Difference Method is:

Consider the simple diffusion problem:

$$\frac{\partial U}{\partial t} = f(x,t) + \frac{\partial^2 U}{\partial x^2} \qquad 0 < x < L\ , t > 0$$

$u(0,t) = u(L,t) = 0$ *boundary conditions*

$u(x,0) = g(x)$ *initial condition*

$f(x,t)$, g(x) , and L are given

In the method of separation variable one finds that the solution is

$$u(x,t) = \sum_{n=1}^{\infty} A_n e^{-\lambda_n^2 t} \sin(\lambda_n x) \qquad (3.4.3.1)$$

And

$$A_n = 2\int_0^L g(x)\sin(\lambda_n x)\, dx \qquad (3.4.3.2)$$

To analyze the magnitude of the solution as time increases:

equation that arises for radioactive decay. The approach used for partial differential equations is different, and instead we investigate how the method does with a test solution. The problem considered is the homogeneous version of the differential equation. So, in (3.3.1.1) we set f (x, t) = 0. To explain how the test solution is selected, one sees in (3.4.3.1) that the separation of variables solution consists of the superposition of functions that are oscillatory in x. In deciding whether a numerical method is stable we determine how well the method does with such solutions. In particular, the start-off assumption to decide on stability is that the solution of the finite difference equation has the form [12] [11]

Let $W_J = e^{rx_i I}, I = \sqrt{-1}$, $x_i = i \; h$ $, r > 0$

The form of backward is:

$$-u_{i,j-1} = \lambda \; u_{i-1,j} - (1+2\lambda) \; u_{i,j} + \lambda \; u_{i+1,j}$$

Equation implies is :

$$-W_{j-1} \; e^{r \; x_i I} = W_j \left[\lambda \; e^{r \; x_{i-1} I} - (1+2\lambda) \; e^{r \; x_i I} + \lambda \; e^{r \; x_{i+1} I} \right]$$

$$W_{j-1} \; e^{r \; x_i I} = W_j \left[\lambda \; e^{r \; (x_i - \mathrm{h}) I} - (1+2\lambda) \; e^{r \; x_i I} + \lambda \; e^{r \; (x_i + \mathrm{h}) I} \right]$$

$$W_{j-1} \; e^{r \; x_i I} = W_j \left[\lambda \; e^{r \; x_i \; I} \; e^{-r \; h \; I} - (1+2\lambda) \; e^{r \; x_i I} + \lambda \; e^{r \; x_i \; I} \; e^{r \; h \; I} \right]$$

$$W_{j-1} \; e^{r \; x_i I} = W_j \; e^{r \; x_i \; I} \left[\lambda \; e^{-r \; h \; I} - (1+2\lambda) + \lambda \; e^{r \; h \; I} \right]$$

$$W_{j-1} = W_j \left[\lambda \; e^{-r \; h \; I} - (1+2\lambda) + \lambda \; e^{r \; h \; I} \right]$$

$$W_{j-1} = W_j \left[\lambda \; (e^{-r \; h \; I} + e^{r \; h \; I}) - 1 - 2\lambda \right]$$

Note that:

$$e^{\theta I} = \cos \; \theta + I \; \sin \; \theta$$

$$e^{-\theta I} = \cos \; \theta - I \; \sin \; \theta$$

$$\cos \; \theta = \frac{e^{-\theta I} + e^{\theta I}}{2}$$

$$W_{j-1} = W_j \left[2 \; \lambda \; \cos(rh) - 1 - 2\lambda \right]$$

$$2\lambda \; \cos(r \; h) = 2\lambda(1 - 2 \; \sin^2(\frac{r \; h}{2}))$$

$$2\lambda \cos(r h) = 2\lambda - 4\lambda \sin^2(\frac{r h}{2})$$

$$W_{j-1} = W_j \left[2\lambda - 4\lambda \sin^2(\frac{r h}{2}) - 1 - 2\lambda \right]$$

$$W_{j-1} = W_j \left[-1 - 4\lambda \sin^2(\frac{r h}{2}) \right]$$

$$\eta = \frac{1}{1 + 4\lambda \sin^2(\frac{r h}{2})} \text{ then } W_{j-1} = W_j \eta$$

The factor η is called the amplification factor since it is responsible for 'amplifying' the solution as j increases (i.e. as time increases).

If $|\eta| \succ 1$, then $\eta^j \to \infty \ as \ j \to \infty$

For stability, we require this factor to be a fraction or one:

$$|\eta| \le 1$$

For the backward difference method, this means:

$$\frac{1}{1 + 4\lambda \sin^2(\frac{r h}{2})} \le 1$$

$$1 \le 1 + 4\lambda \sin^2(\frac{r h}{2})$$

$$0 \le 4\lambda \sin^2(\frac{r h}{2})$$

Which is always true, then the implicit backward difference is stable without stability condition.

3.5. Crank-Nicolson (Implicit) Method

3.5.1. Approximate the Solution Numerically

In numerical analysis, the Crank–Nicolson method is a finite difference method used to approximate the solution of the heat equation like parabolic partial differential equations. This method is a second order method in time, the method was developed by John Crank and Phyllis Nicolson in the middle 20th century. [13]

Crank-Nicolson method can be derived using the Trapezoid rule:
Consider the simple diffusion problem:

$$\frac{\partial U}{\partial t} = f(x,t) + \frac{\partial^2 U}{\partial x^2} \qquad 0 < x < L, t > 0 \qquad (3.5.1.1)$$

The first partial derivative of time u_t is a function of time t and space x ,then can write the on form [12]

$$u_t = F(x,t) \qquad (3.5.1.2)$$

Integrating both sides of (3.5.1.2) from t_j to t_{j+1} gives

$$\int_{t_j}^{t_{j+1}} u_t .dt = \int_{t_j}^{t_{j+1}} \mathrm{F}(\mathrm{x},\mathrm{t}).dt$$

$$u(x, t_{j+1}) - u(x, t_j) = \int_{t_j}^{t_{j+1}} \mathrm{F}(\mathrm{x},\mathrm{t}).dt \qquad (3.5.1.3)$$

Use Trapezoid rule to determine $\int_{t_j}^{t_{j+1}} \mathrm{F}(\mathrm{x},\mathrm{t}).dt$

$$\int_{t_j}^{t_{j+1}} \mathrm{F}(\mathrm{x},\mathrm{t}).dt = \frac{t_{j+1} - t_i}{2}\left[\mathrm{F}(\mathrm{x},\mathrm{t}_{j+1}) + \mathrm{F}(\mathrm{x},\mathrm{t}_j)\right] + O(h^3)$$

$$\int_{t_j}^{t_{j+1}} \mathrm{F}(\mathrm{x},\mathrm{t}).dt = \frac{k}{2}\left[u_{xx}(\mathrm{x}_i,\mathrm{t}_{j+1}) - F(\mathrm{x}_i,\mathrm{t}_{j+1}) + u_{xx}(\mathrm{x}_i,\mathrm{t}_j) - F(\mathrm{x}_i,\mathrm{t}_j)\right] + O(h^3)$$

Drop the truncation error term , and let $u_{i,j}$ be the value that satisfies the new equation : [12]

$$\int_{t_j}^{t_{j+1}} F(x,t)dt = \frac{k}{2}\left[\frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{h^2} + \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2}\right] - \frac{k}{2}\left[f_{i,j+1} + f_{i,j}\right] \qquad (3.5.1.4)$$

Put (3.5.1.3) in (3.5.1.4) and multiply by k , let $\lambda = \frac{k}{h^2}$ and let $u_{i,j}$ be the value that satisfies then gives new equation :

$$u_{i,j+1} - u_{i,j} = \frac{\lambda}{2}\left[u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1} + u_{i+1,j} - 2u_{i,j} + u_{i-1,j}\right] - \frac{k}{2}\left[f_{i,j+1} + f_{i,j}\right]$$

$j = 0,1,\ldots\ldots M-1, i = 1,\ldots\ldots N$

3.5.2. The Stencil and Matrix Form for Explicit Forward-Difference Method is:

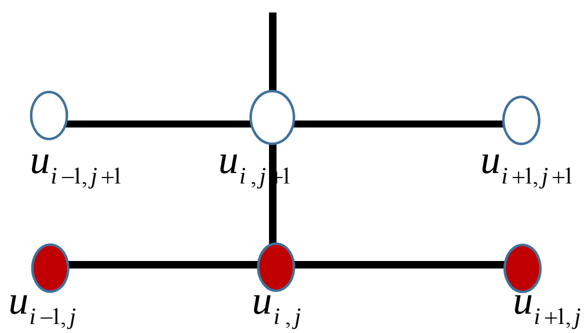


Figure 3.4: The Stencil and Matrix Form for Explicit Forward-Difference Method

That Crank-Nicolson does so much better than either forward and backward method is not unexpected, given its better truncation error.

$$u_j = \left[u_{1,j}\ ,\ u_{2,j}\ ,\ \ldots\ldots\ldots\ ,\ u_{N,j}\right]^T \ and\ \ f_j = \left[f_{1,j}\ ,\ f_{2,j}\ ,\ \ldots\ldots\ldots\ ,\ f_{N,j}\right]^T$$

$$(B+I)\ u_{j+1} = (A+I)\ u_j - k\ \ (f_{j+1} + f_j)\ \ for\ \ j = 0,2,..,M-1$$

Where $u_0 = g(x)$

And
$$B=\begin{bmatrix} 1+2\lambda & -\lambda & 0 & 0 & 0 & 0 & 0 \\ -\lambda & 1+2\lambda & -\lambda & 0 & 0 & 0 & 0 \\ 0 & -\lambda & 1+2\lambda & -\lambda & 0 & 0 & 0 \\ 0 & 0 & \lambda & 1+2\lambda & . & 0 & 0 \\ 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & . & . & -\lambda \\ 0 & 0 & 0 & 0 & 0 & -\lambda & 1+2\lambda \end{bmatrix}$$

$$A=\begin{bmatrix} 1-2\lambda & \lambda & 0 & 0 & 0 & 0 & 0 \\ \lambda & 1-2\lambda & \lambda & 0 & 0 & 0 & 0 \\ 0 & \lambda & 1-2\lambda & \lambda & 0 & 0 & 0 \\ 0 & 0 & \lambda & 1-2\lambda & . & 0 & 0 \\ 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & . & . & \lambda \\ 0 & 0 & 0 & 0 & 0 & \lambda & 1-2\lambda \end{bmatrix}$$

Chapter Four

Method of Line for Solving Parabolic Partial Differential Equation

4.1 Introduction

The method of line(MOL) is a method that enables conversion of parabolic PDEs into sets of ODEs that, in some sense, are equivalent to the PDEs. The basic idea behind the MOL algorithm is discretized along the spatial coordinates only. This approximation is what we call semi discretization. If we discretize in space and leave time continuous, a system of ODEs is obtained.

Then, one of the way salient face of the MOL is the use of existing established numerical methods for ODEs. The derivative of the parabolic PDE problem is approximated by linear combination of function values at the form grid points. On the other hand, arbitrary order approximations can be derived from a Taylor series expansion, [15]

$$U(x) = \sum_{n=0}^{\infty} \frac{(x - x_0)^n}{n!} \left(u^{(n)}(x_0) \right)$$

For parabolic PDE's, an initial semi discretization in space results in a system of ODEs. Solving this system of ODEs yields a discrete solution along lines in time, which is why the way is called method of lines. We exemplify the derivation of such ODEs with a finite difference approximation of the spatial derivatives of the PDE.

The challenge is therefore to convert PDE to ODE's system. Once we have done this, we can apply any algorithm to the initial values of the ODEs to calculate the approximate solution for the PDE (like RK4). Thus, an important of the MOL is the use of current and well distributed ODE's methods, in other words, the basic idea of the MOL is to replace spatial derivatives in the PDE with algebraic estimates. Once this is done, spatial derivatives will not be explicitly mentioned in terms of spatially independent variables in this problem. [15]

4.2. Conversion Partial Differential Equation to System of Ordinary Differential Equations

This is done by introducing approximations for the x derivatives, and then using some methods to solve the resulting problem. To explain this procedure, we carry out the steps for the parabolic heat equation(3.3.1.1), where the boundary conditions are u (0, t) = u (1, t) = 0.

$$u_{xx}(x_i,t) = u_t(x_i,t) + f(x_i,t) \quad where \quad i = 1,2,..N, \tag{4.2.1}$$

[17]

Step (1): Evaluate the equation at each interior point (x_i, t_j)

step (2): Is to evaluate the equation (3.3.1.1) at $x = x_i$, giving us

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t} + f\,(x_i, t) \tag{4.2.2}$$

Step(3): Use 2nd order centered difference approximation for the space derivative $u_{xx}(x_i, t_j)$

$$u_{xx}(x_i, t) = \frac{u(x_{i+1}, t) - 2\,u(x_i, t) + u(x_{i-1}, t)}{h^2} - \frac{h^2}{12}\,u_{xxxx}(\mu_i, \mathrm{t}) \tag{4.2.3}$$

$$\mu_i \in x_0, x_1, \dots, x_N, x_{N+1}$$

Step (4): Substitute the centered difference approximation for the spatial derivative in (4.2.2), we obtain:

$$u_t(x_i, t) = \frac{u(x_{i+1}, t) - 2\,u(x_i, t_j) + u(x_{i-1}, t)}{h^2} - f\,(x_i, \mathrm{t}) + O(\mathrm{h}^2) \tag{4.2.4}$$

$$\mathrm{i} = 0,1,2, \dots \dots \dots .., \mathrm{N}$$

Step (5): Dropping the error term from (4.2.4) yields

$$\frac{d}{dt} u_i = \frac{u_{i+1} - 2\,u_i + u_{i-1}}{h^2} - f_i(\mathrm{t}) + O(\mathrm{h}^2), \text{ for } \mathrm{i} = 0,1,2, \dots \dots \dots .., \mathrm{N}$$

(4.2.5)

Where $u_i(t)$ is the approximation of the implementation of $u\ (\mathrm{x}_i, \mathrm{t})$ and $f_i(\mathrm{t}) = \mathrm{f}(\mathrm{x}_i, t)$. Collecting all u_i together, this last result can be written in vector form as

$$\frac{d}{dt} u_i = Au - f\,(\mathrm{t}), \text{for } \mathrm{t} < 0 \tag{4.2.6}$$

Where

$$u(t)=\begin{bmatrix} u_1(t) \\ u_2(t) \\ . \\ . \\ u_N(t) \end{bmatrix}, f(t)=\begin{bmatrix} f_1(t) \\ f_2(t) \\ . \\ . \\ f_N(t) \end{bmatrix}, \mathrm{A}=\frac{1}{h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & . & . & 0 & 0 & 0 \\ 0 & 0 & 0 & . & . & . & 0 & 0 \\ 0 & 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix} \qquad (4.2.7)$$

Step (6): The initial condition $u\ (\mathrm{x,t}) = g\ (x\ )$ takes the form

$$u(0)=\begin{bmatrix} g_1 \\ g_2 \\ . \\ . \\ g_N \end{bmatrix} \qquad (4.2.8)$$

With (4.2.6) and (4.2.7) we have a system of ODEs.

4.3. Temporal discretization

In our implementation, the governing PDEs are often integrated in time with the classical second-order two-order Runge-Kutta (RK2) method or the fourth-order four-stage Runge-Kutta (RK4) method. Assuming that the governing equation is [15]

$$\left(\frac{\partial U}{\partial t}\right) = R(U) \tag{4.3.1}$$

Where R(U) denotes the residual. [17]

The classical RK4 method integrates from time t_0(step n) to $t_0 + h$ (step n+1)

through the operations

$$\begin{aligned}
U_0 &= u(x, t_0), \quad k_0 = h \quad R(U_0) \\
U_1 &= U_0 + \frac{K_0}{2}, \quad k_1 = h \quad R(U_1) \\
U_2 &= U_1 + \frac{K_1}{2}, \quad k_2 = h \quad R(U_2) \\
U_3 &= U_2 + \frac{K_2}{2}, \quad k_3 = h \quad R(U_3) \\
U_{n+1} &= U_0 + \frac{1}{6}\left[K_0 + 2K_1 + 2K_2 + K_3\right]
\end{aligned} \tag{4.3.2}$$

In order to save computational cost, sometimes the low-order accurate RK2 scheme is preferred. The classical RK2 method integrates from time t_0 (step n) to $t_0 + h$ (step n+1) through the operations

$$
\begin{aligned}
&U_0 = u(x, t_0), \quad k_0 = h \quad R(U_0) \\
&U_1 = U_0 + K_0, \quad k_1 = h \quad R(U_1) \\
&U_2 = U_1 + K_1, \qquad (4.3.3) \\
&U_{n+1} = \frac{1}{2}\left[U_0 + U_2\right]
\end{aligned}
$$

Chapter Five

Pade Approximation Method

5.1. Introduction

There are several polynomials to approximate any continuous differentiable function f(x) on a closed interval, Padé approximants are rational approximations that can be constructed from the coefficients of a given series. The Padé approximant often gives good approximation of the function f(x), Padé approximants are used extensively in computerized environments such as software's like Maple or Matlab. The polynomials are also easily evaluated at arbitrary values. But polynomial approximations have a disadvantage in their tendency for oscillation. This often causes error bounds in polynomial approximation to significantly exceed the average approximation error, because error bounds are determined by the maximum approximation error [1].

Let μ(x) is a rational function of degree N, and

$$\mu(x)=\frac{p_0+p_1x+\cdots+p_nx^n}{q_0+q_1x+\cdots+q_mx^m}=\frac{\sum_{i=0}^{n}p_i\ x^i}{\sum_{i=0}^{m}q_i\ x^i}$$

Where $p(x)$ and $q(x)$ are polynomials whose degree n and m ,and μ(x)is the approximation function for f(x) on a closed interval [a , b].For the interval containing zero, it is required to have $q_0 \neq 0$ in order to make μ(x) defined at zero, but can assume that $q_0=1$, for if this is not the case we simply replace $p(x)$ by $p(x)/q_0$ and $q(x)$ by $q(x)/q_0$.Every polynomial is considered as a rational function if we set $q(x)=1$.

5.2. Technique Padé Approximation

If we consider that Padé Approximation way is the extension of Taylor polynomial approximation to rational functions [1].

If we want to approximate the function f(x) on the period between a and b, then consider the

$$f(x)-\mu(\mathrm{x})=f(x)-\frac{p(x)}{q(x)} \tag{5.2.1}$$

$$f(x)-\mu(\mathrm{x})=\frac{f(x)q(x)-\mathrm{p(x)}}{q(x)} \tag{5.2.2}$$

Now, the Maclaurin series for $f(x)=\sum_{i=0}^{\infty} a_i x^i$,then

$$f(x)-\mu(\mathrm{x})=\frac{\sum_{i=0}^{\infty} a_i x^i \sum_{i=0}^{m} q_i x^i - \sum_{i=0}^{n} p_i x^i}{\sum_{i=0}^{m} q_i x^i} \tag{5.2.3}$$

Now to calculate the values of the constants $q_1, q_2, \ldots\ldots\ldots\ldots, q_m$ and $p_0, p_1, \ldots\ldots\ldots\ldots, p_n$ to obtain that $f^{(j)}(x)-\mu^{(j)}(\mathrm{x})=0 \;\; for \quad j=0,1,\ldots.,N$.

For example the Padé approximation to the $f(x)=e^x$ of degree 6 where n =3 and m=3 can be evaluated as the Maclaurin series for

$$f(x)=\sum_{i=0}^{\infty}\frac{x^i}{i!}=1+x+\frac{x^2}{2!}+\frac{x^3}{3!}+\frac{x^4}{4!}+\ldots\ldots.$$

so we have that $e^x=1+x+\frac{x^2}{2}+\frac{x^3}{6}+\frac{x^4}{24}+\ldots\ldots.$,then

$$\left(1+x+\frac{x^2}{2}+\frac{x^3}{6}+\frac{x^4}{24}+\frac{x^5}{120}+\frac{x^6}{720}+.....\right)\left(1+q_1x+q_2x^2+q_3x^3\right)-\left(p_0+p_1x+p_2x^2+p_3x^3\right)=0$$

And then we give:

$$\left(1+x+\frac{x^2}{2}+\frac{x^3}{6}+\frac{x^4}{24}+\frac{x^5}{120}+\frac{x^6}{720}+.....\right)\left(1+q_1x+q_2x^2+q_3x^3\right)=\left(p_0+p_1x+p_2x^2+p_3x^3\right)$$

Use Maple to expand,

expand $\left(\left(1+x+\frac{x^2}{2}+\frac{x^3}{6}+\frac{x^4}{24}+\frac{x^5}{120}+\frac{x^6}{720}\right)\left(1+q_1x+q_2x^2+q_3x^3\right)\right)$

when run this statement then the result is:

$$1+x+q1\,x+q2\,x^2+q3\,x^3+q3\,x^4+q2\,x^3+q1\,x^2+\frac{1}{2}\,q3\,x^5+\frac{1}{2}\,q2\,x^4+\frac{1}{2}\,q1\,x^3$$
$$+\frac{1}{6}\,q3\,x^6+\frac{1}{6}\,q2\,x^5+\frac{1}{6}\,q1\,x^4+\frac{1}{24}\,q3\,x^7+\frac{1}{24}\,q2\,x^6+\frac{1}{24}\,q1\,x^5+\frac{1}{120}\,q3\,x^8$$
$$+\frac{1}{120}\,q2\,x^7+\frac{1}{120}\,q1\,x^6+\frac{1}{720}\,q3\,x^9+\frac{1}{720}\,q2\,x^8+\frac{1}{720}\,q1\,x^7+\frac{1}{720}\,x^6$$
$$+\frac{1}{120}\,x^5+\frac{1}{24}\,x^4+\frac{1}{6}\,x^3+\frac{1}{2}\,x^2$$

By expanding and collection all of terms , coefficients of x^j for $j=0,1,2,3,4,5,6$ are zeros ,then

$$j=0,x^0:1=p_0$$

$$j=1,x^1:1+q_1=p_1$$

$$j=2,x^2:q_1+q_2+\frac{1}{2}=p_2$$

$$j=3,x^3:q_3+q_2+\frac{1}{2}q_1+\frac{1}{6}=p_3$$

$$j=4,x^4:q_3+\frac{1}{2}q_2+\frac{1}{6}q_1+\frac{1}{24}=0$$

$$j=5,x^5:\frac{1}{2}q_3+\frac{1}{6}q_2+\frac{1}{24}q_1+\frac{1}{120}=0$$

$$j=6,x^6:\frac{1}{6}q_3+\frac{1}{24}q_2+\frac{1}{120}q_1+\frac{1}{720}=0$$

We can write the previous equation in a system as

$$\begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 & -1 & 0 \\ \frac{1}{2} & 1 & 1 & 0 & 0 & -1 \\ \frac{1}{6} & \frac{1}{2} & 1 & 0 & 0 & 0 \\ \frac{1}{24} & \frac{1}{6} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{120} & \frac{1}{24} & \frac{1}{6} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} -1 \\ -\frac{1}{2} \\ -\frac{1}{6} \\ -\frac{1}{24} \\ -\frac{1}{120} \\ -\frac{1}{720} \end{bmatrix}$$

Solve previous system by Matlab then we get that

```
format rat
A = [1 0 0 -1 0 0 ;1 1 0 0 -1 0; 1/2 1 1 0 0 -1; 1/6 1/2 1 0 0 0;1/24 1/6 1/2 0 0 0 ;1/120 1/24 1/6 0 0 0];
b= [-1; -1/2; -1/6; -1/24; -1/120; -1/720];
c=inv(A)*b
```

Then,

$$p_0 = 1, p_1 = \frac{1}{2}, p_2 = \frac{1}{10}, p_3 = \frac{1}{120}, \mathrm{q}_1 = -\frac{1}{2}, \mathrm{q}_2 = \frac{1}{10}, \mathrm{q}_3 = -\frac{1}{120}$$

Then the (3,3) Padé approximation for f(x)=e^x of degree 6 with n=3 and m=3 is

$$e^x = \frac{1 + \frac{1}{2}x + \frac{1}{10}x^2 + \frac{1}{120}x^3}{1 - \frac{1}{2}x + \frac{1}{10}x^2 - \frac{1}{120}x^3}$$

5.3. General Padé Approximants

Padé approximation is a rational approximation where Padé approximant to the function $f(\theta)$ on interval $[a, b]$ is the quotient of two polynomials $P_k(\theta)$ and of degree k and $Q_m(\theta)$ of degree μ. So we denote this quotient

$$M_{k,m}(\theta)=\frac{p_k(\theta)}{q_m(\theta)}+O(\theta^{k+m+1}) \qquad (5.3.1)$$

But if $k = m$,then will be called $M_{k,m}(\theta)$ is a diagonal Padé approximation[2].

Padé approximant $M_{k,m}(\theta)$ will be evaluated for $f(\theta) = e^{\theta}$ as the following [3],[4]

$$P_k(\theta)=\sum_{j=0}^{k}\frac{(m+k-j)!k!}{(m+k)!j!(k-j)!}\theta^j \qquad (5.3.2)$$

And

$$Q_m(\theta)=\sum_{j=0}^{m}\frac{(m+k-j)!m!}{(m+k)!j!(m-j)!}(-\theta)^j \qquad (5.3.3)$$

Now, we can write, the Padé approximation

$$e^{\theta}=\frac{P_k(\theta)}{Q_m(\theta)} \qquad (5.3.4)$$

5.4. Convergence of the Padé Approximants

Following the approach used by Padé and by each successive generation of mathematicians, we shall examine by means of specific examples how approximants imitate the analytic structure of the functions they represent. Then, offer a brief overview of the convergence theorems that have been proven.

If we substitute (k, m), Produce the Padé table

Table 5.1: The Pade' approximation table

$k\ /\ m$	0	1	2	. . .
0		(0,1)	(0,2)	. . .
1	(1,0)	(1,1)	(1,2)	. . .
2	(2,0)	(2,1)	(2,2)	. . .
. . .	. . .	. . .	. . .	. . .

$M_{k,m}(\theta)$ is called the Padé approximant of order $(k+m)$ to e^{θ} and has a leading error term of order $(k+m+1)$.

In general $M(x) = \dfrac{1+p_1\theta+\cdots+p_k\theta^k}{1+q_1\theta+\cdots+q_m\theta^m} + C_{k+m+1}\theta^{k+m+1}$ [11].

5.5. Padé Approximation (1,1) For Solving Parabolic Partial Differential equations

For Padé approximation (1,1) where $k=1$ and $m=1$,using equation (5.3.2) and (5.3.3) we get that

$$P_1(\theta)=\sum_{j=0}^{1}\frac{(2-j)!1!}{(2)!j!(1-j)!}\theta^j=\left[\frac{2!\times 1!}{2!\times 0!\times 1!}\theta^0+\frac{1!\times 1!}{2!\times 1!\times 0!}\theta^1\right]=1+\frac{1}{2}\theta$$

$$Q_1(\theta)=\sum_{j=0}^{1}\frac{(2-j)!1!}{(2)!j!(2-j)!}(-\theta)^j=\left[\frac{2!\times 1!}{2!\times 0!\times 1!}(-\theta)^0+\frac{1!\times 1!}{2!\times 1!\times 1!}(-\theta)^1\right]=1-\frac{1}{2}\theta$$

So

$$e^{\theta} = \frac{1+\frac{1}{2}\theta}{1-\frac{1}{2}\theta}$$

Using Taylor series,

$$u(t+k)=u(t)+(k)\ u^{/}(t)+\frac{(k)^2}{2!}\ u^{//}(t)+\frac{(k)^3}{3!}\ u^{///}(c1)\quad ، c1\in[t ، t+k] \qquad (5.5.1)$$

It is possible to write the previous equation on the form

$$u(t+k)\cong u(t)+kDu(t)+\frac{k^2}{2!}D^2u(t)$$

So

$$u(t+k)\cong u(t)\left[1+kD+\frac{k^2}{2!}D^2\right]$$

And

$$u(t+k) = u(t)\ e^{kD} \qquad (5.5.2)$$

Where D is the differential operator and is a step size distance between any sub interval. Now using the relation in the above relation and e^{kD} can be replaced by Padé approximation with operator D , get that

$$u(t+k) = \frac{1+\frac{kD}{2}}{1-\frac{kD}{2}}\ u(t)$$

But when u(x) is a vector, this is becoming

$$\left[I-\frac{kD}{2}\right]\ u(t+k) = \left[I+\frac{kD}{2}\right]\ u(t) \qquad (5.5.3)$$

Now consider the following given PDE with the conditions

$$\frac{\partial u}{\partial t} = f\ (x_i, t) + \frac{\partial^2 u}{\partial x^2} \qquad 0 < x < L\,, t > 0 \tag{5.5.4}$$

$u(0,t) = u(L,t) = 0$ *boundary conditions*

$u(x,0) = g(x)$ *initial condition*

g(x) , and L are given

Assume that the boundary value associated with (5.5.3) are zero, and by (5.5.4) the (2,2) Padé approximant approximates

$$u(t+k) = u(t)\ e^{kD} \tag{5.5.5}$$

And substituting value of e^{kD} then

$$u(t+k) = u(t) \left[\frac{1+\frac{1}{2}kD}{1-\frac{1}{2}kD} \right] \tag{5.5.6}$$

Equation (5.5.6) in vector – matrix form becomes

$$\left(I - \frac{1}{2}kD \right) u(t+k) = \left(I + \frac{1}{2}kD \right) u(t) \tag{5.5.7}$$

Then on discrete points t_j,

$$\left(I - \frac{1}{2}kD \right) u(t_j + k) = \left(I + \frac{1}{2}kD \right) u(t_j) \quad j = 0,1,2,... \tag{5.5.8}$$

Where $u(t_j + k) = \begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ u_{3,j+1} \\ . \\ . \\ u_{N-1,j+1} \end{bmatrix}$ and matrix $D = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix}$

Applying them on the discrete point where $t_j = t_0 + jk = jk$

$$\left(I - \frac{1}{2}kD\right) =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} - \frac{k}{2h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix}$$

Let be $r = \frac{k}{h^2}$ then $\left(I - \frac{1}{2}kD\right) =$

$$\begin{bmatrix} 1+r & -\frac{1}{2}r & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2}r & 1+r & -\frac{1}{2}r & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{2}r & 1+r & -\frac{1}{2}r & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{2}r & 1+r & -\frac{1}{2}r & 0 & 0 & 0 \\ 0 & 0 & . & . & . & . & . & 0 \\ 0 & 0 & 0 & . & . & . & . & 0 \\ 0 & 0 & 0 & 0 & . & . & . & -\frac{1}{2}r \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2}r & 1+r \end{bmatrix}$$

Now

$$\left(I+\frac{1}{2}kD\right)=$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} + \frac{k}{2h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix}$$

And then

Let be $r=\frac{k}{h^2}$ then $\left(I+\frac{1}{2}kD\right)=$

$$\begin{bmatrix} 1-r & \frac{1}{2}r & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2}r & 1-r & \frac{1}{2}r & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}r & 1-r & \frac{1}{2}r & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2}r & 1-r^2 & \frac{1}{2}r & 0 & 0 & 0 \\ 0 & 0 & . & . & . & . & . & 0 \\ 0 & 0 & 0 & . & . & . & . & 0 \\ 0 & 0 & 0 & 0 & . & . & . & \frac{1}{2}r \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2}r & 1-r \end{bmatrix}$$

Finally, after substituting all the previous matrices in (5.5.8), produce the system:

$$\begin{bmatrix} 1+r & -\frac{1}{2}r & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2}r & 1+r & -\frac{1}{2}r & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{2}r & 1+r & -\frac{1}{2}r & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{2}r & 1+r & -\frac{1}{2}r & 0 & 0 & 0 \\ 0 & 0 & . & . & . & . & . & 0 \\ 0 & 0 & 0 & . & . & . & . & 0 \\ 0 & 0 & 0 & 0 & . & . & . & -\frac{1}{2}r \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2}r & 1+r \end{bmatrix} \begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ u_{3,j+1} \\ u_{4,j+1} \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j+1} \end{bmatrix} =$$

$$\begin{bmatrix} 1-r & \frac{1}{2}r & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2}r & 1-r & \frac{1}{2}r & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}r & 1-r & \frac{1}{2}r & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2}r & 1-r^2 & \frac{1}{2}r & 0 & 0 & 0 \\ 0 & 0 & . & . & . & . & . & 0 \\ 0 & 0 & 0 & . & . & . & . & 0 \\ 0 & 0 & 0 & 0 & . & . & . & \frac{1}{2}r \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2}r & 1-r \end{bmatrix} \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ u_{3,j} \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j} \end{bmatrix} \quad j=0,1,....$$

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

$$for \quad i=2,3,4,............,N-1 \quad , \quad j=1,2,3,..........,\mathrm{N}-1$$

The previous linear system equation can be solved using LU-Decomposition to get the approximated solution.

5.6. Padé Approximation (0,2) For Solving Parabolic Partial Differential Equations

For Padé approximation (0,2) where $k = 0$ and $m = 2$,using equation (5.3.2) and (5.3.3) we get that

$$P_0(\theta)=1$$

$$Q_2(\theta)=\sum_{j=0}^{2}\frac{(2-j)!2!}{(2)!j!(2-j)!}(-\theta)^j=\left[\frac{2!\times2!}{2!\times0!\times2!}(-\theta)^0+\frac{1!\times2!}{2!\times1!\times1!}(-\theta)^1+\frac{0!\times2!}{2!\times2!\times0!}(-\theta)^2\right]=1-\theta+\frac{1}{2}\theta^2$$

So

$$e^{\theta}=\frac{1}{1-\theta+\frac{1}{2}\theta^2}$$

Using Taylor series,

$$u(t+k)=u(t)+(k)\ u^{/}(t)+\frac{(k)^2}{2!}\ u^{//}(t)+\frac{(k)^3}{3!}\ u^{///}(c1)\quad ،\ c1\in[t\ ،\ t+k] \qquad (5.6.1)$$

It is possible to write the previous equation (5.6.1) on the form

$$u(t+k)\cong u(t)+kDu(t)+\frac{k^2}{2!}D^2u(t)$$

So

$$u(t+k)\cong u(t)\left[1+kD+\frac{k^2}{2!}D^2\right]$$

And

$$u(t+k)\ =\ u(t)\ e^{kD} \qquad (5.6.2)$$

Where D is the differential operator and a step size distance between any sub interval. Now using the relation in (5.5.2), which can be replaced by Padé approximation with operator D, get that

$$u(t+k)\quad=\frac{1}{1-(kD)+\frac{1}{2}(kD)^2}\ u(t)$$

But when u(t) is a vector, it becomes

$$\left[1-(kD)+\frac{1}{2}(kD)^2\right]\ u(t+k)\ =\left[I\right]\ u(t) \tag{5.6.3}$$

Now consider the following given PDE with the conditions

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

(5.6.4)

$u(0,t)=u(L,t)=0\ \ boundary\ \ \ conditions$

$u(x,0)=g(x)\qquad initial\ \ \ condition$

g(x) , and L are given

After we convert (5.6.4) to be system $\frac{d}{dt}u_i=Au-f\ (\mathrm{t}\ ),\mathrm{for}\ \ t\prec 0$, the (2,1) Padé approximant approximates

$$u(t+k)\ =\ u(t)\ e^{kD} \tag{5.6.5}$$

And substituting value of e^{kD} then

$$u(t+k)\ =\ u(t)\ \left[\frac{1}{1-(kD)+\frac{1}{2}(kD)^2}\right] \tag{5.6.6}$$

In vector – matrix form (5.6.6) becomes

$$\left(1-(kD)+\frac{1}{2}(kD)^2\right)\ u(t+k)=\ u(t) \tag{5.6.7}$$

Then

$$\left(1-(kD)+\frac{1}{2}(kD)^2\right)\ u(t_j+k)=\ u(t_j)\ \ j=0,1,2,... \tag{5.6.8}$$

Where $u(t_j + k) = \begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ u_{3,j+1} \\ . \\ . \\ u_{N-1,j+1} \end{bmatrix}$ and matrix $D = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix}$

Now

$$\left(1 - (kD) + \frac{(kD)^2}{2}\right) =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} - \frac{k}{h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix} +$$

$$\frac{k^2}{2}\left(\frac{1}{h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix}\right)^2$$

Let be $r = \frac{k}{h^2}$ then

$$\left(1-(kD)+\frac{(kD)^2}{2}\right)=$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} -2r & r & 0 & 0 & 0 & 0 & 0 & 0 \\ r & -2r & r & 0 & 0 & 0 & 0 & 0 \\ 0 & r & -2r & r & 0 & 0 & 0 & 0 \\ 0 & 0 & r & -2r & r & 0 & 0 & 0 \\ 0 & 0 & 0 & r & -2r & r & 0 & 0 \\ 0 & 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & . & . & r \\ 0 & 0 & 0 & 0 & 0 & 0 & r & -2r \end{bmatrix} +$$

$$\left(\frac{1}{2}\begin{bmatrix} 5r^2 & -4r^2 & 1r^2 & 0 & 0 & 0 & 0 & 0 \\ -4r^2 & 6r^2 & -4r^2 & 1r^2 & 0 & 0 & 0 & 0 \\ 1r^2 & -4r^2 & 6r^2 & -4r^2 & 1r^2 & 0 & 0 & 0 \\ 0 & 1r^2 & -4r^2 & 6r^2 & -4r^2 & 1r^2 & 0 & 0 \\ 0 & 0 & 1r^2 & . & . & . & . & 0 \\ 0 & 0 & 0 & . & . & . & . & 1r^2 \\ 0 & 0 & 0 & 0 & . & . & . & -4r^2 \\ 0 & 0 & 0 & 0 & 0 & 1r^2 & -4r^2 & 5r^2 \end{bmatrix}\right)$$

Then

$$\left(1-(kD)+\frac{(kD)^2}{2}\right)=$$

$$\begin{bmatrix} 1+2r+\frac{5r^2}{2} & -r-2r^2 & \frac{r^2}{2} & 0 & 0 & 0 & 0 & 0 \\ -r-2r^2 & 1+2r+3r^2 & -r-2r^2 & \frac{r^2}{2} & 0 & 0 & 0 & 0 \\ \frac{r^2}{2} & -r-2r^2 & 1+2r+3r^2 & -r-2r^2 & \frac{r^2}{2} & 0 & 0 & 0 \\ 0 & \frac{r^2}{2} & -r-2r^2 & 1+2r+3r^2 & -r-2r^2 & \frac{r^2}{2} & 0 & 0 \\ 0 & 0 & . & . & . & . & . & 0 \\ 0 & 0 & 0 & . & . & . & . & \frac{r^2}{2} \\ 0 & 0 & 0 & 0 & . & . & 1+2r+3r^2 & -r-2r^2 \\ 0 & 0 & 0 & 0 & 0 & \frac{r^2}{2} & -r-2r^2 & 1+2r+\frac{5r^2}{2} \end{bmatrix}$$

Finally, after substituting all the previous matrices in (5.6.9), we produce the system:

$$\begin{bmatrix} 1+2r+\frac{5r^2}{2} & -r-2r^2 & \frac{r^2}{2} & 0 & 0 & 0 & 0 & 0 \\ -r-2r^2 & 1+2r+3r^2 & -r-2r^2 & \frac{r^2}{2} & 0 & 0 & 0 & 0 \\ \frac{r^2}{2} & -r-2r^2 & 1+2r+3r^2 & -r-2r^2 & \frac{r^2}{2} & 0 & 0 & 0 \\ 0 & \frac{r^2}{2} & -r-2r^2 & 1+2r+3r^2 & -r-2r^2 & \frac{r^2}{2} & 0 & 0 \\ 0 & 0 & . & . & . & . & . & 0 \\ 0 & 0 & 0 & . & . & . & . & \frac{r^2}{2} \\ 0 & 0 & 0 & 0 & . & . & 1+2r+3r^2 & -r-2r^2 \\ 0 & 0 & 0 & 0 & 0 & \frac{r^2}{2} & -r-2r^2 & 1+2r+\frac{5r^2}{2} \end{bmatrix} \begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ u_{3,j+1} \\ u_{4,j+1} \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j+1} \end{bmatrix} = \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ u_{3,j} \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j} \end{bmatrix} \quad j=0,1,2,\ldots$$

Or can be written as

$$\frac{1}{2}r^2\left[u_{i-2,j+1}+u_{i+2,j+1}\right]-(\mathrm{r}+2\mathrm{r}^2)\left[u_{i-1,j+1}+u_{i+1,j+1}\right]+1+2r+3r^2\left[u_{i,j+1}\right]=u_{i,j}$$

$for \quad i = 2,3,4,............,N-1 \quad , \quad j = 1,2,3,..........,\mathrm{N}-1$

The previous linear system equation can be solved using LU-Decomposition to get the approximated solution.

5.7. Padé Approximation (2,0) For Solving Parabolic Partial Differential Equations

For Padé approximation (2,0) where $k = 2$ and $m = 0$,using equation (5.3.2) and (5.3.3) we get that

$$P_2(\theta)=\sum_{j=0}^{2}\frac{(2-j)!2!}{(2)!j!(2-j)!}\theta^j=\left[\frac{2!\times 2!}{2!\times 0!\times 2!}(\theta)^0+\frac{1!\times 2!}{2!\times 1!\times 1!}(\theta)^1+\frac{0!\times 2!}{2!\times 2!\times 0!}(\theta)^2\right]=1+\theta+\frac{1}{2}\theta^2$$

$$Q_0(\theta)=1$$

So

$$e^{\theta}=\frac{1+\theta+\frac{1}{2}\theta^2}{1}=1+\theta+\frac{1}{2}\theta^2$$

Using Taylor series,

$$u(\mathrm{t}+k)=u(\mathrm{t})+(k)\ u^{/}(t)+\frac{(h)^2}{2!}\ u^{//}(t)+\frac{(\mathrm{k})^3}{3!}\ u^{///}(c1) \quad ، \ c1\in[t \ ، \ t+k] \tag{5.7.1}$$

It is possible to write the previous equation (5.7.1) on the form

$$u(t+k)\cong u(t)+kDu(t)+\frac{k^2}{2!}D^2u(t)$$

So

$$u(t+k)\cong u(\mathrm{t})\left[1+kD+\frac{k^2}{2!}D^2\right]$$

And

$$u(t+k) = u(\mathrm{t})\ e^{kD} \tag{5.7.2}$$

Where D is the differential operator and a step size distance between any sub interval. Now using the relation in (5.5.2), which can be replaced by Padé approximation with operator D, get that

$$u(t+k) = \left[1+(kD)+\frac{(kD)^2}{2}\right] u(t)$$

But when u(t) is a vector, this is becoming

$$u(t+k) = \left[I+(kD)+\frac{1}{2}(kD)^2\right] u(t) \qquad (5.7.3)$$

Now consider the following given PDE with the conditions

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

(5.7.4)

$u(0,t) = u(L,t) = 0$ *boundary conditions*

$u(x,0) = g(x)$ *initial condition*

g(x) , and L are given

After converting (5.7.4) to be system $\frac{d}{dt}u_i = Au - f(\mathrm{t})$, for $t < 0$, and by (5.7.4) the (2,0) Padé approximant approximates

$$u(t+k) = u(t)\ e^{kD} \qquad (5.7.5)$$

And substituting value of e^{kD} then

$$u(t+k) = u(t)\left[I+(kD)+\frac{1}{2}(kD)^2\right] \qquad (5.7.6)$$

Then

$$u(t_j+k) = \left(I+(kD)+\frac{1}{2}(kD)^2\right) u(t_j) \quad j=0,1,2,... \qquad (5.7.7)$$

Where $u(t_j+k)=\begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ u_{3,j+1} \\ . \\ . \\ u_{N-1,j+1} \end{bmatrix}$ and matrix $D=\frac{1}{h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix}$

Now

$$\left(I + (kD) + \frac{1}{2}(kD)^2 \right) =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} + \frac{k}{h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix} +$$

$$\frac{k^2}{2}\left(\frac{1}{h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix} \right)^2$$

Let be $r=\frac{k}{h^2}$ then

$$\left(1+(kD)+\frac{(kD)^2}{2}\right)=$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} -2r & r & 0 & 0 & 0 & 0 & 0 & 0 \\ r & -2r & r & 0 & 0 & 0 & 0 & 0 \\ 0 & r & -2r & r & 0 & 0 & 0 & 0 \\ 0 & 0 & r & -2r & r & 0 & 0 & 0 \\ 0 & 0 & 0 & r & -2r & r & 0 & 0 \\ 0 & 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & . & . & r \\ 0 & 0 & 0 & 0 & 0 & 0 & r & -2r \end{bmatrix} +$$

$$\left(\frac{1}{2}\begin{bmatrix} 5r^2 & -4r^2 & 1r^2 & 0 & 0 & 0 & 0 & 0 \\ -4r^2 & 6r^2 & -4r^2 & 1r^2 & 0 & 0 & 0 & 0 \\ 1r^2 & -4r^2 & 6r^2 & -4r^2 & 1r^2 & 0 & 0 & 0 \\ 0 & 1r^2 & -4r^2 & 6r^2 & -4r^2 & 1r^2 & 0 & 0 \\ 0 & 0 & 1r^2 & . & . & . & . & 0 \\ 0 & 0 & 0 & . & . & . & . & 1r^2 \\ 0 & 0 & 0 & 0 & . & . & . & -4r^2 \\ 0 & 0 & 0 & 0 & 0 & 1r^2 & -4r^2 & 5r^2 \end{bmatrix}\right)$$

$$\left(1+(kD)+\frac{(kD)^2}{2}\right)=$$

$$\begin{bmatrix} 1-2r+\frac{5r^2}{2} & r-2r^2 & \frac{r^2}{2} & 0 & 0 & 0 & 0 & 0 \\ r-2r^2 & 1-2r+3r^2 & r-2r^2 & \frac{r^2}{2} & 0 & 0 & 0 & 0 \\ \frac{r^2}{2} & r-2r^2 & 1-2r+3r^2 & r-2r^2 & \frac{r^2}{2} & 0 & 0 & 0 \\ 0 & \frac{r^2}{2} & r-2r^2 & 1-2r+3r^2 & r-2r^2 & \frac{r^2}{2} & 0 & 0 \\ 0 & 0 & . & . & . & . & . & 0 \\ 0 & 0 & 0 & . & . & . & . & \frac{r^2}{2} \\ 0 & 0 & 0 & 0 & . & . & 1-2r+3r^2 & -r-2r^2 \\ 0 & 0 & 0 & 0 & 0 & \frac{r^2}{2} & r-2r^2 & 1-2r+\frac{5r^2}{2} \end{bmatrix}$$

Finally, after substituting all the previous matrices in (5.7.7), we produce the system:

$$\begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ u_{3,j+1} \\ u_{4,j+1} \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j+1} \end{bmatrix} = \begin{bmatrix} 1-2r+\frac{5r^2}{2} & r-2r^2 & \frac{r^2}{2} & 0 & 0 & 0 & 0 & 0 \\ r-2r^2 & 1-2r+3r^2 & r-2r^2 & \frac{r^2}{2} & 0 & 0 & 0 & 0 \\ \frac{r^2}{2} & r-2r^2 & 1-2r+3r^2 & r-2r^2 & \frac{r^2}{2} & 0 & 0 & 0 \\ 0 & \frac{r^2}{2} & r-2r^2 & 1-2r+3r^2 & r-2r^2 & \frac{r^2}{2} & 0 & 0 \\ 0 & 0 & . & . & . & . & . & 0 \\ 0 & 0 & 0 & . & . & . & . & \frac{r^2}{2} \\ 0 & 0 & 0 & 0 & . & . & 1-2r+3r^2 & -r-2r^2 \\ 0 & 0 & 0 & 0 & 0 & \frac{r^2}{2} & r-2r^2 & 1-2r+\frac{5r^2}{2} \end{bmatrix} \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ u_{3,j} \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j} \end{bmatrix} \quad j=0,1,2,\ldots$$

Or can write as

$$\frac{1}{2}r^2\left[u_{i-2,j}+u_{i+2,j}\right]+(\mathrm{r}-2\mathrm{r}^2)\left[u_{i-1,j}+u_{i+1,j}\right]+1-2r+3r^2\left[u_{i,j}\right]=u_{i,j+1}$$

$$for \quad i = 2,3,4,............,N-1 \quad , \quad j = 1,2,3,..........,\mathrm{N}-1$$

The previous linear system equation can be solved using LU-Decomposition to get the approximated solution.

5.8 Padé Approximation (2,1) For Solving Parabolic Partial Differential Equations

For Padé approximation (1,2) where $k = 2$ and $\mu = 1$,using equation (5.3.2) and (5.3.3) we get that

$$P_2(\theta) = \sum_{j=0}^{2} \frac{(3-j)!2!}{(3)!j!(2-j)!}\theta^j = \left[\frac{3!\times 2!}{3!\times 0!\times 2!}\theta^0 + \frac{2!\times 2!}{3!\times 1!\times 1!}\theta^1 + \frac{1!\times 2!}{3!\times 2!\times 0!}\theta^2\right] = 1 + \frac{2}{3}\theta + \frac{1}{6}\theta^2$$

$$Q_1(\theta) = \sum_{j=0}^{1} \frac{(3-j)!1!}{(3)!j!(1-j)!}(-\theta)^j = \left[\frac{3!\times 1!}{3!\times 0!\times 1!}(-\theta)^0 + \frac{2!\times 1!}{3!\times 1!\times 0!}(-\theta)^1\right] = 1 - \frac{1}{3}\theta$$

So

$$e^{\theta} = \frac{1 + \frac{2}{3}\theta + \frac{1}{6}\theta^2}{1 - \frac{1}{3}\theta}$$

Using Taylor series,

$$u(\mathrm{t}+k) = u(\mathrm{t}) + (k)\ u'(t) + \frac{(k)^2}{2!}\ u''(t) + \frac{(\mathrm{k})^3}{3!}\ u'''(c1) \quad ، \ c1 \in [t \ ، \ t+k] \qquad (5.8.1)$$

It is possible to write the previous equation (5.8.1) on the form

$$u(t+k) \cong u(t) + kDu(t) + \frac{k^2}{2!} D^2 u(t)$$

So

$$u(t+k) \cong u(\mathrm{t}) \left[1 + kD + \frac{k^2}{2!} D^2 \right]$$

And

$$u(t+k) \; = \; u(\mathrm{t}) \; e^{kD} \tag{5.8.2}$$

Where D is the differential operator and a step size distance between any sub interval. Now using the relation in (5.8.2), can be replaced by Padé approximation with operator D, get that

$$u(t+k) \quad = \frac{1 + \frac{2(kD)}{3} + \frac{(kD)^2}{6}}{1 - \frac{(kD)}{3}} \; u(t)$$

But when u(t) is a vector, it becomes

$$\left[I - \frac{(kD)}{3} \right] u(t+k) \; = \left[I + \frac{2(kD)}{3} + \frac{(kD)^2}{6} \right] u(t) \tag{5.8.3}$$

Now consider the following given PDE with the conditions

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

(5.8.4)

$u(0,t) = u(L,t) = 0 \quad boundary \quad conditions$

$u(x,0) = g(x) \qquad initial \quad condition$

g(x) , and L are given

After converting (5.8.4) to be system $\frac{d}{dt}u_i = Au - f(\mathrm{t}),$for $t < 0$, and by (5.8.4) the (1,2) Padé approximant approximates

$$u(t+k) = u(t)\ e^{kD} \tag{5.8.5}$$

And substituting value of e^{kD} then

$$u(t+k) = u(t)\left[\frac{1+\frac{2(kD)}{3}+\frac{(kD)^2}{6}}{1-\frac{(kD)}{3}}\right] \tag{5.8.6}$$

In vector-matrix form (5.8.6) becomes

$$\left(I-\frac{(kD)}{3}\right)\ u(t+k) = \left(I+\frac{2(kD)}{3}+\frac{(kD)^2}{6}\right)\ u(t) \tag{5.8.8}$$

Then

$$\left(I-\frac{(kD)}{3}\right)\ u(t_j+k) = \left(I+\frac{2(kD)}{3}+\frac{(kD)^2}{6}\right)\ u(t_j)\quad j=0,1,2,... \tag{5.8.9}$$

Where $u(t_j+k) = \begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ u_{3,j+1} \\ . \\ . \\ u_{N-1,j+1} \end{bmatrix}$ and matrix $D = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix}$

Applying them on the discrete point where $t_j = t_0 + jk = jk$

$$\left(I-\frac{(kD)}{3}\right)=\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&.&0&0\\0&0&.&.&.&0\\0&0&0&.&.&0\\0&0&0&0&0&1\end{bmatrix}-\frac{k}{3h^2}\begin{bmatrix}-2&1&0&0&0&0\\1&-2&1&0&0&0\\0&1&-2&.&0&0\\0&0&.&.&.&0\\0&0&0&.&.&1\\0&0&0&0&1&-2\end{bmatrix}$$

Let be $r=\frac{k}{h^2}$ then $\left(I-\frac{(kD)}{3}\right)$

$$\left(I-\frac{(kD)}{3}\right)=\begin{bmatrix}1+\frac{2r}{3}&\frac{-r}{3}&0&0&0&0&0&0\\\frac{-r}{3}&1+\frac{2r}{3}&\frac{-r}{3}&0&0&0&0&0\\0&\frac{-r}{3}&1+\frac{2r}{3}&\frac{-r}{3}&0&0&0&0\\0&0&\frac{-r}{3}&1+\frac{2r}{3}&\frac{-r}{3}&0&0&0\\0&0&0&\frac{-r}{3}&1+\frac{2r}{3}&\frac{-r}{3}&0&0\\0&0&0&0&.&.&.&0\\0&0&0&0&0&.&.&\frac{-r}{3}\\0&0&0&0&0&0&\frac{-r}{3}&1+\frac{2r}{3}\end{bmatrix}$$

Now

$$\left(I + \frac{2(kD)}{3} + \frac{(kD)^2}{6} \right) =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} + \frac{2k}{3h^2} \begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix} +$$

$$\frac{k^2}{6} \left(\frac{1}{h^2} \begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix} \right)^2$$

And then

$$\left(I + \frac{2(kD)}{3} + \frac{(kD)^2}{6} \right) =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} + \frac{2k}{3h^2} \begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix} +$$

$$\left(\frac{k^2}{6h^4} \begin{bmatrix} 5 & -4 & 1 & 0 & 0 & 0 & 0 & 0 \\ -4 & 6 & -4 & 1 & 0 & 0 & 0 & 0 \\ 1 & -4 & 6 & -4 & 1 & 0 & 0 & 0 \\ 0 & 1 & -4 & 6 & -4 & 1 & 0 & 0 \\ 0 & 0 & 1 & . & . & . & . & 0 \\ 0 & 0 & 0 & . & . & . & . & 1 \\ 0 & 0 & 0 & 0 & . & . & . & -4 \\ 0 & 0 & 0 & 0 & 0 & 1 & -4 & 5 \end{bmatrix} \right)$$

Let be $r = \dfrac{k}{h^2}$ then $\left(I + \dfrac{2(kD)}{3} + \dfrac{(kD)^2}{6} \right) =$

$$\begin{bmatrix} 1-\frac{4r}{3}+\frac{5r^2}{6} & \frac{2r}{3}-\frac{2r^2}{3} & \frac{r^2}{6} & 0 & 0 & 0 & 0 & 0 \\ \frac{2r}{3}-\frac{2r^2}{3} & 1-\frac{4r}{3}+r^2 & \frac{2r}{3}-\frac{2r^2}{3} & \frac{r^2}{6} & 0 & 0 & 0 & 0 \\ \frac{r^2}{6} & \frac{2r}{3}-\frac{2r^2}{3} & 1-\frac{4r}{3}+r^2 & \frac{2r}{3}-\frac{2r^2}{3} & \frac{r^2}{6} & 0 & 0 & 0 \\ 0 & \frac{r^2}{6} & \frac{2r}{3}-\frac{2r^2}{3} & 1-\frac{4r}{3}+r^2 & \frac{2r}{3}-\frac{2r^2}{3} & \frac{r^2}{6} & 0 & 0 \\ 0 & 0 & . & . & . & . & . & 0 \\ 0 & 0 & 0 & . & . & . & . & \frac{r^2}{6} \\ 0 & 0 & 0 & 0 & . & . & . & \frac{2r}{3}-\frac{2r^2}{3} \\ 0 & 0 & 0 & 0 & 0 & \frac{r^2}{6} & \frac{2r}{3}-\frac{2r^2}{3} & 1-\frac{4r}{3}+\frac{5r^2}{6} \end{bmatrix}$$

Finally, after substituting all the previous matrices in (5.8.9), produce the system:

$$\begin{bmatrix} 1+\frac{2r}{3} & \frac{-r}{3} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{-r}{3} & 1+\frac{2r}{3} & \frac{-r}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{-r}{3} & 1+\frac{2r}{3} & \frac{-r}{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-r}{3} & 1+\frac{2r}{3} & \frac{-r}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{-r}{3} & 1+\frac{2r}{3} & \frac{-r}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & . & . & \frac{-r}{3} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{-r}{3} & 1+\frac{2r}{3} \end{bmatrix} \begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ u_{3,j+1} \\ u_{4,j+1} \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j+1} \end{bmatrix} =$$

$$\begin{bmatrix} 1-\frac{4r}{3}+\frac{5r^2}{6} & \frac{2r}{3}-\frac{2r^2}{3} & \frac{r^2}{6} & 0 & 0 & 0 & 0 & 0 \\ \frac{2r}{3}-\frac{2r^2}{3} & 1-\frac{4r}{3}+r^2 & \frac{2r}{3}-\frac{2r^2}{3} & \frac{r^2}{6} & 0 & 0 & 0 & 0 \\ \frac{r^2}{6} & \frac{2r}{3}-\frac{2r^2}{3} & 1-\frac{4r}{3}+r^2 & \frac{2r}{3}-\frac{2r^2}{3} & \frac{r^2}{6} & 0 & 0 & 0 \\ 0 & \frac{r^2}{6} & \frac{2r}{3}-\frac{2r^2}{3} & 1-\frac{4r}{3}+r^2 & \frac{2r}{3}-\frac{2r^2}{3} & \frac{r^2}{6} & 0 & 0 \\ 0 & 0 & . & . & . & . & . & 0 \\ 0 & 0 & 0 & . & . & . & . & \frac{r^2}{6} \\ 0 & 0 & 0 & 0 & . & . & . & \frac{2r}{3}-\frac{2r^2}{3} \\ 0 & 0 & 0 & 0 & 0 & \frac{r^2}{6} & \frac{2r}{3}-\frac{2r^2}{3} & 1-\frac{4r}{3}+\frac{5r^2}{6} \end{bmatrix} \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ u_{3,j} \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j} \end{bmatrix} \quad j = 0,1,2,...$$

Or can be written as

$$\frac{1}{3}r\left[u_{i-1,j+1} + u_{i+1,j+1}\right] + 1 + \frac{2}{3}r\left[u_{i,j+1}\right] =$$

$$\frac{1}{6}r^2\left[u_{i-2,j} + u_{i+2,j}\right] + \frac{2}{3}r - \frac{2}{3}r^2\left[u_{i-1,j} + u_{i+1,j}\right] + (1 - \frac{4}{3}r + r^2)\left[u_{i,j}\right]$$

$$for \quad i = 2,3,4,............,N-1 \quad , \quad j = 1,2,3,..........,N-1$$

The previous linear system equation can be solved using LU-Decomposition to get the approximated solution.

5.9. Padé Approximation (1,2) For Solving Parabolic Partial Differential Equations

For Padé approximation (2,1) where $k = 1$ and $m = 2$,using equation (5.3.2) and (5.3.3) we get that

$$P_1(\theta) = \sum_{j=0}^{1} \frac{(3-j)!1!}{(3)!\,j\,!(1-j)!}\theta^j = \left[\frac{3!\times 1!}{3!\times 0!\times 1!}\theta^0 + \frac{2!\times 1!}{3!\times 1!\times 0!}\theta^1\right] = 1 + \frac{1}{3}\theta$$

$$Q_2(\theta) = \sum_{j=0}^{2} \frac{(3-j)!2!}{(3)!\,j\,!(2-j)!}(-\theta)^j = \left[\frac{3!\times 2!}{3!\times 0!\times 2!}(-\theta)^0 + \frac{2!\times 2!}{3!\times 1!\times 1!}(-\theta)^1 + \frac{1!\times 2!}{3!\times 2!\times 0!}(-\theta)^2\right] = 1 - \frac{2}{3}\theta + \frac{1}{6}\theta^2$$

So, $$e^{\theta} = \frac{1 + \frac{1}{3}\theta}{1 - \frac{2}{3}\theta + \frac{1}{6}\theta^2}$$

Using Taylor series,

$$u(t+k) = u(t) + (k)\ u'(t) + \frac{(k)^2}{2!}\ u''(t) + \frac{(k)^3}{3!}\ u'''(c1) \quad ، \ c1 \in [t \ ، \ t+k] \qquad (5.9.1)$$

It is possible to write the previous equation (5.9.1) on the form

$$u(t+k) \cong u(t) + kDu(t) + \frac{k^2}{2!}D^2u(t)$$

So

$$u(t+k) \cong u(t)\left[1 + kD + \frac{k^2}{2!}D^2\right]$$

And

$$u(t+k) = u(\mathrm{t})\ e^{kD} \tag{5.9.2}$$

Where D is the differential operator and is a step size distance between any sub interval. We get that

$$u(t+k) \quad = \frac{1+\frac{(kD)}{3}}{1-\frac{2(kD)}{3}+\frac{(kD)^2}{6}}\ u(t)$$

But when u(t) is a vector, this is becoming

$$\left[I-\frac{2(kD)}{3}+\frac{(kD)^2}{6}\right]\ u(t+k) \quad = \left[I+\frac{(kD)}{3}\right]\ u(t) \tag{5.9.3}$$

Now consider the following given PDE with the conditions

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

(5.9.4)

$u(0,t) = u(L,t) = 0$ *boundary conditions*

$u(x,0) = g(x)$ *initial condition*

g(x) , and L are given

After converting (5.9.4) to be system $\frac{d}{dt}u_i = Au - f\ (\mathrm{t}\ )$,for $t < 0$, and by (5.9.4) the (2,1) Padé approximant approximates

$$u(t+k) = u(t)\ e^{kD} \tag{5.9.5}$$

And substituting value of e^{kD} then

$$u(t+k) \;=\; u(t)\left[\frac{1+\frac{(kD)}{3}}{1-\frac{2(kD)}{3}+\frac{(kD)^2}{6}}\right] \qquad (5.9.6)$$

Multiplying both sides (5.9.7) In the amount $\left(I-\frac{2(kD)}{3}+\frac{(kD)^2}{6}\right)$, this is becoming

$$\left(I-\frac{2(kD)}{3}+\frac{(kD)^2}{6}\right)u(t+k)=\left(I+\frac{(kD)}{3}\right)u(t) \qquad (5.9.8)$$

Then

$$\left(I-\frac{2(kD)}{3}+\frac{(kD)^2}{6}\right)u(t_j+k)=\left(I+\frac{(kD)}{3}\right)u(t_j)\quad j=0,1,2,... \qquad (5.9.9)$$

Where $u(t_j+k)=\begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ u_{3,j+1} \\ . \\ . \\ u_{N-1,j+1} \end{bmatrix}$ and matrix $D=\frac{1}{h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix}$

Applying them on the discrete point where $t_j=t_0+jk=jk$

$$\left(I-\frac{2(kD)}{3}+\frac{(kD)^2}{6}\right)=$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} - \frac{2k}{3h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix} +$$

$$\frac{k^{2}}{6}\left(\frac{1}{h^{2}}\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix}\right)^{2}$$

And then

$$\left(I-\frac{2(kD)}{3}+\frac{(kD)^2}{6}\right)=$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} - \frac{2k}{3h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix} +$$

$$\left(\frac{k^2}{6h^4}\begin{bmatrix} 5 & -4 & 1 & 0 & 0 & 0 & 0 & 0 \\ -4 & 6 & -4 & 1 & 0 & 0 & 0 & 0 \\ 1 & -4 & 6 & -4 & 1 & 0 & 0 & 0 \\ 0 & 1 & -4 & 6 & -4 & 1 & 0 & 0 \\ 0 & 0 & 1 & . & . & . & . & 0 \\ 0 & 0 & 0 & . & . & . & . & 1 \\ 0 & 0 & 0 & 0 & . & . & . & -4 \\ 0 & 0 & 0 & 0 & 0 & 1 & -4 & 5 \end{bmatrix}\right)$$

Let be $r=\dfrac{k}{h^2}$ then $\left(I-\dfrac{2(kD)}{3}+\dfrac{(kD)^2}{6}\right)=$

$$\begin{bmatrix} 1+\frac{4r}{3}+\frac{5r^2}{6} & \frac{-2r}{3}-\frac{2r^2}{3} & \frac{r^2}{6} & 0 & 0 & 0 & 0 & 0 \\ \frac{-2r}{3}-\frac{2r^2}{3} & 1+\frac{4r}{3}+r^2 & \frac{-2r}{3}-\frac{2r^2}{3} & \frac{r^2}{6} & 0 & 0 & 0 & 0 \\ \frac{r^2}{6} & \frac{-2r}{3}-\frac{2r^2}{3} & 1+\frac{4r}{3}+r^2 & \frac{-2r}{3}-\frac{2r^2}{3} & \frac{r^2}{6} & 0 & 0 & 0 \\ 0 & \frac{r^2}{6} & \frac{-2r}{3}-\frac{2r^2}{3} & 1+\frac{4r}{3}+r^2 & \frac{-2r}{3}-\frac{2r^2}{3} & \frac{r^2}{6} & 0 & 0 \\ 0 & 0 & . & . & . & . & . & 0 \\ 0 & 0 & 0 & . & . & . & . & \frac{r^2}{6} \\ 0 & 0 & 0 & 0 & . & . & . & \frac{-2r}{3}-\frac{2r^2}{3} \\ 0 & 0 & 0 & 0 & 0 & \frac{r^2}{6} & \frac{-2r}{3}-\frac{2r^2}{3} & 1+\frac{4r}{3}+\frac{5r^2}{6} \end{bmatrix}$$

Now

$$\left(I+\frac{(kD)}{3}\right)=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}+\frac{k}{3h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix}$$

Let be $r=\frac{k}{h^2}$ then $\left(I+\frac{(kD)}{3}\right)$

$$\left(I+\frac{(kD)}{3}\right)=\begin{bmatrix} 1-\frac{2r}{3} & \frac{r}{3} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{r}{3} & 1-\frac{2r}{3} & \frac{r}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{r}{3} & 1-\frac{2r}{3} & \frac{r}{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{r}{3} & 1-\frac{2r}{3} & \frac{r}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{r}{3} & 1-\frac{2r}{3} & \frac{r}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & . & . & \frac{r}{3} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{r}{3} & 1-\frac{2r}{3} \end{bmatrix}$$

Finally, after substituting all the previous matrices in (5.9.9), produce the system:

$$\begin{bmatrix} 1+\frac{4r}{3}+\frac{5r^2}{6} & \frac{-2r}{3}-\frac{2r^2}{3} & \frac{r^2}{6} & 0 & 0 & 0 & 0 & 0 \\ \frac{-2r}{3}-\frac{2r^2}{3} & 1+\frac{4r}{3}+r^2 & \frac{-2r}{3}-\frac{2r^2}{3} & \frac{r^2}{6} & 0 & 0 & 0 & 0 \\ \frac{r^2}{6} & \frac{-2r}{3}-\frac{2r^2}{3} & 1+\frac{4r}{3}+r^2 & \frac{-2r}{3}-\frac{2r^2}{3} & \frac{r^2}{6} & 0 & 0 & 0 \\ 0 & \frac{r^2}{6} & \frac{-2r}{3}-\frac{2r^2}{3} & 1+\frac{4r}{3}+r^2 & \frac{-2r}{3}-\frac{2r^2}{3} & \frac{r^2}{6} & 0 & 0 \\ 0 & 0 & . & . & . & . & . & 0 \\ 0 & 0 & 0 & . & . & . & . & \frac{r^2}{6} \\ 0 & 0 & 0 & 0 & . & . & . & \frac{-2r}{3}-\frac{2r^2}{3} \\ 0 & 0 & 0 & 0 & 0 & \frac{r^2}{6} & \frac{-2r}{3}-\frac{2r^2}{3} & 1+\frac{4r}{3}+\frac{5r^2}{6} \end{bmatrix}\begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ u_{3,j+1} \\ u_{4,j+1} \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j+1} \end{bmatrix}=$$

$$\begin{bmatrix} 1-\frac{2r}{3} & \frac{r}{3} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{r}{3} & 1-\frac{2r}{3} & \frac{r}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{r}{3} & 1-\frac{2r}{3} & \frac{r}{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{r}{3} & 1-\frac{2r}{3} & \frac{r}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{r}{3} & 1-\frac{2r}{3} & \frac{r}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & . & . & \frac{r}{3} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{r}{3} & 1-\frac{2r}{3} \end{bmatrix} \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ u_{3,j} \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j} \end{bmatrix} \quad j=0,1,2,...$$

Or can write as

$$\frac{1}{6}r^2\left[u_{i-2,j+1}+u_{i+2,j+1}\right]-\frac{2}{3}(r^2+r)\left[u_{i-1,j+1}+u_{i+1,j+1}\right]+(r^2+\frac{4}{3}r+1)\left[u_{i,j+1}\right]=$$
$$\frac{1}{3}r\left[u_{i-1,j}+u_{i+1,j}\right]+(1-\frac{2}{3}r)\left[u_{i,j}\right]$$

$$for \quad i=2,3,4,............,N-1 \quad , \quad j=1,2,3,..........,\mathrm{N}-1$$

The previous linear system equation can be solved using LU-Decomposition to get the approximated solution.

5.10. Padé Approximation (2,2) For Solving Parabolic Partial Differential Equations

For Padé approximation (2,2) where $k = 2$ and $m = 2$,using equation (5.3.2) and (5.3.3) we get that

$$P_2(\theta)=\sum_{j=0}^{2}\frac{(4-j)!2!}{(4)!j!(2-j)!}\theta^j=\left[\frac{4!\times 2!}{4!\times 0!\times 2!}\theta^0+\frac{3!\times 2!}{4!\times 1!\times 1!}\theta^1+\frac{2!\times 2!}{4!\times 2!\times 0!}\theta^2\right]=1+\frac{1}{2}\theta+\frac{1}{12}\theta^2$$

$$Q_2(\theta)=\sum_{j=0}^{\mu}\frac{(4-j)!2!}{(4)!j!(2-j)!}(-\theta)^j=\left[\frac{4!\times 2!}{4!\times 0!\times 2!}(-\theta)^0+\frac{3!\times 2!}{4!\times 1!\times 1!}(-\theta)^1+\frac{2!\times 2!}{4!\times 2!\times 0!}(-\theta)^2\right]=1-\frac{1}{2}\theta+\frac{1}{12}\theta^2$$

So

$$e^{\theta}=\frac{1+\frac{1}{2}\theta+\frac{1}{12}\theta^2}{1-\frac{1}{2}\theta+\frac{1}{12}\theta^2}$$

Using Taylor series,

$$u(t+k)=u(t)+(k)\ u'(t)+\frac{(k)^2}{2!}\ u''(t)+\frac{(k)^3}{3!}\ u'''(c1)\quad ،\ c1\in[t\ ،\ t+k] \tag{5.10.1}$$

It is possible to write the previous equation on the form

$$u(t+k)\cong u(t)+kDu(t)+\frac{k^2}{2!}D^2u(t)$$

So

$$u(t+k)\cong u(t)\left[1+kD+\frac{k^2}{2!}D^2\right]$$

And

$$u(t+k)\ =\ u(t)\ e^{kD} \tag{5.10.2}$$

Where D is the differential operator and a step size distance between any sub interval, get that

$$u(t+k)\ =\frac{1+\frac{kD}{2}+\frac{(kD)^2}{12}}{1-\frac{kD}{2}+\frac{(kD)^2}{12}}\ u(t)$$

But when u(t) is a vector, this is becoming

$$\left[I-\frac{kD}{2}+\frac{(kD)^2}{12}\right]u(t+k)=\left[I+\frac{kD}{2}+\frac{(kD)^2}{12}\right]u(t) \qquad (5.10.3)$$

And

$$\left[I+\frac{kD}{2}+\frac{(kD)^2}{12}\right]u(t-k)=\left[I-\frac{kD}{2}+\frac{(kD)^2}{12}\right]u(t) \qquad (5.10.4)$$

Now consider the following given PDE with the conditions

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

(5.10.5)

$u(0,t)=u(L,t)=0$ *boundary conditions*

$u(x,0)=g(x)$ *initial condition*

g(x) , and L are given

After converting (5.10.5) to be system $\frac{d}{dt}u_i = Au - f(\mathrm{t})$, for $t < 0$, and by (5.10.5) the (2,2) Padé approximant approximates

$$u(t+k) = u(t)\ e^{kD} \qquad (5.10.6)$$

And substituting value of e^{kD} then

$$u(t+k) = u(t)\left[\frac{1+\frac{1}{2}kD+\frac{1}{12}(kD)^2}{1-\frac{1}{2}kD+\frac{1}{12}(kD)^2}\right] \qquad (5.10.7)$$

Equation (5.10.7) in vector – matrix form becomes

$$\left(I-\frac{1}{2}kD+\frac{1}{12}(kD)^2\right)u(t+k)=\left(I+\frac{1}{2}kD+\frac{1}{12}(kD)^2\right)u(t) \qquad (5.10.8)$$

Then on discrete point t_j

$$\left(I-\frac{1}{2}kD+\frac{1}{12}(kD)^2\right)u(t_j+k)=\left(I+\frac{1}{2}kD+\frac{1}{12}(kD)^2\right)u(t_j) \quad j=0,1,2,... \qquad (5.10.9)$$

Where $u(t_j+k)=\begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ u_{3,j+1} \\ . \\ . \\ u_{N-1,j+1} \end{bmatrix}$ and matrix $D=\frac{1}{h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix}$

Applying them on the discrete point where $t_j=t_0+jk=jk$

$$\left(I-\frac{1}{2}kD+\frac{1}{12}(kD)^2\right)=$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}-\frac{k}{2h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix}+\frac{k^2}{12}\left(\frac{1}{h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix}\right)^2$$

And then

$$\left(I-\frac{1}{2}kD+\frac{1}{12}(kD)^2\right)=$$

$$\begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&.&.&.&0\\ 0&0&0&0&0&.&.&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix}-\frac{k}{2h^2}\begin{bmatrix} -2&1&0&0&0&0&0&0\\ 1&-2&1&0&0&0&0&0\\ 0&1&-2&1&0&0&0&0\\ 0&0&1&-2&1&0&0&0\\ 0&0&0&1&-2&1&0&0\\ 0&0&0&0&.&.&.&0\\ 0&0&0&0&0&.&.&1\\ 0&0&0&0&0&0&1&-2 \end{bmatrix}+\left(\frac{k^2}{12h^4}\begin{bmatrix} 5&-4&1&0&0&0&0&0\\ -4&6&-4&1&0&0&0&0\\ 1&-4&6&-4&1&0&0&0\\ 0&1&-4&6&-4&1&0&0\\ 0&0&1&.&.&.&.&0\\ 0&0&0&.&.&.&.&1\\ 0&0&0&0&.&.&.&-4\\ 0&0&0&0&0&1&-4&5 \end{bmatrix}\right)$$

Let be $r=\dfrac{k}{h^2}$ then $\left(I-\dfrac{1}{2}kD+\dfrac{1}{12}(kD)^2\right)=$

$$\begin{bmatrix} 1+r+\frac{5}{12}r^2 & -\frac{1}{2}r-\frac{1}{3}r^2 & \frac{1}{12}r^2 & 0 & 0 & 0 & 0 & 0\\ -\frac{1}{2}r-\frac{1}{3}r^2 & 1+r+\frac{1}{2}r^2 & -\frac{1}{2}r-\frac{1}{3}r^2 & \frac{1}{12}r^2 & 0 & 0 & 0 & 0\\ \frac{1}{12}r^2 & -\frac{1}{2}r-\frac{1}{3}r^2 & 1+r+\frac{1}{2}r^2 & -\frac{1}{2}r-\frac{1}{3}r^2 & \frac{1}{12}r^2 & 0 & 0 & 0\\ 0 & \frac{1}{12}r^2 & -\frac{1}{2}r-\frac{1}{3}r^2 & 1+r+\frac{1}{2}r^2 & -\frac{1}{2}r-\frac{1}{3}r^2 & \frac{1}{12}r^2 & 0 & 0\\ 0 & 0 & . & . & . & . & . & 0\\ 0 & 0 & 0 & . & . & . & . & \frac{1}{12}r^2\\ 0 & 0 & 0 & 0 & . & . & . & -\frac{1}{2}r-\frac{1}{3}r^2\\ 0 & 0 & 0 & 0 & 0 & \frac{1}{12}r^2 & -\frac{1}{2}r-\frac{1}{3}r^2 & 1+r+\frac{5}{12}r^2 \end{bmatrix}$$

Now

$$\left(I + \frac{1}{2}kD + \frac{1}{12}(kD)^2 \right) =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} + \frac{k}{2h^2} \begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix} +$$

$$\frac{k^2}{12} \left(\frac{1}{h^2} \begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix} \right)^2$$

And then

$$\left(I+\frac{1}{2}kD+\frac{1}{12}(kD)^2\right)=$$

$$\begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&.&.&.&0\\ 0&0&0&0&0&.&.&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix}+\frac{k}{2h^2}\begin{bmatrix} -2&1&0&0&0&0&0&0\\ 1&-2&1&0&0&0&0&0\\ 0&1&-2&1&0&0&0&0\\ 0&0&1&-2&1&0&0&0\\ 0&0&0&1&-2&1&0&0\\ 0&0&0&0&.&.&.&0\\ 0&0&0&0&0&.&.&1\\ 0&0&0&0&0&0&1&-2 \end{bmatrix}+$$

$$\left(\frac{k^2}{12h^4}\begin{bmatrix} 5&-4&1&0&0&0&0&0\\ -4&6&-4&1&0&0&0&0\\ 1&-4&6&-4&1&0&0&0\\ 0&1&-4&6&-4&1&0&0\\ 0&0&1&.&.&.&.&0\\ 0&0&0&.&.&.&.&1\\ 0&0&0&0&.&.&.&-4\\ 0&0&0&0&0&1&-4&5 \end{bmatrix}\right)$$

Let be $r=\frac{k}{h^2}$ then $\left(I+\frac{1}{2}kD+\frac{1}{12}(kD)^2\right)=$

$$\begin{bmatrix}
1-r+\frac{5}{12}r^2 & \frac{1}{2}r-\frac{1}{3}r^2 & \frac{1}{12}r^2 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2}r-\frac{1}{3}r^2 & 1-r+\frac{1}{2}r^2 & \frac{1}{2}r-\frac{1}{3}r^2 & \frac{1}{12}r^2 & 0 & 0 & 0 & 0 \\
\frac{1}{12}r^2 & \frac{1}{2}r-\frac{1}{3}r^2 & 1-r+\frac{1}{2}r^2 & \frac{1}{2}r-\frac{1}{3}r^2 & \frac{1}{12}r^2 & 0 & 0 & 0 \\
0 & \frac{1}{12}r^2 & \frac{1}{2}r-\frac{1}{3}r^2 & 1-r+\frac{1}{2}r^2 & \frac{1}{2}r-\frac{1}{3}r^2 & \frac{1}{12}r^2 & 0 & 0 \\
0 & 0 & . & . & . & . & . & 0 \\
0 & 0 & 0 & . & . & . & . & \frac{1}{12}r^2 \\
0 & 0 & 0 & 0 & . & . & . & \frac{1}{2}r-\frac{1}{3}r^2 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{12}r^2 & \frac{1}{2}r-\frac{1}{3}r^2 & 1-r+\frac{5}{12}r^2
\end{bmatrix}$$

Finally, after substituting all the previous matrices in (5.10.10), produce the system:

$$\begin{bmatrix}
1+r+\frac{5}{12}r^2 & -\frac{1}{2}r-\frac{1}{3}r^2 & \frac{1}{12}r^2 & 0 & 0 & 0 & 0 & 0 \\
-\frac{1}{2}r-\frac{1}{3}r^2 & 1+r+\frac{1}{2}r^2 & -\frac{1}{2}r-\frac{1}{3}r^2 & \frac{1}{12}r^2 & 0 & 0 & 0 & 0 \\
\frac{1}{12}r^2 & -\frac{1}{2}r-\frac{1}{3}r^2 & 1+r+\frac{1}{2}r^2 & -\frac{1}{2}r-\frac{1}{3}r^2 & \frac{1}{12}r^2 & 0 & 0 & 0 \\
0 & \frac{1}{12}r^2 & -\frac{1}{2}r-\frac{1}{3}r^2 & 1+r+\frac{1}{2}r^2 & -\frac{1}{2}r-\frac{1}{3}r^2 & \frac{1}{12}r^2 & 0 & 0 \\
0 & 0 & . & . & . & . & . & 0 \\
0 & 0 & 0 & . & . & . & . & \frac{1}{12}r^2 \\
0 & 0 & 0 & 0 & . & . & . & -\frac{1}{2}r-\frac{1}{3}r^2 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{12}r^2 & -\frac{1}{2}r-\frac{1}{3}r^2 & 1+r+\frac{5}{12}r^2
\end{bmatrix}
\begin{bmatrix}
u_{1,j+1} \\ u_{2,j+1} \\ u_{3,j+1} \\ u_{4,j+1} \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j+1}
\end{bmatrix} =$$

$$
\begin{bmatrix}
1-r+\frac{5}{12}r^2 & \frac{1}{2}r-\frac{1}{3}r^2 & \frac{1}{12}r^2 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2}r-\frac{1}{3}r^2 & 1-r+\frac{1}{2}r^2 & \frac{1}{2}r-\frac{1}{3}r^2 & \frac{1}{12}r^2 & 0 & 0 & 0 & 0 \\
\frac{1}{12}r^2 & \frac{1}{2}r-\frac{1}{3}r^2 & 1-r+\frac{1}{2}r^2 & \frac{1}{2}r-\frac{1}{3}r^2 & \frac{1}{12}r^2 & 0 & 0 & 0 \\
0 & \frac{1}{12}r^2 & \frac{1}{2}r-\frac{1}{3}r^2 & 1-r+\frac{1}{2}r^2 & \frac{1}{2}r-\frac{1}{3}r^2 & \frac{1}{12}r^2 & 0 & 0 \\
0 & 0 & . & . & . & . & . & 0 \\
0 & 0 & 0 & . & . & . & . & \frac{1}{12}r^2 \\
0 & 0 & 0 & 0 & . & . & . & \frac{1}{2}r-\frac{1}{3}r^2 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{12}r^2 & \frac{1}{2}r-\frac{1}{3}r^2 & 1-r+\frac{5}{12}r^2
\end{bmatrix}
\begin{bmatrix}
u_{1,j} \\ u_{2,j} \\ u_{3,j} \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j}
\end{bmatrix}
$$

$$
\frac{1}{12}r^2\left[u_{i-2,j+1}+u_{i+2,j+1}\right]-\left(\frac{-1}{3}r^2-\frac{1}{2}r\right)\left[u_{i-1,j+1}+u_{i+1,j+1}\right]+\left(\frac{1}{2}r^2+r+1\right)\left[u_{i,j+1}\right]=
$$

$$
\frac{1}{12}r^2\left[u_{i-2,j+1}+u_{i+2,j+1}\right]+\left(\frac{-1}{3}r^2+\frac{1}{2}r\right)\left[u_{i-1,j}+u_{i+1,j}\right]+\left(\frac{1}{2}r^2-r+1\right)\left[u_{i,j}\right]
$$

$$
for \quad i=2,3,4,.............,N-1 \quad , \quad j=1,2,3,...........,\mathrm{N}-1
$$

The previous linear system equation can be solved using LU-Decomposition to get the approximated solution.

5.11. Padé Approximation Stencils and Type of Solution, and Order of Error Term

In mathematics, especially the areas of numerical analysis concentrating on the numerical solution of partial differential equations, a stencil is a geometric arrangement of a nodal group that relate to the point of interest by using a numerical approximation routine. Stencils are the basis for many algorithms to numerically solve partial differential equations (PDE).

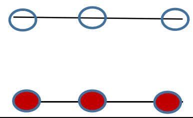
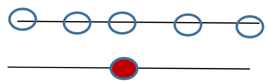
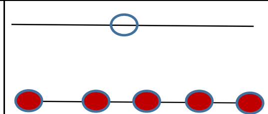
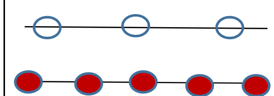
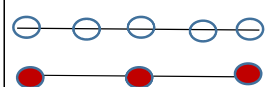
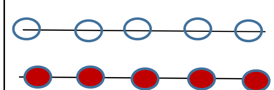
Stencils are classified into two categories: Explicit and Implicit, the difference being the layers from the point of interest that are also used for calculation.

In the notation used for one-dimensional stencils n-1, n, n+1…... indicate the time steps where time step n and n-1 have known solutions and time step n+1 is to be calculated. The spatial location of finite volumes used in the calculation are indicated by j-1, j and j+1…...

Graphical representations of node arrangements and their coefficients arose early in the study of PDEs. Authors continue to use varying terms for these such as "relaxation patterns", "operating instructions", "logenzes", or "point patterns”. [1][2] The term "stencil" was coined for such patterns to reflect the concept of laying out a stencil in the usual sense over a computational grid to reveal just the numbers needed at a particular step. [19]

See the following table for Padé Approximation stencils

Table 5.2: The Pade' approximation stencils and type of solution, and order of error term

Type of Padé	Stencils	Type of solution	Order of error
(1,1)		Implicit method	$O(\theta^3)$
(0,2)		Implicit method	$O(\theta^3)$
(2,0)		Explicit method	$O(\theta^3)$
(2,1)		Explicit method	$O(\theta^4)$
(1,2)		Explicit method	$O(\theta^4)$
(2,2)		Implicit method	$O(\theta^5)$

Chapter Six

Numerical Comparison and Examples

6.1 Forward-difference (Explicit) Method Algorithm

The following algorithm implements the forward-difference (Explicit) Method for solving parabolic partial differential equation using the Matlab software.

Algorithm 6.1

To approximate the solution to the parabolic partial differential equation

$$\frac{\partial U}{\partial t} + f(x,t) = \frac{\partial^2 U}{\partial x^2} \qquad 0 < x < L \quad , 0 < t < T$$

Subject to the boundary conditions

$$u(0,t) = u(L,t) = 0 \qquad 0 < t < T$$

And the initial conditions

$$u(x,0) = g(x) \quad 0 \le x \le L$$

Input:

Endpoints a and b, boundary and initial conditions, number of subintervals N, maximum time T, constant α and integers M.

Output:

Approximations $w_{i,j}$ to $u(x_i, t_j)$ for each $i = 0,1,\ldots\ldots\ldots\ldots,M-1$ *and* $j = 0,1,\ldots\ldots\ldots\ldots,N$.

Step 1:

Set $h = \dfrac{b-a}{N}$, $k = \dfrac{T-a}{M}$, $r = \alpha \dfrac{k}{h^2}$.

Step 2:

$i = 1,.............,N + 1$ set $w_{i,j} = g(i \quad h)$ (initial conditions).

Step 3:

$i = 1,.............,M$ set boundary conditions.

Step 4:

Set and solve a tridiagonal linear system.

Step 5:

$For\ i = 0, 1, \ldots, N$

Set

$x_i = a + i * h$

$err_i = abs(Exact - W_i)$

$Output\ (x_i$, approximate$_i$, $Exact_i$, error $_i)$

Plot and Stop (The process is complete)

6.2 Backward-Difference (implicit) Method Algorithm

The following algorithm implements the backward-difference (implicit) Method for solving parabolic partial differential equation using the Matlab software.

Algorithm 6.2

To approximate the solution to the parabolic partial differential equation

$$\frac{\partial U}{\partial t} + f(x,t) = \frac{\partial^2 U}{\partial x^2} \qquad 0 < x < L \quad , 0 < t < T$$

Subject to the boundary conditions

$u(0,t)=u(L,t)=0 \qquad ,0 < t < T$

And the initial conditions

$u(x,0)=g(x) \quad 0 \le x \le L$

Input:

Endpoints a and b, boundary and initial conditions, number of subintervals N, maximum time T, constant α and integers M.

Output:

Approximations $w_{i,j}$ to $u(x_i, t_j)$ for each $i = 0,1,\ldots\ldots\ldots\ldots, M-1$ *and* $j = 0,1,\ldots\ldots\ldots\ldots, N$.

Step 1:

Set $h = \dfrac{b-a}{N}$, $k = \dfrac{T-a}{M}$, $r = \alpha\dfrac{k}{h^2}$.

Step 2:

$i = 1,\ldots\ldots\ldots\ldots, N+1$ set $w_{i,j} = g(i \quad h)$ (initial conditions).

Step 3:

$i = 1,\ldots\ldots\ldots\ldots, M$ set boundary conditions.

Step 4:

Define the third diagonals $-\lambda, 1+2\lambda \; ,-\lambda$ for the matrix A.

<u>Step 5:</u>

Solve the system are generated by input A,B and u_j vector

u=u (2: n,1);

u=u (2: n,1);

z (:1) =B*u;

for i=1:m

Q (2: n, i) =inv (A)*z(:,i);

z (: i+1) =B*Q (2: n,i);

end

Step 6:

$For\ i = 0, 1, \ldots, N$

Set

$x_i = a + i * h$

$err_i = abs(Exact - W_i)$

$Output$ (x_i, approximate$_i$, $Exact_i$, error $_i$)

Plot and Stop (The process is complete)

6.3 Crank-Nicolson Method Algorithm

The following algorithm implements the Crank-Nicolson Method for solving parabolic partial differential equation using the Matlab software.

Algorithm 6.3

To approximate the solution to the parabolic partial differential equation

$$\frac{\partial U}{\partial t} + f(x,t) = \frac{\partial^2 U}{\partial x^2} \qquad 0 < x < L \quad , 0 < t < T$$

Subject to the boundary conditions

$$u(0,t) = u(L,t) = 0 \qquad , 0 < t < T$$

And the initial conditions

$$u(x,0) = g(x) \quad 0 \le x \le L$$

Input:

Endpoints a and b, boundary and initial conditions, number of subintervals N, maximum time T, constant α and integers M.

Output:

Approximations $w_{i,j}$ to $u(x_i, t_j)$ for each $i = 0,1,\ldots\ldots\ldots\ldots, M-1$ *and* $j = 0,1,\ldots\ldots\ldots\ldots, N$.

Step 1:

Set $h = \dfrac{b-a}{N}$, $k = \dfrac{T-a}{M}$, $r = \alpha \dfrac{k}{h^2}$.

Step 2:

$i = 1,\ldots\ldots\ldots\ldots, N+1$ set $w_{i,j} = g(i \quad h)$ (initial conditions).

Step 3:

$i = 1, \ldots\ldots\ldots\ldots, M$ set boundary conditions.

Step 4:

Define the third diagonals for the matrix A.

Define the third diagonals for the matrix B.

Step 5:

Solve the system are generated by input A,B and u_j vector

u=u (2: n,1);

u=u (2: n,1);

z (:1) =B*u;

for i=1:m

Q (2: n, i) =inv (A)*z(:,i);

z (: i+1) =B*Q (2: n,i);

end

Step 6:

$For\ i = 0, 1, \ldots, N$

Set

$x_i = a + i * h$

$err_i = abs(Exact - W_i)$

$Output$ (x_i, approximate$_i$, $Exact_i$, error $_i$)

Plot and Stop (The process is complete)

6.4 Method of Line Algorithm

The following algorithm implements the MOL for solving parabolic partial differential equation using the Matlab software.

Algorithm 6.4

To approximate the solution to the parabolic partial differential equation

$$\frac{\partial U}{\partial t} + f(x,t) = \frac{\partial^2 U}{\partial x^2} \qquad 0 < x < L \quad ,0 < t < T$$

Subject to the boundary conditions

$$u(0,t) = u(L,t) = 0 \qquad ,0 < t < T$$

And the initial conditions

$$u(x,0) = g(x) \quad 0 \le x \le L$$

Input:

Endpoints a and b, boundary and initial conditions, number of subintervals N and n, maximum time T, constant α and integers M.

Output:

Approximations $w_{i,j}$ to $u(x_i, t_j)$ for each $i = 0,1,............., M-1$ *and* $j = 0,1,............., N$.

Step 1:

Set $h = \frac{b-a}{N}$, $k = \frac{T-a}{M}$, $r = \alpha \frac{k}{h^2}$.

Step 2:

$i = 1,............., N+1$ set $w_{i,j} = g(i \quad h)$ (initial conditions).

Step 3:

$i = 1, \ldots\ldots\ldots\ldots, M$ set boundary conditions.

Step 4:

Define the matrix A=ut, and define the system of ODE's by use new m-file.

Step 5:

Use Runge–Kutta methods to solve new system of ODE's

$i = 1, \ldots\ldots\ldots\ldots, \mathrm{M}+1$

$$k_1 = k \times pde\_1(t, y(:, i))$$

$$k_2 = k \times pde\_1(t + \frac{k}{2}, y(:, i) + \frac{k_1}{2})$$

$$k_3 = k \times pde\_1(t + \frac{k}{2}, y(:, i) + \frac{k_2}{2})$$

$$k_4 = k \times pde\_1(t + k, y(:, i) + k_3)$$

$$y(:, i+1) = y(:, i) + \frac{(k_1 + 2k_2 + 2k_3 + k_2)}{6}$$

Step 6:

Compute exact solution and plot the solution

$For\ i = 0, 1, \ldots, N$

Set

$x_i = a + i * h$

$err_i = abs(Exact - W_i)$

$Output$ (x_i, approximate$_i$, $Exact_i$, error $_i$)

Plot and Stop (The process is complete)

6.5 Padé Approximation (2,2), (1,1), (0,2), (2,0), (1,2), (2,1)

The following algorithm implements the Padé approximation (2,2), (1,1), (0,2), (2,0), (1,2), (2,1) for solving parabolic partial differential equation using the Matlab software.

Algorithm 6.5

To approximate the solution to the parabolic partial differential equation

$$\frac{\partial U}{\partial t} + f(x,t) = \frac{\partial^2 U}{\partial x^2} \qquad 0 < x < L \quad , 0 < t < T$$

Subject to the boundary conditions

$$u(0,t) = u(L,t) = 0 \qquad , 0 < t < T$$

And the initial conditions

$$u(x,0) = g(x) \quad 0 \le x \le L$$

Input:

Endpoints a and b, boundary and initial conditions, number of subintervals N and n, maximum time T, constant α and integers M, and $r = \frac{k}{h^2}$.

<u>Output:</u>

Approximations $w_{i,j}$ to $u(x_i, t_j)$ for each $i = 0,1,\ldots\ldots\ldots, M-1$ *and* $j = 0,1,\ldots\ldots\ldots, N$.

Step 1:

Set $h = \frac{b-a}{N}$, $k = \frac{T-a}{M}$, $r = \alpha \frac{k}{h^2}$.

Step 2:

$i = 1,.............,N+1$ set $w_{i,j} = g(i \quad h)$ (initial conditions).

Step 3:

$i = 1,..............,M$ set boundary conditions.

Step 4:

Define the matrix A is a diagonal matrix Define the matrix B is a diagonal matrix

Define first and last element in matrix A and B

Step 5:

Solve the system are generated by input A,B and u_j vector

u=u (2: n,1);

u=u (2: n,1);

z (:1) =B*u;

for i=1:m

Q (2: n, i) =inv (A)*z(:,i);

z (: i+1) =B*Q (2: n,i);

end

Step 6:

Compute exact solution and plot the solution

$For\ i = 0, 1, \ldots, N$

Set

$x_i = a + i * h$

$err_i = abs(Exact - W_i)$

$Output\ (x_i$, approximate$_i$, $Exact_i$, error $_i)$

Plot and Stop (The process is complete)

6.5 Numerical Examples and Results

To test the efficiency and effectiveness of the numerical methods that have been developed and studied in previous chapters, we will test the following examples:

6.6.1. Example 1

Consider the following Parabolic partial differential equation:

$$\frac{\partial\ U}{\partial\ t} - \frac{\partial^2 U}{\partial x^2} = 0 \qquad 0 \prec x \prec 1 \qquad 0 \le t$$

With the following boundary conditions: $u(0,t) = u(1,t) = 0 \quad 0 \prec t$, and initial conditions $u(x,0) = \sin(\pi x) \quad 0 \le x \le 1$.

The exact solution is E(x) $= u(x,t) = e^{-\pi^2 \times 0.5} \times \sin(\pi x)$, the following tables represent the results that have been obtained after solving example 1 using the previous methods. . (When h=0.1, k=0,0005 and t=0.5)

Forward-Difference (Explicit) Method Algorithm Example 1

Using forward-difference method algorithm 6.1 for solving example 1, the following table represents the numerical and the exact results for N=10:

Table 6. 1: the exact and the approximated solutions for x_i where i =0...10

x_i	App. At j=1	App. At j=2
0.0	0.000000000000000	0.000000000000000
0.1	0. 309016994374947	0. 307504557552076
0.2	0. 587785252292473	0. 584908426500721
0.3	0. 809016994374947	0. 805057383366834
0.4	0. 951056516295154	0. 946401714384385
0.5	1. 000000000000000	0. 995105651629515
0.6	0. 951056516295154	0. 946401714384386
0.7	0. 809016994374947	0. 805057383366834
0.8	0. 587785252292473	0. 584908426500721
0.9	0. 309016994374948	0. 307504557552076
1.0	0.000000000000000	0.000000000000000

And when last level:

x_i	App. At last level	$Exact_i$	Err_i
0.0	0.000000000000000	0.000000000000000	0.0
0.1	0. 002320425211800	0.002222414178513	$9.8 * 10^{-5}$
0.2	0. 004413711036516	0.004227282972762	$1.8 * 10^{-4}$
0.3	0. 006074952072844	0.005818355856426	$2.5 * 10^{-4}$
0.4	0. 007141534473603	0.006839887529993	$3.0 * 10^{-4}$
0.5	0. 007509053722089	0.007191883355826	$3.1 * 10^{-4}$
0.6	0. 007141534473603	0.006839887529993	$3.0 * 10^{-4}$
0.7	0. 006074952072844	0.005818355856426	$2.5 * 10^{-4}$
0.8	0. 004413711036516	0.004227282972762	$1.8 * 10^{-4}$
0.9	0. 002320425211800	0.002222414178513	$9.8 * 10^{-5}$
1.0	0.000000000000000	0.000000000000000	0.0000000

Maximum Error = $3.171 * 10^{-4}$

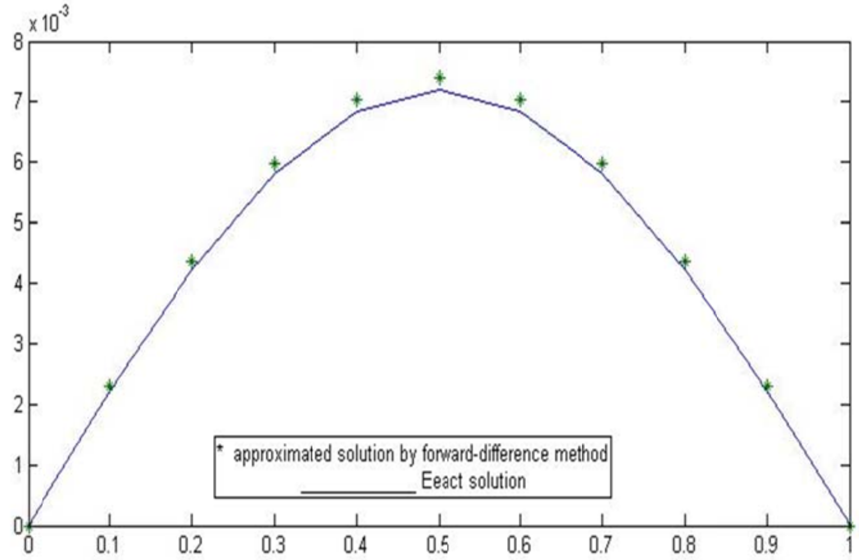


Figure 6. 1: The exact and the approximated solutions for example 1 using forward-difference (Explicit) Method

Backward-Difference (implicit) Method Algorithm Example 1

Using backward-difference method algorithm 6.2 for solving example 1, the following table represents the numerical and the exact results for $\mathrm{N} = 10$:

Table 6. 2: the exact and the approximated solutions for x_i where i =0...10

x_i	App. At j=1	App. At j=2
0.0	0.000000000000000	0.000000000000000
0.1	0. 309016994374947	0. 304523738605363
0.2	0. 587785252292473	0. 579238571934385
0.3	0. 809016994374947	0. 797253498050028
0.4	0. 951056516295154	0. 937227696984785
0.5	1. 000000000000000	0. 985459518889331
0.6	0. 951056516295154	0. 937227696984785
0.7	0. 809016994374947	0. 797253498050028
0.8	0. 587785252292473	0. 579238571934385
0.9	0. 309016994374948	0. 304523738605363
1.0	0.000000000000000	0.000000000000000

And when last level:

x_i	$Exact_i$	App. At last level	Err_i
0.0	0.000000000000000	0.000000000000000	0.0
0.1	0.002222414178513	0.002134474537761	$8.7 * 10^{-5}$
0.2	0.004227282972762	0.004060011836007	$1.6 * 10^{-4}$
0.3	0.005818355856426	0.005588126887980	$2.3 * 10^{-4}$
0.4	0.006839887529993	0.006569237145387	$2.7 * 10^{-4}$
0.5	0.007191883355826	0.006907304700437	$2.8 * 10^{-4}$
0.6	0.006839887529993	0.006569237145387	$2.7 * 10^{-4}$
0.7	0.005818355856426	0.005588126887980	$2.3 * 10^{-4}$
0.8	0.004227282972762	0.004060011836008	$1.6 * 10^{-4}$
0.9	0.002222414178513	0.002134474537761	$8.7 * 10^{-5}$
1.0	0.000000000000000	0.000000000000000	0.0000000

Maximum Error = $2.845 * 10^{-4}$

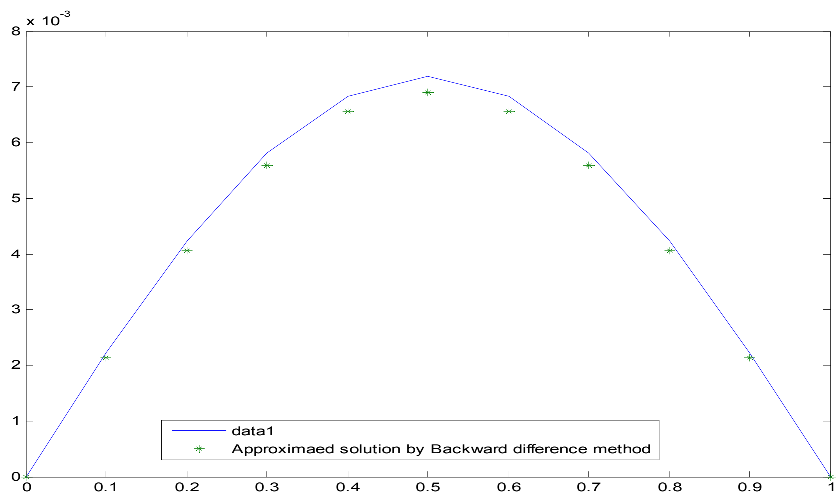


Figure 6. 2: The exact and the approximated solutions for example 1 using backward-difference (Implicit) Method

Crank-Nicolson Method Algorithm Example 1

Using Crank-Nicolson method algorithm 6.3 for solving example 1, the following table represents the numerical and the exact results for N =

10:

Table 6. 3: the exact and the approximated solutions for x_i where i =0...10

x_i	App. At j=1	App. At j=2
0.0	0.000000000000000	0.000000000000000
0.1	0. 280179657638192	0. 254033409110845
0.2	0. 532933378260296	0. 483200258183083
0.3	0. 733519866653096	0. 665068099330199
0.4	0. 862304319764463	0. 781834441112953
0.5	0. 906680418029808	0. 822069380438708
0.6	0. 862304319764463	0. 781834441112953
0.7	0. 733519866653096	0. 665068099330199
0.8	0. 532933378260296	0. 483200258183083
0.9	0. 280179657638192	0. 254033409110845
1.0	0.000000000000000	0.000000000000000

And when last level:

x_i	$Exact_i$	App. At last level	Err_i
0.0	0.000000000000000	0.000000000000000	0.0
0.1	0.002222414178513	0.002305123367789	$8.2 * 10^{-5}$
0.2	0.004227282972762	0.004384605199600	$1.5 * 10^{-4}$
0.3	0.005818355856426	0.006034891325133	$2.1 * 10^{-4}$
0.4	0.006839887529993	0.007094440240202	$2.5 * 10^{-4}$
0.5	0.007191883355826	0.007459535914688	$2.6 * 10^{-4}$
0.6	0.006839887529993	0.007094440240202	$2.5 * 10^{-4}$
0.7	0.005818355856426	0.006034891325133	$2.1 * 10^{-4}$
0.8	0.004227282972762	0.004384605199600	$1.5 * 10^{-4}$
0.9	0.002222414178513	0.002305123367789	$8.2 * 10^{-5}$
1.0	0.000000000000000	0.000000000000000	0.0000000

Maximum Error = $2.6 * 10^{-4}$

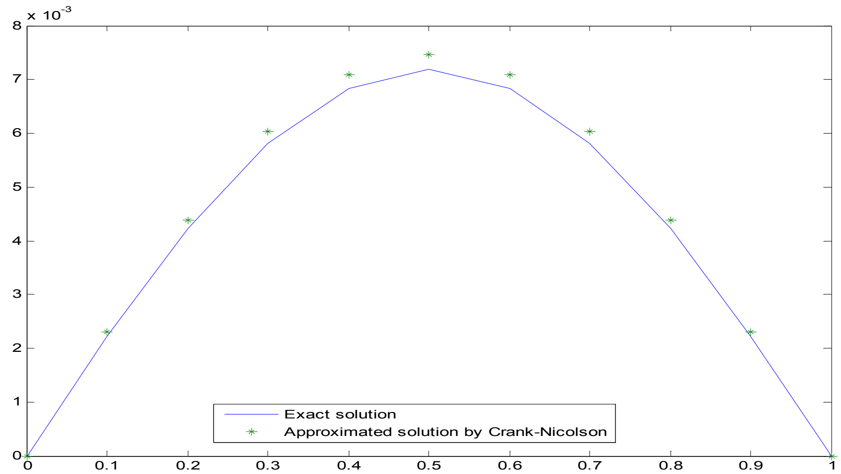


Figure 6. 3: The exact and the approximated solutions for example 1 using Crank-Nicolson Method

MOL Algorithm Example 1

Using MOL 6.4 for solving example 1, the following table represents the numerical and the exact results for N = 10:

Table 6.4: the exact and the approximated solutions for x_i where i =0...10

x_i	App. At j=1	App. At j=2
0.0	0.000000000000000	0.0000000000000000
0.1	0. 309016994374947	0. 307508252717500
0.2	0. 587785252292473	0. 584915455123030
0.3	0. 809016994374947	0. 805067057435506
0.4	0. 951056516295154	0. 946413086934176
0.5	1. 000000000000000	0. 995117609436013
0.6	0. 951056516295154	0. 946413086934176
0.7	0. 809016994374947	0. 805067057435506
0.8	0. 587785252292473	0. 584915455123030
0.9	0. 309016994374948	0. 307508252717500
1.0	0.0000000000000000	0.0000000000000000

And when last level:

x_i	$Exact_i$	App. At last level	Err_i
0.0	0.000000000000000	0.000000000000000	0.0
0.1	0.002222414178513	0.002314162620051	$9.1 * 10^{-5}$
0.2	0.004227282972762	0.004401798879132	$1.7 * 10^{-4}$
0.3	0.005818355856426	0.006058556394787	$2.4 * 10^{-4}$
0.4	0.006839887529993	0.007122260198076	$2.8 * 10^{-4}$
0.5	0.007191883355826	0.007488787549473	$2.9 * 10^{-4}$
0.6	0.006839887529993	0.007122260198076	$2.8 * 10^{-4}$
0.7	0.005818355856426	0.006058556394787	$2.4 * 10^{-4}$
0.8	0.004227282972762	0.004401798879132	$1.7 * 10^{-4}$
0.9	0.002222414178513	0.002314162620051	$9.1 * 10^{-5}$
1.0	0.000000000000000	0.000000000000000	0.0000000

Maximum Error = $2.969 * 10^{-4}$

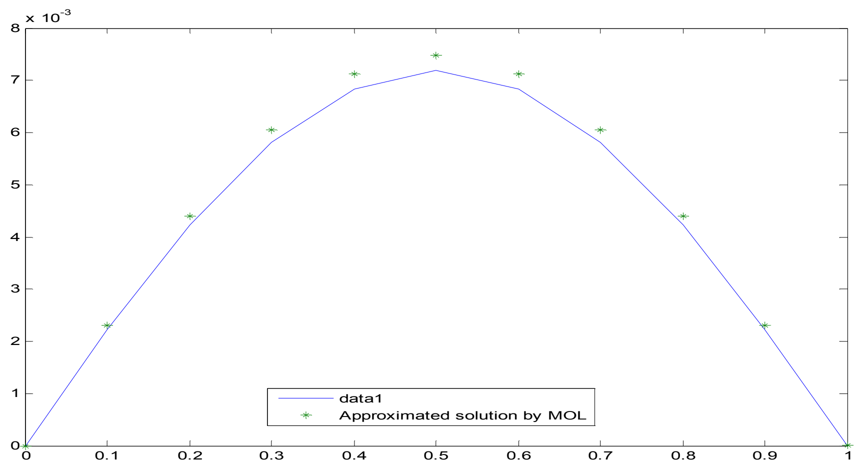


Figure 6. 4: The exact and the approximated solutions for example 1 using MOL

Padé Approximation (1,1) Algorithm Example 1

Using (1,1) Padé approximation 6.5 for solving example 1, the following table represents the numerical and the exact results for N = 10:

Table 6.5: the exact and the approximated solutions for x_i where i =0...10

x_i	App. At j=1	App. At j=2
0.0	0.000000000000000	0.0000000000000000
0.1	0. 307508249713065	0. 306006871346552
0.2	0. 584915449408255	0. 582059658050462
0.3	0. 805067049569794	0. 801136389976289
0.4	0. 946413077687477	0. 941792310205789
0.5	0. 995117599713458	0. 990259037259475
0.6	0. 946413077687477	0. 941792310205789
0.7	0. 805067049569794	0. 801136389976290
0.8	0. 584915449408256	0. 582059658050462
0.9	0. 307508249713065	0. 306006871346552
1.0	0.0000000000000000	0.0000000000000000

And when last level:

x_i	$Exact_i$	App. At last level	Err_i
0.0	0.000000000000000	0.000000000000000	0.0
0.1	0.002222414178513	0.002314140010197	$9.1 * 10^{-5}$
0.2	0.004227282972762	0.004401755872634	$1.7 * 10^{-4}$
0.3	0.005818355856426	0.006058497201422	$2.4 * 10^{-4}$
0.4	0.006839887529993	0.007122190612102	$2.8 * 10^{-4}$
0.5	0.007191883355826	0.007488714382449	$2.9 * 10^{-4}$
0.6	0.006839887529993	0.007122190612102	$2.8 * 10^{-4}$
0.7	0.005818355856426	0.006058497201422	$2.4 * 10^{-4}$
0.8	0.004227282972762	0.004401755872634	$1.7 * 10^{-4}$
0.9	0.002222414178513	0.002314140010197	$9.1 * 10^{-5}$
1.0	0.000000000000000	0.000000000000000	0.0000000

Maximum Error = $2.9683 * 10^{-4}$

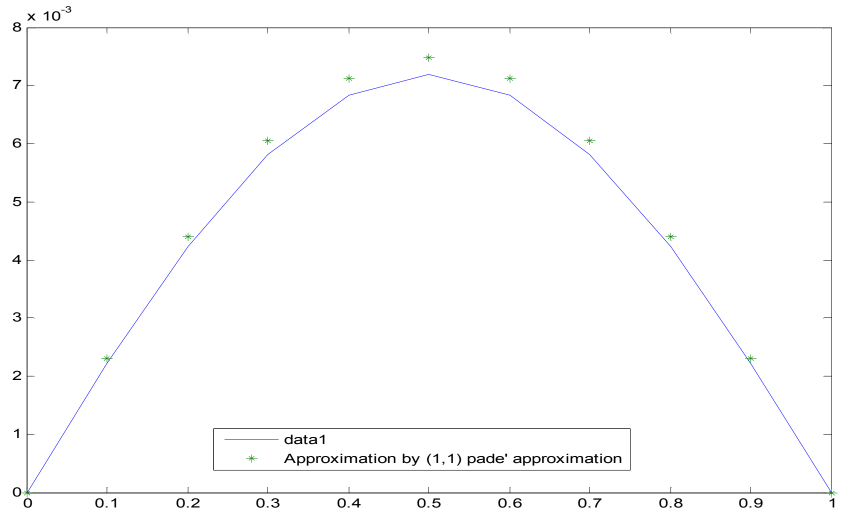


Figure 6. 5: The exact and the approximated solutions for example 1 using Pade' approximation (1,1)

Padé Approximation (0,2) Algorithm Example 1

Using (0,2) Padé approximation 6.5 for solving example 1, the following table represents the numerical and the exact results for N = 10:

Table 6.6: the exact and the approximated solutions for x_i where i =0...10

x_i	App. At j=1	App. At j=2
0.0	0.000000000000000	0.0000000000000000
0.1	0. 307508258704312	0. 306006889241249
0.2	0. 584915466510624	0. 582059692088198
0.3	0. 805067073109185	0. 801136436825214
0.4	0. 946413105359690	0. 941792365280002
0.5	0. 995117628809745	0. 990259095167930
0.6	0. 946413105359690	0. 941792365280002
0.7	0. 805067073109185	0. 801136436825214
0.8	0. 584915466510624	0. 582059692088198
0.9	0. 307508258704312	0. 306006889241249
1.0	0.0000000000000000	0.0000000000000000

And when last level:

x_i	$Exact_i$	App. At last level	Err_i
0.0	0.000000000000000	0.000000000000000	0.0
0.1	0.002222414178513	0.002314207674425	$9.1 * 10^{-5}$
0.2	0.004227282972762	0.004401884577645	$1.7 * 10^{-4}$
0.3	0.005818355856426	0.006058674348671	$2.4 * 10^{-4}$
0.4	0.006839887529993	0.007122398861183	$2.8 * 10^{-4}$
0.5	0.007191883355826	0.007488933348492	$2.9 * 10^{-4}$
0.6	0.006839887529993	0.007122398861183	$2.8 * 10^{-4}$
0.7	0.005818355856426	0.006058674348671	$2.4 * 10^{-4}$
0.8	0.004227282972762	0.004401884577645	$1.7 * 10^{-4}$
0.9	0.002222414178513	0.002314207674425	$9.1 * 10^{-5}$
1.0	0.000000000000000	0.000000000000000	0.0000000

Maximum Error = $2.9704 * 10^{-4}$

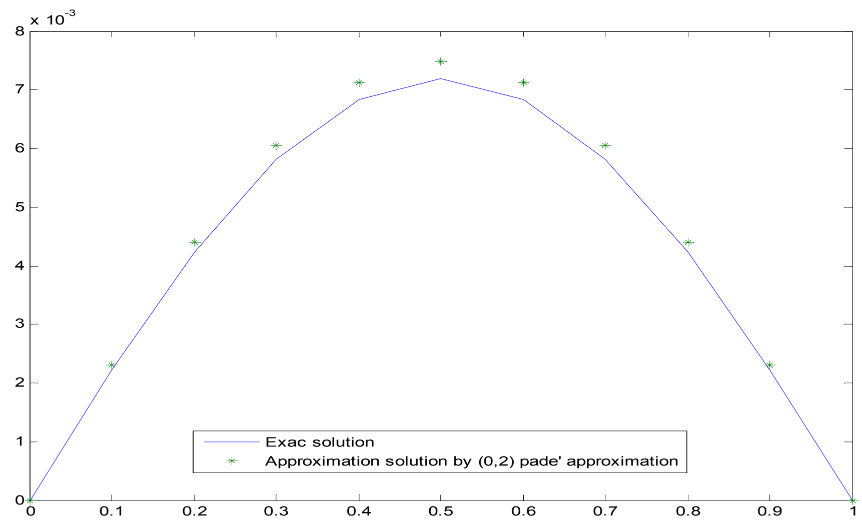


Figure 6.6: The exact and the approximated solutions for example 1 using Pade' approximation (0,2)

Padé Approximation (2,0) Algorithm Example 1

Using (2,0) Padé approximation 6.5 for solving example 1, the following table represents the numerical and the exact results for N = 10:

Table 6.7: the exact and the approximated solutions for x_i where i =0...10

x_i	App. At j=1	App. At j=2
0.0	0.000000000000000	0.0000000000000000
0.1	0. 307508258748426	0. 306006889329046
0.2	0. 584915466594534	0. 582059692255198
0.3	0. 805067073224677	0. 801136437055070
0.4	0. 946413105495459	0. 941792365550215
0.5	0. 995117628952501	0. 990259095452048
0.6	0. 946413105495459	0. 941792365550215
0.7	0. 805067073224677	0. 801136437055070
0.8	0. 584915466594534	0. 582059692255198
0.9	0. 307508258748426	0. 306006889329046
1.0	0.0000000000000000	0.0000000000000000

And when last level:

x_i	$Exact_i$	App. At last level	Err_i
0.0	0.000000000000000	0.000000000000000	0.0
0.1	0.002222414178513	0.002314208006413	$9.1 * 10^{-5}$
0.2	0.004227282972762	0.004401885209124	$1.7 * 10^{-4}$
0.3	0.005818355856426	0.006058675217827	$2.4 * 10^{-4}$
0.4	0.006839887529993	0.007122399882937	$2.8 * 10^{-4}$
0.5	0.007191883355826	0.007488934422828	$2.9 * 10^{-4}$
0.6	0.006839887529993	0.007122399882937	$2.8 * 10^{-4}$
0.7	0.005818355856426	0.006058675217827	$2.4 * 10^{-4}$
0.8	0.004227282972762	0.004401885209124	$1.7 * 10^{-4}$
0.9	0.002222414178513	0.002314208006413	$9.1 * 10^{-5}$
1.0	0.000000000000000	0.000000000000000	0.0000000

Maximum Error = $2.9705 * 10^{-4}$

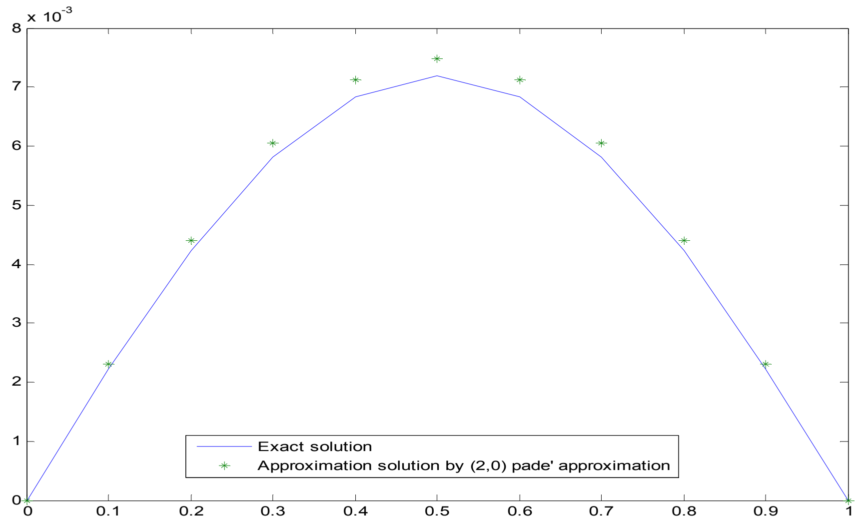


Figure 6. 7: The exact and the approximated solutions for example 1 using Pade' approximation (2,0)

Padé Approximation (2,1) Algorithm Example 1

Using (2,1) Padé approximation 6.5 for solving example 1, the following table represents the numerical and the exact results for N = 10:

Table 6.8: the exact and the approximated solutions for x_i where i =0...10

x_i	App. At j=1	App. At j=2
0.0	0.000000000000000	0.0000000000000000
0.1	0. 307508252719946	0. 306006877330953
0.2	0. 584915455127684	0. 582059669433470
0.3	0. 805067057441912	0. 801136405643656
0.4	0. 946413086941707	0. 941792328623883
0.5	0. 995117609443932	0. 990259056625405
0.6	0. 946413086941707	0. 941792328623883
0.7	0. 805067057441912	0. 801136405643656
0.8	0. 584915455127684	0. 582059669433470
0.9	0. 307508252719947	0. 306006877330954
1.0	0.0000000000000000	0.0000000000000000

And when last level:

x_i	$Exact_i$	App. At last level	Err_i
0.0	0.000000000000000	0.000000000000000	0.0
0.1	0.002222414178513	0.002314162638464	$9.1 * 10^{-5}$
0.2	0.004227282972762	0.004401798914156	$1.7 * 10^{-4}$
0.3	0.005818355856426	0.006058556442994	$2.4 * 10^{-4}$
0.4	0.006839887529993	0.007122260254747	$2.8 * 10^{-4}$
0.5	0.007191883355826	0.007488787609060	$2.9 * 10^{-4}$
0.6	0.006839887529993	0.007122260254747	$2.8 * 10^{-4}$
0.7	0.005818355856426	0.006058556442994	$2.4 * 10^{-4}$
0.8	0.004227282972762	0.004401798914156	$1.7 * 10^{-4}$
0.9	0.002222414178513	0.002314162638464	$9.1 * 10^{-5}$
1.0	0.000000000000000	0.000000000000000	0.0000000

Maximum Error = $2.9690 * 10^{-5}$

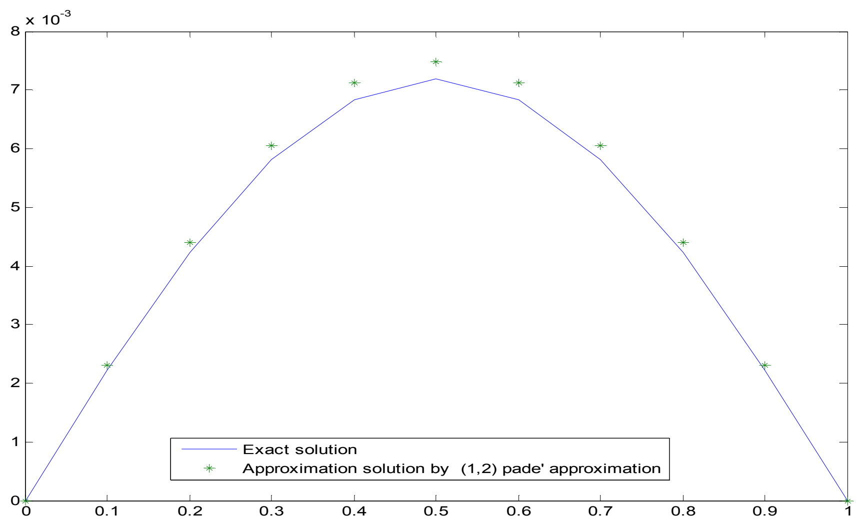


Figure6. 8: The exact and the approximated solutions for example 1 using Pade' approximation (2,1)

Padé Approximation (1,2) Algorithm Example 1

Using (1,2) Padé approximation 6.5 for solving example 1, the following table represents the numerical and the exact results for N = 10:

Table 6.9: the exact and the approximated solutions for x_i where i =0...10

x_i	App. At j=1	App. At j=2
0.0	0.000000000000000	0.0000000000000000
0.1	0. 298462966883305	0. 289062374610289
0.2	0. 584619148476617	0. 581094827031851
0.3	0. 805060879814834	0. 801101187367423
0.4	0. 946413000177270	0. 941791376075799
0.5	0. 995117608563431	0. 990259018194599
0.6	0. 946413000177270	0. 941791376075798
0.7	0. 805060879814833	0. 801101187367423
0.8	0. 584619148476617	0. 581094827031851
0.9	0. 298462966883305	0. 289062374610289
1.0	0.0000000000000000	0.0000000000000000

And when last level:

x_i	$Exact_i$	App. At last level	Err_i
0.0	0.000000000000000	0.000000000000000	0.0
0.1	0.002222414178513	0. 00069850781858	$1.5 * 10^{-3}$
0.2	0.004227282972762	0. 00174755626737	$2.4 * 10^{-3}$
0.3	0.005818355856426	0. 00259959775947	$3.2 * 10^{-3}$
0.4	0.006839887529993	0. 00315337099172	$3.6 * 10^{-3}$
0.5	0.007191883355826	0. 00334530339313	$3.8 * 10^{-3}$
0.6	0.006839887529993	0. 00315337099172	$3.6 * 10^{-3}$
0.7	0.005818355856426	0. 00259959775947	$3.2* 10^{-3}$
0.8	0.004227282972762	0. 00174755626737	$2.4 * 10^{-3}$
0.9	0.002222414178513	0. 00069850781858	$1.5 * 10^{-3}$
1.0	0.000000000000000	0.000000000000000	0.0000000

Maximum Error = $3.8465 * 10^{-3}$

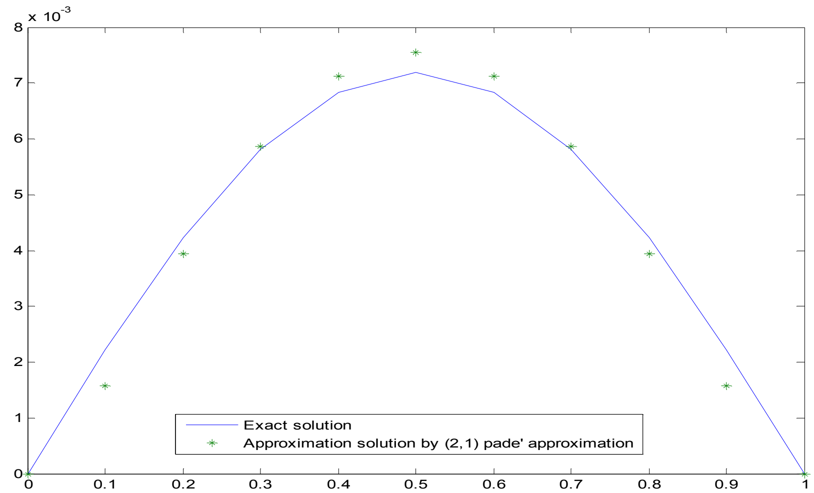


Figure6.9: The exact and the approximated solutions for example 1 using Pade' approximation (1,2)

Padé Approximation (2,2) Algorithm Example 1

Using (2,2) Padé approximation 6.5 for solving example 1, the following table represents the numerical and the exact results for N = 10:

Table 6.10: the exact and the approximated solutions for xi where i =0...10

x_i	App. At j=1	App. At j=2
0.0	0.000000000000000	0.0000000000000000
0.1	0. 278135629280731	0. 250340120580335
0.2	0. 529148977817020	0. 476362263415966
0.3	0. 728312930486849	0. 655659562923915
0.4	0. 856183225239916	0. 770774084133244
0.5	0. 900244318657566	0. 810439833280047
0.6	0. 856183225239916	0. 770774084133244
0.7	0. 728312930486849	0. 655659562923915
0.8	0. 529148977817020	0. 476362263415966
0.9	0. 278135629280731	0. 250340120580335
1.0	0.0000000000000000	0.0000000000000000

And when last level:

x_i	$Exact_i$	App. At last level	Err_i
0.0	0.000000000000000	0.000000000000000	0.0
0.1	0.002222414178513	0.002189564355091	$3.2 * 10^{-5}$
0.2	0.004227282972762	0.004209651554766	$1.7 * 10^{-5}$
0.3	0.005818355856426	0.005792891565915	$2.5 * 10^{-5}$
0.4	0.006839887529993	0.006809960653377	$2.9 * 10^{-5}$
0.5	0.007191883355826	0.007160416686187	$3.1 * 10^{-5}$
0.6	0.006839887529993	0.006809960653378	$2.9 * 10^{-5}$
0.7	0.005818355856426	0.005792891565915	$2.5 * 10^{-5}$
0.8	0.004227282972762	0.004209651554766	$1.7 * 10^{-5}$
0.9	0.002222414178513	0.002189564355091	$3.2 * 10^{-5}$
1.0	0.000000000000000	0.000000000000000	0.0000000

Maximum Error = $3.2 * 10^{-5}$

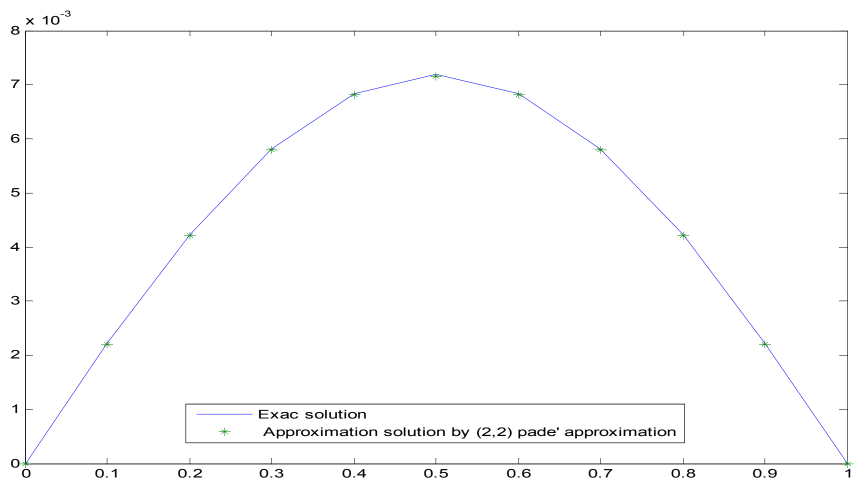


Figure7. 10: The exact and the approximated solutions for example 1 using Pade' approximation (2,2)

6.6.2. Example 2

Consider the following Parabolic partial differential equation:

$$\frac{\partial U}{\partial t} - \frac{1}{6}\frac{\partial^2 U}{\partial x^2} = 0 \qquad 0 < x < 1, 0 \leq t$$

With the following boundary conditions: $u(0,t) = u(1,t) = 0$, $0 < t$, and initial conditions $u(x,0) = 2\sin(2\pi x)$ $0 \le x \le 1$.

The exact solution is E(x) $= u(x,t) = 2e^{\frac{-\pi^2}{4} \times 0.5} \times \sin(2\pi x)$, the following tables represent the results that have been obtained after solving example 2 using the previous methods.(When h=0.1 , k=0,0005 and t=0.5)

Forward-Difference (Explicit) Method Algorithm Example 2

Using forward-difference method algorithm 6.1 for solving example 2, the following table represents the numerical and the exact results for N = 10:

Table 6. 15: the exact and the approximated solutions for x_i where i =0...10

x_i	App. At j=1	App. At j=2
0.0	0.000000000000000	0.0000000000000000
0.1	1. 175570504584946	1.1 174167292158135
0.2	1. 902113032590307	1. 899842587190290
0.3	1. 902113032590307	1. 899842587190290
0.4	1. 175570504584947	1. 174167292158135
0.5	0. 000000000000000	0. 000000000000000
0.6	-1. 175570504584946	-1. 174167292158135
0.7	-1. 902113032590307	-1. 899842587190290
0.8	-1. 902113032590307	-1. 899842587190291
0.9	-1. 175570504584947	-1. 174167292158136
1.0	0.0000000000000000	0.0000000000000000

And when last level:

x_i	$Exact_i$	App. At last level	Err_i
0.0	0.000000000000000	0.000000000000000	0.0000000
0.1	0.342341334840069	0. 358639691527622	$1.6298 * 10^{-2}$
0.2	0.553919915525240	0. 580291210606469	$2.6371 * 10^{-2}$
0.3	0.553919915525240	0. 580291210606469	$2.6371 * 10^{-2}$
0.4	0.342341334840069	0. 358639691527622	$1.6298 * 10^{-2}$
0.5	0.000000000000000	0. 000000000000000	0.0000000
0.6	0.342341334840069	-0. 358639691527622	$1.6298 * 10^{-2}$
0.7	-0.553919915525240	-0. 580291210606469	$2.6371 * 10^{-2}$
0.8	-0.553919915525240	-0. 580291210606469	$2.6371 * 10^{-2}$
0.9	-0.342341334840069	-0. 358639691527622	$1.6298 * 10^{-2}$
1.0	0.000000000000000	0.000000000000000	0.0000000

Maximum Error = $2.5070 * 10^{-2}$

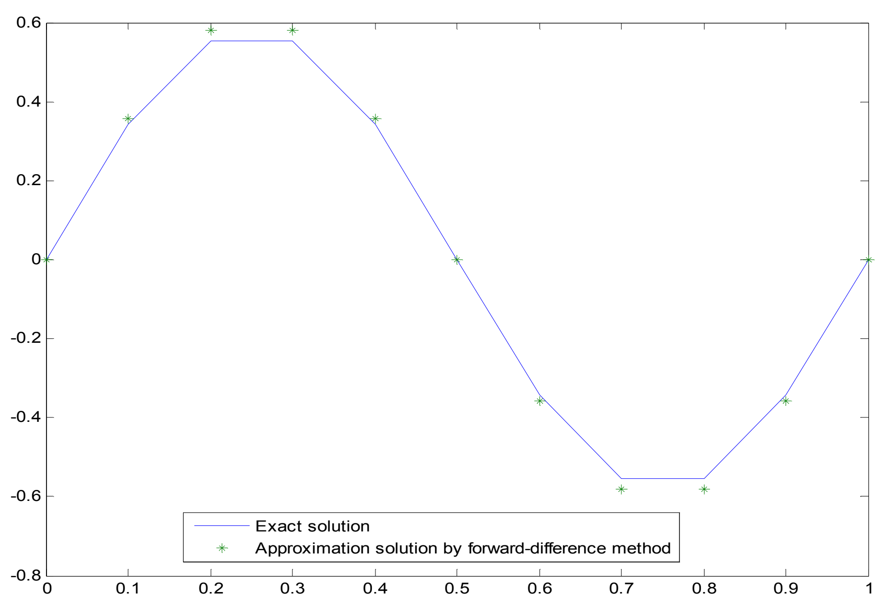


Figure 6. 15: The exact and the approximated solutions for example 1 using forward-difference (Explicit) Method

Backward-Difference (implicit) Method Algorithm Example 2

Using backward-difference method algorithm 6.2 for solving example 2, the following table represents the numerical and the exact results for $\mathrm{N} = 10$:

Table 6. 16: the exact and the approximated solutions for $\boldsymbol{xi}$ where $\boldsymbol{i}$ =0...10

x_i	App. At j=1	App. At j=2
0.0	0.000000000000000	0.0000000000000000
0.1	1. 174168965097034	1. 171370896962237
0.2	1. 899845294062291	1. 895317924717350
0.3	1. 899845294062291	1. 895317924717351
0.4	1. 174168965097035	1. 171370896962238
0.5	0. 000000000000000	0. 000000000000000
0.6	-1. 174168965097034	-1. 171370896962237
0.7	-1. 899845294062291	-1. 895317924717351
0.8	-1. 899845294062291	-1. 895317924717351
0.9	-1. 174168965097035	-1. 171370896962238
1.0	0.0000000000000000	0.0000000000000000

And when last level:

x_i	$Exact_i$	App. At last level	Err_i
0.0	0.000000000000000	0.000000000000000	0.0000000
0.1	0.342341334840069	0. 326847222671705	$1.5494 * 10^{-2}$
0.2	0.553919915525240	0. 528849915411329	$2.5070 * 10^{-2}$
0.3	0.553919915525240	0. 528849915411337	$2.5070 * 10^{-2}$
0.4	0.342341334840069	0. 326847222671717	$1.5494 * 10^{-2}$
0.5	0.000000000000000	0. 000000000000008	$8.0000 * 10^{-15}$
0.6	-0.342341334840069	-0. 326847222671734	$1.5494 * 10^{-2}$
0.7	-0.553919915525240	-0. 528849915411367	$2.5070 * 10^{-2}$
0.8	-0.553919915525240	-0. 528849915411368	$2.5070 * 10^{-2}$
0.9	-0.342341334840069	-0. 326847222671733	$1.5494 * 10^{-2}$
1.0	0.000000000000000	0.000000000000000	0.0000000

Maximum Error = $2.5070 * 10^{-2}$

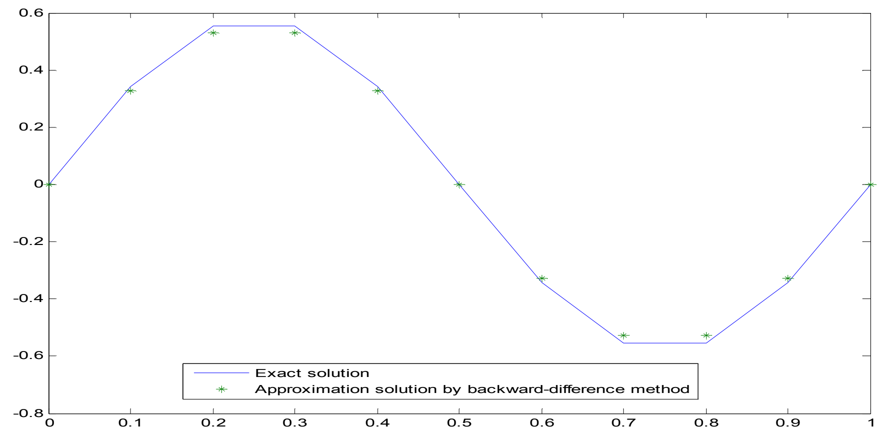


Figure 6. 16: The exact and the approximated solutions for example 2 using backward-difference (Implicit) Method

Crank-Nicolson Method Algorithm Example 2

Using Crank-Nicolson method algorithm 6.3 for solving example 2, the following table represents the numerical and the exact results for $N = 10$:

Table 6. 17: the exact and the approximated solutions for x_i where i =0...10

x_i	App. At j=1	App. At j=2
0.0	0.000000000000000	0.0000000000000000
0.1	1. 172767426606379	1. 174168129126511
0.2	1. 897577557147868	1. 899843941433569
0.3	1. 897577557147868	1. 899843941433569
0.4	1. 172767426606379	1. 174168129126511
0.5	0. 000000000000000	0. 000000000000000
0.6	-1. 172767426606378	-1. 174168129126510
0.7	-1. 897577557147868	-1. 899843941433569
0.8	-1. 897577557147868	-1. 899843941433569
0.9	-1. 172767426606379	-1. 174168129126511
1.0	0.0000000000000000	0.0000000000000000

And when last level:

x_i	$Exact_i$	App. At last level	Err_i
0.0	0.000000000000000	0.000000000000000	0.0000000
0.1	0.342341334840069	0. 356332725900775	$1.3991 * 10^{-2}$
0.2	0.553919915525240	0. 576558461811351	$2.2638 * 10^{-2}$
0.3	0.553919915525240	0. 576558461811354	$2.2638 * 10^{-2}$
0.4	0.342341334840069	0. 356332725900774	$1.3991 * 10^{-2}$
0.5	0.000000000000000	0. 000000000000002	$2.0000 * 10^{-15}$
0.6	-0.342341334840069	-0. 356332725900770	$1.3991 * 10^{-2}$
0.7	-0.553919915525240	-0. 576558461811347	$2.2638 * 10^{-2}$
0.8	-0.553919915525240	-0. 576558461811344	$2.2638 * 10^{-2}$
0.9	-0.342341334840069	-0. 356332725900769	$1.3991 * 10^{-2}$
1.0	0.000000000000000	0.000000000000000	0.0000000

Maximum Error = $2.2638 * 10^{-2}$

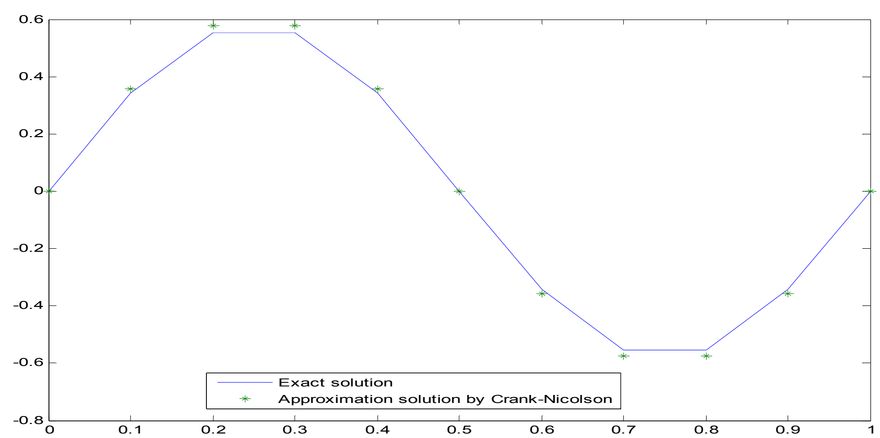


Figure 6. 17: The exact and the approximated solutions for example 2 using Crank-Nicolson Method

MOL Algorithm Example 2

Using MOL 7.4 for solving example 2, the following table represents the numerical and the exact results for N = 10:

Table 6.18: the exact and the approximated solutions for x_i where i =0...10

x_i	App. At j=1	App. At j=2
0.0	0.000000000000000	0.0000000000000000
0.1	1. 175570504584946	1. 153332139214378
0.2	1. 902113032590307	1. 866130601566489
0.3	1. 902113032590307	1. 866130601566490
0.4	1. 175570504584947	1. 153332139214379
0.5	0. 000000000000000	0. 000000000000000
0.6	-1. 175570504584946	-1. 153332139214378
0.7	-1. 902113032590307	-1. 866130601566489
0.8	-1. 902113032590307	-1. 866130601566490
0.9	-1. 175570504584947	-1. 153332139214379
1.0	0.0000000000000000	0.0000000000000000

And when last level:

x_i	$Exact_i$	App. At last level	Err_i
0.0	0.000000000000000	0.000000000000000	0.0000000
0.1	0.342341334840069	0. 346269581834599	$3.2183 * 10^{-3}$
0.2	0.553919915525240	0. 560275952678594	$6.2074 * 10^{-3}$
0.3	0.553919915525240	0. 560275952678594	$6.2638 * 10^{-3}$
0.4	0.342341334840069	0. 346269581834599	$3.3991 * 10^{-3}$
0.5	0.000000000000000	0. 000000000000000	$0.0000 * 10^{-3}$
0.6	-0.342341334840069	-0. 346269581834599	$3.3991 * 10^{-3}$
0.7	-0.553919915525240	-0. 560275952678594	$6.2638 * 10^{-3}$
0.8	-0.553919915525240	-0. 560275952678594	$6.2638 * 10^{-3}$
0.9	-0.342341334840069	-0. 346269581834599	$3.3991 * 10^{-3}$
1.0	0.000000000000000	0.000000000000000	0.0000000

Maximum Error = $6.2638 * 10^{-3}$

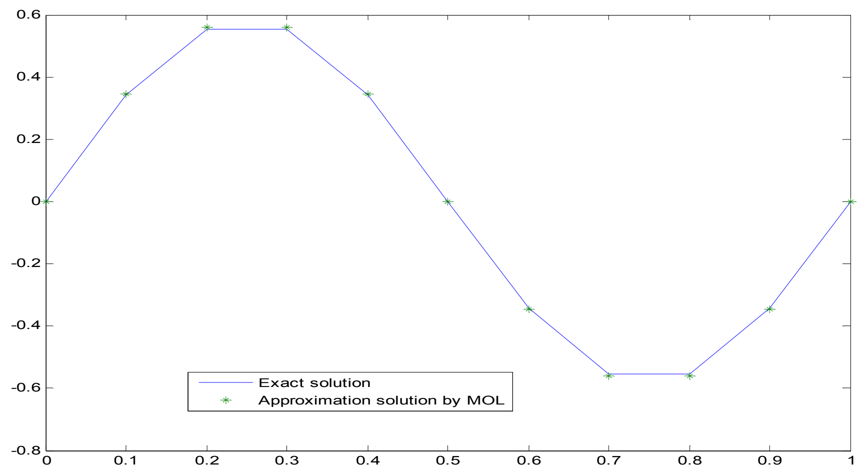


Figure 6. 18: The exact and the approximated solutions for example 2 using MOL

Padé approximation (1,1) Algorithm Example 2

Using (1,1) Padé approximation 6.5 for solving example 2, the following table represents the numerical and the exact results for N = 10:

Table 6.19: the exact and the approximated solutions for x_i where i =0...10

x_i	App. At j=1	App. At j=2
0.0	0.000000000000000	0.0000000000000000
0.1	1. 174168129126511	1. 172767426606379
0.2	1. 899843941433569	1. 897577557147868
0.3	1. 899843941433569	1. 897577557147868
0.4	1. 174168129126511	1. 172767426606379
0.5	0. 000000000000000	0. 000000000000000
0.6	-1. 174168129126510	-1. 172767426606378
0.7	-1. 899843941433569	-1. 897577557147868
0.8	-1. 899843941433569	-1. 897577557147868
0.9	-1. 174168129126511	-1. 172767426606379
1.0	0.0000000000000000	0.0000000000000000

And when last level:

x_i	$Exact_i$	App. At last level	Err_i
0.0	0.000000000000000	0.000000000000000	0.0000000
0.1	0.342341334840069	0. 356332725900775	$1.3991 * 10^{-2}$
0.2	0.553919915525240	0. 576558461811351	$2.2638* 10^{-2}$
0.3	0.553919915525240	0. 576558461811354	$2.2638 * 10^{-2}$
0.4	0.342341334840069	0. 356332725900774	$1.3991 * 10^{-2}$
0.5	0.000000000000000	0. 000000000000002	$2.0000* 10^{-15}$
0.6	-0.342341334840069	-0. 356332725900770	$1.3991 * 10^{-2}$
0.7	-0.553919915525240	-0. 576558461811347	$2.2638 * 10^{-2}$
0.8	-0.553919915525240	-0. 576558461811344	$2.2638 * 10^{-2}$
0.9	-0.342341334840069	-0. 356332725900769	$1.3991 * 10^{-2}$
1.0	0.000000000000000	0.000000000000000	0.0000000

Maximum Error = $2.2638 * 10^{-2}$

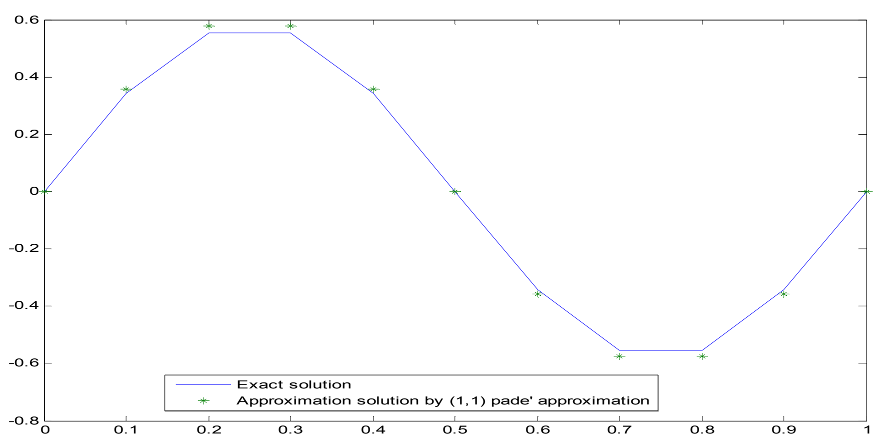


Figure 6.19: The exact and the approximated solutions for example 2 using Pade' approximation (1,1)

Padé Approximation (0,2) Algorithm Example 2

Using (0,2) Padé approximation 7.5 for solving example 2, the following table represents the numerical and the exact results for N = 10:

Table 6.20: the exact and the approximated solutions for x_i where i =0...10

x_i	App. At j=1	App. At j=2
0.0	0.000000000000000	0.0000000000000000
0.1	1. 174168129625435	1. 172767427603038
0.2	1. 899843942240846	1. 897577558760497
0.3	1. 899843942240847	1. 897577558760498
0.4	1. 174168129625436	1. 172767427603038
0.5	0. 000000000000000	0. 000000000000000
0.6	-1. 174168129625435	-1. 172767427603038
0.7	-1. 899843942240847	-1. 897577558760498
0.8	-1. 899843942240847	-1. 897577558760498
0.9	-1. 174168129625436	-1. 172767427603038
1.0	0.0000000000000000	0.0000000000000000

And when last level:

x_i	$Exact_i$	App. At last level	Err_i
0.0	0.000000000000000	0.000000000000000	0.0000000
0.1	0.342341334840069	0. 356332877312932	$1.3991 * 10^{-2}$
0.2	0.553919915525240	0. 576558706801392	$2.2638 * 10^{-2}$
0.3	0.553919915525240	0. 576558706801406	$2.2638 * 10^{-2}$
0.4	0.342341334840069	0. 356332877312956	$1.3991 * 10^{-2}$
0.5	0.000000000000000	0. 000000000000004	$4.0000 * 10^{-15}$
0.6	-0.342341334840069	-0. 356332877312948	$1.3991 * 10^{-2}$
0.7	-0.553919915525240	-0. 576558706801400	$2.2638 * 10^{-2}$
0.8	-0.553919915525240	-0. 576558706801402	$2.2638 * 10^{-2}$
0.9	-0.342341334840069	-0. 356332877312947	$1.3991 * 10^{-2}$
1.0	0.000000000000000	0.000000000000000	0.0000000

Maximum Error = $2.26387 * 10^{-2}$

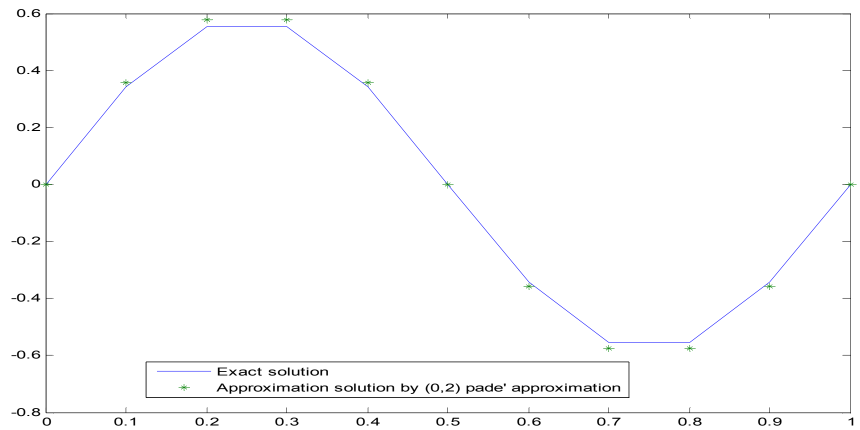


Figure6.20: The exact and the approximated solutions for example 2 using Pade' approximation (0,2)

Padé Approximation (2,0) Algorithm Example 2

Using (2,0) Padé approximation 6.5 for solving example 2, the following table represents the numerical and the exact results for N = 10:

Table 6.21: the exact and the approximated solutions for x_i where i =0...10

x_i	App. At j=1	App. At j=2
0.0	0.000000000000000	0.0000000000000000
0.1	1. 174168129626031	1. 172767427604229
0.2	1. 899843942241811	1. 897577558762424
0.3	1. 899843942241811	1. 897577558762424
0.4	1. 174168129626032	1. 172767427604229
0.5	0. 000000000000000	0. 000000000000000
0.6	-1. 174168129626031	-1. 172767427604228
0.7	-1. 899843942241811	-1. 897577558762424
0.8	-1. 899843942241811	-1. 897577558762424
0.9	-1. 174168129626032	-1. 172767427604229
1.0	0.0000000000000000	0.0000000000000000

And when last level:

x_i	$Exact_i$	App. At last level	Err_i
0.0	0.000000000000000	0.000000000000000	0.0000000
0.1	0.342341334840069	0. 356332877493831	$1.3991 * 10^{-2}$
0.2	0.553919915525240	0. 576558707094071	$2.2638* 10^{-2}$
0.3	0.553919915525240	0. 576558707094070	$2.2638 * 10^{-2}$
0.4	0.342341334840069	0. 356332877493830	$1.3991 * 10^{-2}$
0.5	0.000000000000000	0. 000000000000004	$0.0000* 10^{-15}$
0.6	-0.342341334840069	-0. 356332877493830	$1.3991 * 10^{-2}$
0.7	-0.553919915525240	-0. 576558707094071	$2.2638 * 10^{-2}$
0.8	-0.553919915525240	-0. 576558707094070	$2.2638 * 10^{-2}$
0.9	-0.342341334840069	-0. 356332877493831	$1.3991 * 10^{-2}$
1.0	0.000000000000000	0.000000000000000	0.0000000

Maximum Error = $2.26387 * 10^{-2}$

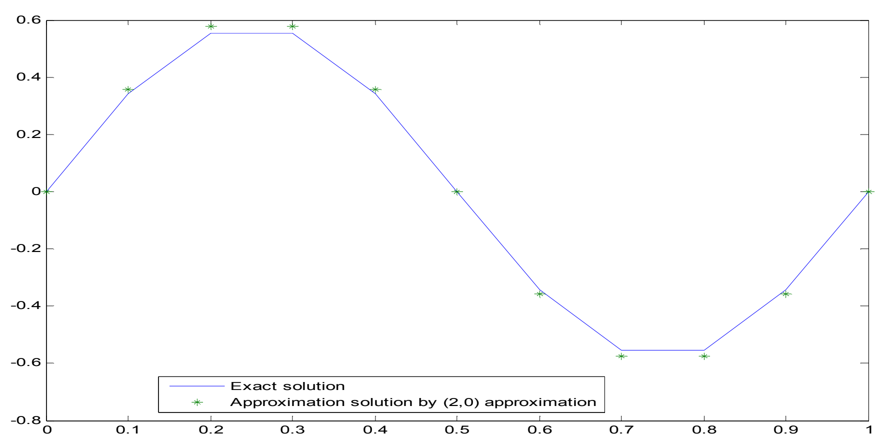


Figure6. 21: The exact and the approximated solutions for example 2 using Pade' approximation (2,0)

Padé Approximation (2,1) Algorithm Example 2

Using (2,1) Padé approximation 7.5 for solving example 2, the following table represents the numerical and the exact results for N = 10:

Table 6.22: the exact and the approximated solutions for x_i where i =0...10

x_i	App. At j=1	App. At j=2
0.0	0.000000000000000	0.0000000000000000
0.1	1. 174168129292951	1. 172767426938864
0.2	1. 899843941702875	1. 897577557685839
0.3	1. 899843941702876	1. 897577557685839
0.4	1. 174168129292952	1. 172767426938863
0.5	0. 000000000000000	0. 000000000000000
0.6	-1. 174168129292951	-1. 172767426938863
0.7	-1. 899843941702876	-1. 897577557685839
0.8	-1. 899843941702876	-1. 897577557685839
0.9	-1. 174168129292952	-1. 172767426938864
1.0	0.0000000000000000	0.0000000000000000

And when last level:

x_i	$Exact_i$	App. At last level	Err_i
0.0	0.000000000000000	0.000000000000000	0.0000000
0.1	0.342341334840069	0. 356332776411663	$1.3991 * 10^{-2}$
0.2	0.553919915525240	0. 576558543539681	$2.2638 * 10^{-2}$
0.3	0.553919915525240	0. 576558543539691	$2.2638 * 10^{-2}$
0.4	0.342341334840069	0. 356332776411667	$1.3991 * 10^{-2}$
0.5	0.000000000000000	-0. 000000000000005	$5.0000 * 10^{-15}$
0.6	-0.342341334840069	-0. 356332776411679	$1.3991 * 10^{-2}$
0.7	-0.553919915525240	-0. 576558543539710	$2.2638 * 10^{-2}$
0.8	-0.553919915525240	-0. 576558543539711	$2.2638 * 10^{-2}$
0.9	-0.342341334840069	-0. 356332776411678	$1.3991 * 10^{-2}$
1.0	0.000000000000000	0.000000000000000	0.0000000

Maximum Error = $2.26386 * 10^{-2}$

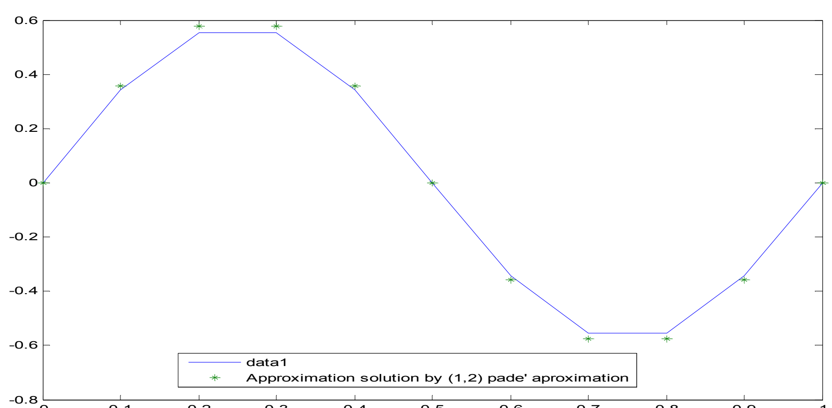


Figure6.22: The exact and the approximated solutions for example 2 using Pade' approximation (2,1)

Padé Approximation (1,2) Algorithm Example 2

Using (1,2) Padé approximation 7.5 for solving example 2, the following table represents the numerical and the exact results for N = 10:

Table 6.23: the exact and the approximated solutions for x_i where i =0...10

x_i	App. At j=1	App. At j=2
0.0	0.000000000000000	0.0000000000000000
0.1	1. 171738722624411	1. 167931641224419
0.2	1. 899838885724617	1. 897559949021956
0.3	1. 899843935118166	1. 897577517088887
0.4	1. 174168129287377	1. 172767426870479
0.5	0. 000000000000000	0. 000000000000000
0.6	-1. 174168129287376	-1. 172767426870479
0.7	-1. 899843935118166	-1. 897577517088887
0.8	-1. 899838885724617	-1. 897559949021956
0.9	-1. 171738722624411	-1. 167931641224419
1.0	0.0000000000000000	0.0000000000000000

And when last level:

x_i	$Exact_i$	App. At last level	Er_i
0.0	0.000000000000000	0.000000000000000	0.0000000
0.1	0.342341334840069	0. 242553599432976	$9.9787 * 10^{-2}$
0.2	0.553919915525240	0. 535197204021888	$1.8722 * 10^{-2}$
0.3	0.553919915525240	0. 585936075298137	$3.2016 * 10^{-2}$
0.4	0.342341334840069	0. 375645096825699	$3.3303 * 10^{-2}$
0.5	0.000000000000000	0. 000000000000004	$4.0000 * 10^{-15}$
0.6	-0.342341334840069	-0. 375645096825690	$1.3303 * 10^{-2}$
0.7	-0.553919915525240	-0. 585936075298124	$3.2016 * 10^{-2}$
0.8	-0.553919915525240	-0. 535197204021870	$3.8722 * 10^{-2}$
0.9	-0.342341334840069	-0. 242553599432973	$1.9787 * 10^{-2}$
1.0	0.000000000000000	0.000000000000000	0.0000000

Maximum Error = $9.9787 * 10^{-2}$

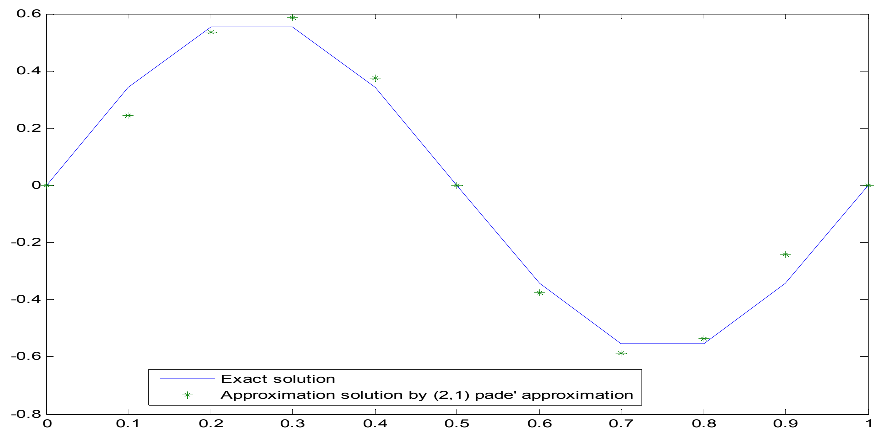


Figure 6. 23: The exact and the approximated solutions for example 2 using Pade' approximation (1,2)

Padé Approximation (2,2) Algorithm Example 2

Using (2,2) Padé approximation 7.5 for solving example 2, the following table represents the numerical and the exact results for N = 10:

Table 6.24: the exact and the approximated solutions for x_i where i =0...10

x_i	App. At j=1	App. At j=2
0.0	0.000000000000000	0.0000000000000000
0.1	1. 168226624483893	1. 160928622156394
0.2	1. 890231916977007	1. 878425014054166
0.3	1. 890231918454771	1. 878425016982061
0.4	1. 168227572225919	1. 878425016982061
0.5	0. 000000000000000	0. 000000000000000
0.6	-1. 168227572225918	-1. 160930505814887
0.7	-1. 890231918454771	-1. 878425016982061
0.8	-1. 890231916977007	-1. 878425014054165
0.9	-1. 168226624483893	-1. 160928622156394
1.0	0.0000000000000000	0.0000000000000000

And when last level:

x_i	$Exact_i$	App. At last level	Err_i
0.0	0.000000000000000	0.000000000000000	0.0000000
0.1	0.342341334840069	0. 342055900787213	$2.9787 * 10^{-4}$
0.2	0.553919915525240	0. 553546638417656	$3.8722* 10^{-4}$
0.3	0.553919915525240	0. 553546671357841	$3.2016 * 10^{-4}$
0.4	0.342341334840069	0. 342110657249714	$2.3303 * 10^{-4}$
0.5	0.000000000000000	0. 000000000000000	$0.0000* 10^{-15}$
0.6	-0.342341334840069	-0. 342110657249713	$2.3303 * 10^{-4}$
0.7	-0.553919915525240	-0. 553546671357842	$3.2016 * 10^{-4}$
0.8	-0.553919915525240	-0. 553546638417643	$3.8722 * 10^{-4}$
0.9	-0.342341334840069	-0. 342055900787212	$2.9787 * 10^{-4}$
1.0	0.000000000000000	0.000000000000000	0.0000000

Maximum Error = $3.8722 * 10^{-4}$

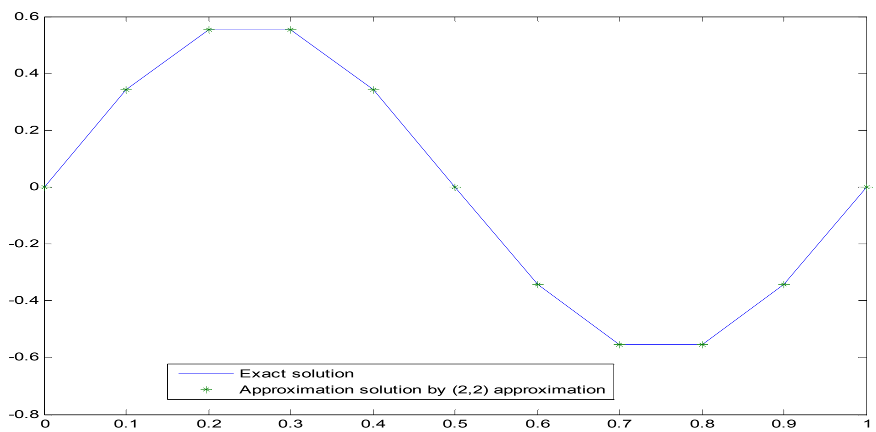


Figure6.24: The exact and the approximated solutions for example 2 using Pade' approximation (2,2)

Chapter Seven

Application Diffusion In Ceramic

7.1. Introduction

The required steps to assemble ceramic or metal particles into sophisticated form at a rapid rate with high-dimensional accuracy and low scrap rates has been a distinguished feature of materials processing research during the last decade. [23]

Here, we want to model the diffusion of organic vehicle of a ceramic body within reach of a high heating temperature. The model should predict the critical heating rate and reach to the boiling state of the actual body. [22]

In a technical matter, we would like to prevent the boiling state, because this will lead to defects (bubbles) in the material. [25]
Production of clay would be a relevant example, it involves oven heating of row products, namely mud and water. This leads to expulsion of water leaving the hard clay substance. A similar principle is employed for the production of ceramic item. The raw material, a mixture of ceramics and polymers, is heated to be left with just the ceramic as a result. The problem however is the occurrence of porosity that can occur in the end product due to the dissipation of monomer inside the ceramic body. [23]

The organic vehicles is acting similar of the polymers role in our previous example, which are decomposed by heating in low molecular called monomers. This monomer diffuses to the surface and evaporate. [22]

Ceramics is the most cost effective and widely used material. With an excellent combination of properties and attractive price has a wide range of application. It is available in purity ranges of 94–99.8 % and usable for critical high temperature application. it exhibits strong ionic interatomic bonding giving rise for its excellent properties such as solidity, ebullition and its wear resistance. The chemical inertness is of particular interest at high temperature. [22]

7.2. Diffusion Theory

Diffusion is the process by which matter is transported from one part of a system to another as a result of a random molecular montions. [23]

Diffusion and also heat execution are described with parabolic partial differential equations. In the simplest case without any heat loss or heat source, we get for the heat transport problem, which is similar to diffusion problem without any sink or source. [25]

$$u_t - \propto^2 u_{xx} = 0$$

Where $\propto^2$ is a material constant.

Such equations could be solved analytically with the Fourier method.

Four our problem, take an infinite cylinder of radius containing a ceramic polymer mixture, as the ceramic polymer mixture is heated up thermal degradation of the polymer produces monomers uniformly throughout the organic phase. [23]

Some monomer molecules stay inside the body while others evaporate from the surface and the resulting concentration of monomers stimulates the outward diffusion. Excessive heat will cause some monomer molecules to evaporate inside the ceramic body leaving minor holes inside the ceramic material, giving rise to internal defects. [22]

7.3. Diffusion Equation

The general diffusion equations for finite geometrical shape are following: [25]

I) In a rectangular parallelepiped the homogeneous diffusion equation is:

$$\frac{\partial d}{\partial t} = \frac{\partial}{\partial x}\left(M \times \frac{\partial d}{\partial x}\right) + \frac{\partial}{\partial y}\left(M \times \frac{\partial d}{\partial y}\right) + \frac{\partial}{\partial z}\left(M \times \frac{\partial d}{\partial z}\right) \qquad (7.3.1)$$

Where M may be a function of x, y, z and d

d is concentration of monomer a solution within the polymer phase based and unit is identified ($kilogram^{-3}$)

II) In a finite cylinder the diffusion equation is

$$\frac{\partial d}{\partial t} = \frac{1}{r}\left\{\frac{\partial}{\partial r}\left(rM \times \frac{\partial d}{\partial r}\right) + \frac{\partial}{\partial \emptyset}\left(\frac{M}{r} \times \frac{\partial d}{\partial \emptyset}\right) + \frac{\partial}{\partial z}\left(rM \times \frac{\partial d}{\partial z}\right)\right\} \qquad (7.3.2)$$

In terms of the cylindrical co-ordinates r ,$\emptyset$, z using the transformation

$$x = r\cos(\emptyset)$$

$$y = r\sin(\emptyset)$$

$$z = z$$

And

$$\frac{\partial u}{\partial \emptyset} = \frac{\partial u}{\partial x} \times \frac{\partial x}{\partial \emptyset}$$

In the equation (7.3.1) we obtain the formula (7.3.2)

III) In a sphere the diffusion equation is

$$\frac{\partial d}{\partial t} = \frac{1}{r^2}\left\{\frac{\partial}{\partial r}\left(r^2 M \frac{\partial d}{\partial r}\right)\right\} + \frac{1}{\sin(\theta)}\frac{\partial}{\partial \theta}\left(M\sin(\theta)\frac{\partial d}{\partial \theta}\right) + \frac{\partial}{\partial \emptyset}\left(\frac{M}{sin^2(\theta)}\frac{\partial d}{\partial \emptyset}\right) \quad (7.3.3)$$

In terms of spherical polar coordinates r , $\emptyset$, θ

Using again the transformation

$$x = r\ \sin(\cos(\emptyset))$$

$$y = rsin\ (\sin(\emptyset))$$

$$z = r\cos(\theta)$$

In the equation (7.3.1) we obtain the formula (7.3.3).

Where M may be a function of x, y, z and d

d is concentration of monomer in solution in the polymer phase based and unit is ($kilogram^{-3}$), and r radius (meter).

7.4. The Existing Model

In moulded ceramic bodies, the diffusion equation in an infinite plate, an infinite cylinder and a sphere is: [25]

$$\frac{\partial d}{\partial t} = \frac{1}{r^b}\left\{\frac{\partial}{\partial r}\left(r^b M \times \frac{\partial d}{\partial r}\right)\right\} + W \qquad (7.4.1)$$

Where $b = 0$ in the plate , 2r is the thickness of the plate.

Where $b = 1$ in the cylinder , r is the radius of the cylinder.

Where $b = 2$ in the sphere , r is the radius of the sphere.

And W is the rate of production of monomer based on total volume of suspension ($kilogram^{-3}/second^{-1}$) and given by :

$$W = P_p V_p K_0 \exp\left(-\frac{E}{RT}\right) exp\left\{\frac{-K_0 RT^2 \exp(-\frac{E}{RT})}{ZE} \times \left[1 - \frac{2RT}{E} + \frac{6(RT)^2}{E^2}\right]\right\}$$

(7.4.2)

Where V_p: is volume fraction of polymer in ceramic polymer body with condition ($0 < V_p < 1$).[25]

K_0 : specific rate constant for thermal degradation with unit ($second^{-1}$).

E: activation energy for thermal degradation with unit ($Jmol^{-1}$).

T: temperature

R: gas constant

Z: heating rate

The boundary conditions are

$\frac{\partial d}{\partial r} = 0$ at r = 0 and d=$d_s = 0$ at r =r_0 (7.4.3)

Where d_s is the surface concentration of monomer.

When the ceramic polymer suspension is heated, the polymer will produce monomers. These monomers are generated evenly throughout the organic phase. Some evaporate from the surface and resulting concentration of monomers stimulates the outward diffusion. [22]

A rapid gas flow posts the cylinder, sphere and plate ensures evaporation at the surface.

So we get a concentration difference between the surface and the interior of the body. This stimulates to the surface. [23]

This gas flow takes the monomers on the surface and remove these monomers from the cylinder. So we get a concentration difference between the surface and the interior of the body. This stimulates to the surface. [25] [24]

In the center of the infinite cylinder, the concentration of monomer is used to determine the variation of center vapour pressure of monomer with temperature. Once the concentration of monomer reaches its maximum value in the center, the vapour pressure in the center will cause an ambient pressure a bubble forms. [23]

Assuming that an infinite cylinder of radius injection moulded using the ceramic polymer suspension, is heated at a consistent rate z (k/s) and analysis of the kinetics of thermal degradation allows calculation of the mass fraction of polymer remaining (h) at given absolute temperature (T) using [25].

$$h = exp\left\{\frac{K_0 RT^2 \exp(-\frac{E}{RT})}{ZE}\left[1 - \frac{2RT}{E} + \frac{6(RT)^2}{E^2}\right]\right\} \qquad (7.4.4)$$

At the time t the concentration profile of the monomer, d = d (r, t), throughout the infinite cylinder, which has radius r, is determined by its rate of production, diffusion through the bulk and evaporation at the surface. Thus d satisfies the partial differential equation [25]

$$\frac{\partial d}{\partial t} = \frac{1}{r}\left\{\left(r^2 M \frac{\partial d}{\partial r}\right)\right\} + W; 0 < r < r_0, t > 0 \quad (7.4.5)$$

Subject to the initial distribution

D (r , 0) = 0 ; $0 \leq r \leq r_0$

And the boundary conditions

$$\frac{\partial d(0,t)}{\partial r} = 0\ ;\ t > 0$$

$$D(r_0, t) = D_s = 0$$

Where D_s: is concentration of monomer at the surface of the cylinder based on the total volume of suspension ($kilogram^{-3}$), r_0: is radius of cylinder (meter) and t : is time .

7.5. Diffusion Coefficient

The method of solution requires a knowledge of diffusion coefficient as a function of temperature and monomer construction

$$C \equiv C(t, c)$$

Where T=zt, T is temperature, z is the heating rate and t is the time.

From the free volume theory of Vrentas and Duda (1982) we get for continuous phase [24]

$$C_B = C_{01}(1-\emptyset)^2(1-2\varepsilon\emptyset)\exp\left\{\frac{-W_1V_1(0)+W_2\varphi V_2(0)}{V_f/\omega}\right\} \qquad (7.5.1)$$

Where C_B: diffusion coefficient of dispersed phase ($meter^2/second^{\ -1}$).

C_{01}: pre-exponential factor for diffusion.

W_1, W_2: the weight fraction of monomer in polymer monomer solution between 0 and 1.

V_f: average hole free volume per unit mass.

C_B denotes the diffusion coefficient and

$$C_{01} = C\exp\left(-\frac{E_D}{RT}\right) \ \ and \ \ \emptyset = \frac{W_1V_1(0)}{W_1V_1+W_2V_2} \qquad (7.5.2)$$

Subscripts 1 and 2 represent monomers and polymers respectively.

$\frac{V_f}{\omega}$: related to free volume parameters K_{11} and K_{12} which can be calculated using the Williams-Landel-Ferry constants c_1 and c_2 this gives :

$$\frac{V_f}{\omega} = \left(\frac{K_{11}}{\omega}\right)W_1[(c_2)_1 + T - (Tg)_1] + \left(\frac{K_{12}}{\omega}\right)W_2[(c_2)_2 + T - (Tg)_2]$$

(7.5.3)

And ε is an interaction parameter.

7.6. Shrinking Undegrated Core Model

If a polymer layer of thickness $r_0 - r_1$ has been removed from the infinite cylinder after time t, the volume fraction of polymer lost based on the volume of the body is :

$$1 - \left(\frac{r_1^2}{r_0^2}\right) \qquad (7.6.1)$$

Where r_1 is the radius of the cylinder containing polymer at time t. This volume fraction is equal to the weight fraction of polymer based on the total weight of polymer 1-h, from which we get [25]

$$r_1 = r_0\sqrt{h} \qquad (7.6.2)$$

Where h is given above.

7.7. Distributed Porosity Model

For the distributed porosity model, Maxwell's equation gives

$$\frac{C}{C_B} = 3V_A\left(\frac{C_A+2C_B}{C_A-C_B} - V_A\right)^{-1} + 1 \qquad (7.7.1)$$

Where C is the diffusion coefficient, V is the volume fraction and subscripts A, B refer to the dispersed and continuous phases respectively.

If $C_A = 0$ then the equation becomes [25]

$$\frac{C}{C_B} = 1 - \left(\frac{3V_c}{V_c+2} - V_A\right) \qquad (7.7.2)$$

If one of the dispersed phases has a zero transport coefficient, both expression reduce to [25]

$$C = \frac{C_pV_p + \frac{3C_uV_uC_p}{2C_p+C}}{V_p + \frac{3V_c}{2} + \frac{3V_uC_p}{2C_p+C_u}} \qquad (7.7.3)$$

Where C_u is the diffusion coefficient in the pore. V_u is the volume fraction of porosity and it is obtained from

$$V_u = (1 - V_c)(1 - h) \qquad (7.7.4)$$

Where V_c is the ceramic volume fraction

7.8. Vapour pressure

The calculation of vapour pressure of monomer in ceramic body is necessary because at the end, we have to check whether vapour pressure is greater than ambient pressure for boiling. [22]

The activity of the monomer in polymer α is given by the Flory-Higgins equation.

$$\alpha = \theta_1 \exp(\theta_2 + \varepsilon\, \theta_2^2) \qquad (7.8.1)$$

In which θ_1 and θ_2 are volume fraction of monomer and polymer respectively based on the total polymer. ε is the interaction parameter.

The vapour pressure of monomer over the polymer monomer solution at each time step is $\alpha\, P_1^0$, where P_1^0 is the vapour pressure of monomer over pure polymer. We can estimate P_1^0 using the Clausius-Clapeyron equation [25]

$$\ln P_1^0 = -\frac{\Delta H_{vap}}{RT} + i \qquad (7.8.2)$$

When the vapour pressure of monomer in the polymer exceeds ambient pressure, the nucleation of a bubble in the ceramic suspension, is possible.

7.9. Numerical Solution

We need to keep in mind, to solve our problem with the Crank-Nicolson method. Consider the equation [11]

$$\frac{-\lambda}{2} U_{i-1,j+1} + (1+\lambda) U_{i,j+1} - \frac{\lambda}{2} U_{i+1,j+1} = \frac{\lambda}{2} U_{i-1,j} + (1-\lambda) U_{i,j} + \frac{\lambda}{2} U_{i+1,j}$$

For the unsteady state situation, the appropriate finite difference form is obtained as shown below (Crank and Henry 1949)

$$\left[r\,M\frac{\partial d}{\partial r}\right]_{i+\frac{1}{2}}^{j} \approx \left(r_{i+\frac{1}{2}}\,M_{i+\frac{1}{2}} \times \frac{d_{m+1}-d_m}{h}\right)$$

$$\left[r\,M\frac{\partial d}{\partial r}\right]_{i-\frac{1}{2}}^{j} \approx \left(r_{i-\frac{1}{2}}\,M_{i-\frac{1}{2}} \times \frac{d_{m}-d_{m-1}}{h}\right)$$

And

$$\left(\frac{1}{r}\frac{\partial}{\partial r}\left(rM\frac{\partial d}{\partial r}\right)\right)_i^j \approx \frac{1}{r_i}\left(\frac{r_{i+\frac{1}{2}}\,M_{i+\frac{1}{2}}(d_{i+1}-d_i)-r_{i-\frac{1}{2}}\,M_{i-\frac{1}{2}}(d_i-d_{i-1})}{h^2}\right)$$

Now adopting the Crank-Nicolson method, we get

$$\frac{(d_{i,j+1}-d_{i,j})}{k} =$$

$$\frac{1}{2r_ih^2}\left\{r_{i+\frac{1}{2}} + d_{i+\frac{1}{2}}\left[\left(d_{i+1,j+1}+d_{i+1,j}\right)-\left(d_{i,j+1}+d_{i,j}\right)\right]\right\}$$

(7.9.1)

$$r_{i-\frac{1}{2}}\,M_{i-\frac{1}{2}}\left[\left(d_{i,j+1}+d_{i,j}\right)-\left(d_{i-1,j+1}+d_{i-1,j}\right)\right]+\frac{1}{2}\left[f(n+1)+f(n)\right]$$

Where $d_{i+\frac{1}{2}}$ and $d_{i-\frac{1}{2}}$ are approximated by the mean values of d at time level j and are taken to be the values of d for the concentrations given by

$$\frac{1}{2}\left(d_{i+1,j}+d_{i,j}\right)$$

And

$$\frac{1}{2}\left(d_{i,j}+d_{i-1,j}\right)$$

Respectively.

Rearranging equation (7.9.1) to give d_{n+1} terms in ascending order of m on the left hand side and letting [25]

$$\lambda = \frac{k}{h^2}$$

$$r_{i-1/2} = r_{i1}$$

$$r_{i+1/2} = r_{i2}$$

$$d_{i-1/2} = d(i-1)$$

$$d_{i+1/2} = d(i)$$

$$\frac{-\lambda(i-0.5)d(i-1)}{2i} M_{i-1,j+1} + \left(1 + \frac{\lambda(i+0.5)d(i)}{2i} + \frac{\lambda(i-0.5)d(i-1)}{2i}\right) M_{i,j+1} - \left(\frac{\lambda(i+0.5)d(i)}{2i}\right) M_{i+1,j+1} = \left(\frac{\lambda(i-0.5)d(i-1)}{2i}\right) M_{i-1,j} + \left(1 - \frac{\lambda(i+0.5)d(i)}{2i} - \frac{\lambda(i-0.5)d(i-1)}{2i}\right) M_{i,j} + \frac{\lambda(i+0.5)d(i)}{2i} M_{i+1,j} + \frac{k}{2}[f(n+1) + f(n)]$$

The finite difference representation of this equation at r = 0 has to be dealt with separately since [25]

$\frac{\partial d}{\partial r} = 0$ at r = 0

A Maclaurin series expansion is performed on $\frac{1}{r}\frac{\partial d}{\partial r}$ to obtain its limiting value , as $r \to 0$ which is equal to $\frac{\partial\ ^2 d}{\partial r^2}$ [11]. Therefore

$$\left(\frac{1}{r}\right)\frac{\partial}{\partial r}\left(rM\frac{\partial d}{\partial r}\right) \approx 2\,M\frac{\partial\ ^2 d}{\partial r^2}$$

And then

$$\frac{\partial d}{\partial r} = 2M\frac{\partial\ ^2 d}{\partial r^2} + error$$

Using finite differences

$$2M\frac{\partial\ ^2 d}{\partial r^2} = 2M_{0,j}\frac{(d_{1,j} - 2d_{0,j} + d_{-1,j})}{h^2}$$

From symmetry (cylinder), since ($d_{-1,j} = d_{1,j}$)

$$2M\frac{\partial\ ^2 d}{\partial r^2} = 4M_{0,j}\frac{(d_{1,j} - 2d_{0,j})}{h^2}$$

And then using the Crank-Nicolson and using M at time level n, we obtain

$$\frac{dd}{dt} = 2M_{0,n}\frac{\left(d_{1,j+1} - d_{0,j+1}\right)}{h^2} + 2M_{0,n}\frac{\left(d_{1,j} - d_{0,j}\right)}{h^2} + error$$

Rearranging to give n+1 terms in ascending the (L.H.S) gives

$$\left(1+2\lambda M_{0,n}\right)d_{0,j+1}-2\lambda M_{0,n}d_{1,j+1}=\left(1+2\lambda M_{0,n}\right)d_{0,j}+2\lambda M_{0,n}d_{1,j}$$

Thus we end up with n simultaneous equations in n unknowns, where n is the number of nodes. The coefficients can be arranged in a tridiagonal matrix as follows [25] [11]

$$\begin{bmatrix} e_0 & f_0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ d_1 & e_1 & f_1 & \cdots & 0 & 0 & 0 & 0 \\ . & . & . & . & \cdots & 0 & 0 & 0 \\ 0 & . & . & . & . & . & 0 & 0 \\ 0 & 0 & . & . & . & . & . & 0 \\ 0 & 0 & 0 & . & . & d_{n-2} & e_{n-2} & f_{n-2} \\ 0 & 0 & 0 & 0 & . & . & d_{n-1} & e_{n-1} \end{bmatrix} \begin{bmatrix} u_{0,j+1} \\ u_{1,j+1} \\ u_{2,j+1} \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ u_{N-1,j} \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ b_{N-1} \end{bmatrix}$$

When the coefficients are given by

$$e_0=1+2\lambda M_0$$

$$f_0=-2\lambda M_0$$

$$b_0=(1-2\lambda M_0)d_{0,j}+2\lambda M_{0,n}d_{1,j}$$

$$e_j=1+\frac{j+0.5}{2j}M(j)+\frac{\lambda(j-0.5)M(j-1)}{2j}$$

$$f_j=\frac{-\lambda(j+0.5)}{2j}M(j)$$

$$b_j=\frac{\lambda(j-0.5)}{2j}M(j-1)d_{i-1,j}+\left(1-\frac{\lambda(j+0.5)M(j)}{2j}-\frac{\lambda(j-0.5)M(j-1)}{2j}\right)d_{i,j}+\frac{\lambda(j+0.5)M(j)}{2j}d_{i+1,j}$$

And the matrix can be solving by any numerical methods.

Conclusion

The numerical results that have been obtained from testing the numerical methods that have been studied and developed through this work show the following conclusions:

For example 1, we have following results

Numerical Methods	Maximum Error
Forward Difference Method	$3.171 * 10^{-4}$
Backward Difference Method	$2.845 * 10^{-4}$
Crank-Nicolson Method	$2.676 * 10^{-4}$
Method of Line	$2.969 * 10^{-4}$
Pade’ Approximation (1,1)	$2.9683 * 10^{-4}$
Pade’ Approximation (0,2)	$2.9704 * 10^{-4}$
Pade’ Approximation (2,0)	$2.9705 * 10^{-4}$
Pade’ Approximation (1,2)	$2.9690 * 10^{-4}$
Pade’ Approximation (2,1)	$3.8465 * 10^{-3}$
Pade’ Approximation (2,2)	$3.2 * 10^{-5}$

From the above table we can see that Pade Approximation (2,2) Method is the most efficient method for solving example 1.

We have used the numerical methods that have been developed in our work for solving example 2 and get the following results:

Numerical Methods	Maximum Error
Forward Difference Method	$2.5070 * 10^{-2}$
Backward Difference Method	$2.5070 * 10^{-2}$
Crank-Nicolson Method	$2.2638 * 10^{-2}$
Method of Line	$2.2638 * 10^{-3}$
Pade’ Approximation (1,1)	$2.2638 * 10^{-2}$
Pade’ Approximation (0,2)	$2.2638 * 10^{-2}$
Pade’ Approximation (2,0)	$2.2638 * 10^{-2}$
Pade’ Approximation (1,2)	$2.2638 * 10^{-2}$
Pade’ Approximation (2,1)	$9.9787 * 10^{-2}$
Pade’ Approximation (2,2)	$3.8722 * 10^{-4}$

From the above table we can see that Pade Approximation (2,2) Method is the most efficient method for solving example 2.

Reference

[1] Richard L. Burden, J. Douglas Faires, **Numerical Analysis (9th edition)**, Brooks Cole, 2010.

[2] Josef Kallrath, **On Rational Function Techniques and Pad´e Approximants**, Ludwigshafen, Germany, BASF – AG, 2012.

[3] C. Brezinski, **History of Continued Fractions and Pad´e Approximants**, Springer-Verlag, Berlin, 1991, (79-81).

[4] E. B. Saff, **Introduction to Pade Approximants**, Vanderbilt University, Center for Constructive Approximation, 1986, (14-16).

[5] Kendall E. Atkinson, Weimin Han, David E. Stewart, **Numerical Solution of Ordinary Differential Equations**,2009, (4-11) (95-111).

[6] S.A. Matar, M.J. Edirisinghe, J.R.G. Evans and E.H. Twizell, **The Effect of Porosity Development on the Removal of Organic Vehicle from Moulded Ceramic Bodies**, Journal. Of Materials Research, **8** (1993) 617-625, Materials Research Society, U. S. A.

[7] S.A. Matar, M.J. Edirisinghe, J.R.G. Evans and E.H. Twizell, **Modelling the Removal of Organic Vehicle from Ceramic or Metal Mouldings: The Effect of Gas Permeation on the Incidence of Defects**, Journal. Of Materials Science, **30** (1995) 3805-3810,

[8] S.A. Matar, J.R.G. Evans, M.J. Edirisinghe and E.H. Twizell**, The Influence of Monomer and Polymer Properties on the Removal of Organic Vehicle from Moulded Ceramic Bodies**, Journal of Materials Research **10** (1995) 2060 – 2072, Materials Research Society, U. S. A.

[9] S.A. Matar, M.J. Edirisinghe, J.R.G. Evans and E.H. Twizell, **Diffusion of Degradation Products in Ceramic Mouldings During Pyrolysis: Effect of Geometry**, Journal of American Ceramics Society, 79 (1996) (749 – 755).

[10] K. Kubota, F. Dey, S.A. Matar and E.H. Twizell, **A repeated-dose model of percutaneous drug absorption**.

[11] G.D. SMITH, **Numerical Solution of Partial Differential Equations: Finite Difference Methods (3rd ed.)**,1984 Oxford, (ch.2 –ch.3).

[12] Mark H. Holmes, **Introduction to Numerical Methods in Differential Equations (Texts in Applied Mathematics)**, New York,2006, pp (92-135).

[13] Stanley J. Farlow, **Partial differential Equations for Scientists and Engineers,** New York,1993, pp (14-35).

[14] Victor Ivrii, **Partial Differential Equations,** Department of mathematics, University of Toronto ,2017, pp (1-25).

[15] William E Schiesser, **Method of Lines Analysis of Turing Models**, World Scientific Publishing Company (2017), pp (224-245).

[16] Abdul-Majid Wazwaz, **Partial Differential Equations and Solitary Waves Theory,** Chicago, IL 60655,2009, pp (92-110).

[17] ChenYi-Tung, LiJichun, **Computational Partial Differential Equations Using MATLAB,** Chapman Hall /CRC Applied Mathematics Nonlinear Science ,2008, pp (90-110).

[18] Kesan C., **Taylor polynomial solutions of second order linear partial differential equations.** Applied Mathematics and Computation.152 ,2004, pp (19-55).

[19] Milne, William Edmund ,**Numerical solution of differential equations (1st ed.). Wiley**. 2017, pp. (128–131).

[20] William E. Boyce , Richard C. DiPrima, **Elementary Differential Equations and Boundary Value Problems (7th ed.),** 2017,pp. (439-455).

[21] Martha Abell, Emil Iacob, **Topics in Classical and Modern Analysis: In Memory of Yingkang Hu (Applied and Numerical Harmonic Analysis),** 2019.

[22] Joshua Pelleg, **Diffusion in Ceramics-Springer International Publishing** ,2016, pp (15-100).

[23] Helmut Mehrer, **Diffusion in Solids,** 2007, pp (112-121,612-618,246-654).

[24] Vrentas and Duda, **free volume theory,** 1982.

[25] Costas Lefcaditis, **Numerical Methods for Diffusion Problems and Their Applications in Ceramic**

Mouldings,1994, pp (42-61).

جامعة النجاح الوطنية

كلية الدراسات العليا

دراسة لطرق عددية برتب عليا لحل المعادلات التفاضلية الجزئية المكافئة مع تطبيقاتها

إعداد

أحمد "محمد عمر" راضي قصف

إشراف

د. سمير مطر

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

2019

دراسة لطرق عددية برتب عليا لحل المعادلات التفاضلية الجزئية المكافئة مع تطبيقاتها

إعداد

أحمد "محمد عمر" راضي قصف

إشراف

د. سمير مطر

الملخص

نظرا لأهمية المسائل التفاضلية الجزئية المكافئة ومدى استخدامها الواسع في مجالات الدراسات العلمية والفيزيائية والهندسية .نشرت الكثير من الدراسات والبحوث وايضا طورت العديد من الطرق العددية لحلها ولكن في هذه الرسالة تم التركيز على المسائل التفاضلية الجزئية من النوع المكافئ .

لقد قمنا في هذه الرسالة بدراسة بعض الطرق العددية كطريقة تقريب المشتقات عن طريق الفروق المحدودة وطريقة كرانك نيكلسون وطريقة الخطوط كما استخدمنا تقريب بادي من الدرجة (1,1) ومن الدرجة (0,2) ومن الدرجة (2,0) ومن الدرجة (1,2) ومن الدرجة (2,1) ومن الدرجة (2,2)

بعض الامثلة قد تم حلها لاختبار مدى فعالية الطرق العددية التي تم دراستها وتطويرها في هذه الرسالة لتحديد الطريق الافضل فعالية وقد تم اجراء مقارانات بين النتائج التحليلية والنتائج التقريبية . وقد اظهرت بعض الطرق العددية دقة عالية في قرب النتائج التحليلية من النتائج العددية.