DIFFERENTIAL QUADRATURE METHOD FOR TIME-DEPENDENT DIFFUSION EQUATION

MAKBULE AKMAN

NOVEMBER 2003

DIFFERENTIAL QUADRATURE METHOD FOR TIME-DEPENDENT DIFFUSION EQUATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

MAKBULE AKMAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF MATHEMATICS

NOVEMBER 2003

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Ersan AKYILDIZ
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Münevver TEZER
Supervisor

Examining Committee Members

Prof. Dr. Kemal LEBLEBİCİOGLU

Prof. Dr. Bülent KARASÖZEN

Prof. Dr. Hasan TAŞELİ

Prof. Dr. Münevver TEZER

Assoc. Prof. Dr. Tanıl ERGENÇ

ABSTRACT

DIFFERENTIAL QUADRATURE METHOD FOR TIME-DEPENDENT DIFFUSION EQUATION

Makbule Akman
M.Sc., Department of Mathematics
Supervisor: Prof. Dr. Münevver Tezer

November 2003, 75 pages

This thesis presents the Differential Quadrature Method (DQM) for solving time-dependent diffusion or heat conduction problem. DQM discretizes the space derivatives giving a system of ordinary differential equations with respect to time and the fourth order Runge Kutta Method (RKM) is employed for solving this system. Stabilities of the ordinary differential equations system and RKM are considered and step sizes are arranged accordingly.

The procedure is applied to several time dependent diffusion problems and the solutions are presented in terms of graphics comparing with the exact solutions. This method exhibits high accuracy and efficiency comparing to the other numerical methods.

Keywords: Time-dependent diffusion equation, Differential quadrature method, Runge-Kutta method.

ÖZ

ZAMANA BAĞLI DİFÜZYON DENKLEMİ İÇİN DİFERENSİYEL KARE YAPMA METODU

Makbule Akman
Yüksek Lisans, Matematik Bölümü
Tez Danışmanı: Prof. Dr. Münevver Tezer

Kasım 2003, 75 sayfa

Bu tez zamana bağlı difüzyon veya ısı iletim problemini çözmek için diferensiyel kare yapma metodunu sunmuştur. Diferensiyel kare yapma metodu uzay türevlerini zamana bağlı bir adi diferensiyel denklem sistemi verecek şekilde ayırır ve bu sistemi çözmek için dördüncü dereceden Runge-Kutta metodu uygulanmıştır. Adi diferensiyel denklem sisteminin ve Runge-Kutta metodunun stabilitesi göz önünde tutulmuş ve adım genişliği ona göre ayarlanmıştır.

Bu yöntem, birçok zamana bağlı difüzyon denklemine uygulanmıştır ve çözümler analitik çözümle karşılaştırılarak grafikler ile sunulmuştur. Bu metod diğer nümerik metodlarla karşılaştırıldığında yüksek doğruluk ve etkinlik göstermiştir.

Anahtar Kelimeler: Zamana bağlı difüzyon denklemi, Diferensiyel kare yapma metodu, Dördüncü dereceden Runge-Kutta metodu.

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to the members of the Mathematics Department of Middle East Technical University, especially to Prof. Dr. Münevver TEZER for her valuable guidance and helpful suggestions in every aspect from the very beginning onwards.

Deepest thanks to Sevin Gümgüm for her close friendship and valuable support during the thesis period. In completion of this thesis, I would like to also thank Öğ.Bnb.Nejla BİLECEN for her virtual support and significant helpful, and TUBİTAK for its monetary support.

Finally, deepest regards and love to my parents and to my family, who have provided me the love, support and educational background that has enabled me to be successful both in my personal and academic life.

And I would like to dedicate this thesis to my family...

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The Differential Quadrature Method(DQM) is a numerical solution technique for initial and/or boundary value problems. It was developed by the late Richard Bellmann and his associates in the early 70s and since then, the technique has been succesfully employed in a variety of problems in engineering and phsical sciences. The method has been projected by its proponents as a potential alternative to the conventional numerical solution techniques such as the finite difference and finite element methods. (Bert,Malik (1996)).

The DQ method, akin to the convential integral quadrature method, app-roximates the derivative of a function at any location by a linear summation of all the functional values along a mesh (grid) line. The key procedure in the DQ application lies in the determination of the weighting coefficients. The DQ method and its applications were rapidly developed after the late 1980s, thanks to the innovative work in the computation of the weighting coefficients by other researchers. As a result, the DQ method has emerged as a powerful numerical discretization tool in the past decade. (Shu and Richards(1990), Shu(2000)).

At the first time the Differential Quadrature Method was mentioned in a book written by Bellman and Roth in 1986. There are many innovative ideas contained in this book. In 1996, Bert and Malik presented a comprehensive review of the chronological development and the application of the DQ method. The textbook of Shu (2000) represents the first comprehensive work on the DQ method and applications. However, there is no abundant book which systematically describes both the theoretical analysis and the application of the DQ method. Since there are many achievements in the DQ method, the number of reference books on the DQ method and its applications will increase.

In seeking an efficient discretization technique to obtain accurate numerical solutions using a considerably small number of grid points, Bellman(1971, 1972) introduced the method of DQ where a partial derivative of a function with respect to a coordinate direction is expressed as a linear weighted sum of all the functional values at all grid points along that direction. Bellman(1972) suggested two methods to determine the weighting coefficients of the first order derivative. The first method solves an algebraic equation system. The second uses a simple algebraic formulations, but with the coordinates of grid points chosen as the roots of the Legendre Polynomials. Unfortunately, when the order of the algebraic equation system is large, its matrix is ill-conditioned. Thus, it is difficult to obtain the weighting coefficients.

To further improve the computation of, Quann and Chang(1989a,b) applied Lagrange Interpolated polynomials as test functions and obtained explicit formulations to calculate the weighting coefficients for the discretization of the first and second order derivatives.

Shu and Richards(1990) generalized all the current methods for determination of the weighting coefficients under the analysis of a high order polynomial approximation and the analysis of a linear vector space. The weighting coefficients of the first order derivative are determined by a simple algebraic formulation whereas the weighting coefficients of the second and higher order derivatives are determined by a recurrence relationship.

Applications of DQ method may be found in the available literature include biosciences, transport processes fluid mechanics, statik and dynamic structural mechanics, statik aeroelasticity and lubrication mechanics. It has been claimed that the DQ method has the capability of producing highly accurate solutions with minimal computational effort. (Bert, Malik(1996)).

Civan(1994) considered a proper application of the DQ method for the numerical solution of a model of an isothermal reactor with axial mixing and showed that the DQ method alleviates the numerical difficulties encountered in finite difference and quadrature solutions while satisfying the boundary conditions accurately.

Fung(2001) solved first order initial value problems by taking the time derivative at a sampling grid point as a weighted linear sum of the given initial condition and the function values. His algorithm is unconditionally stable and the sampling grid points are the roots of Legendre Polynomials. In part II, he extended this differential quadrature algorithm to solve second order initial value problems.

In a later paper, Fung(2003) modified the weighting coefficient matrices in the DQM for the imposition of boundary conditions containing higher order derivatives.

A Differential Quadrature Method proposed by Wu and Liu(1999) chooses the function values and some derivatives whenever necessary as independent variables. Therefore, the delta-type grid arrangement used in the classic DQM is exempt while applying the boundary condition exactly. The explicit weighting coefficients can also be obtained.

In the past decade, the DQ method has succesfully applied with explicit computation of the weighting coefficients to the simulation of many incompressible viscous flows, free vibration analysis of beams, plates and shells. Apart from the work on the explicit computation of the weighting coefficients and its application in various areas, significant contributions were made in the theoretical analysis such as the error estimates, relationship between the DQ method and conventional discretization techniques, effect of the grid point distribution on the accuracy of the DQ results and the stability condition (Shu, 2000)

The technique of Differential Quadrature Method for the solution of two-dimensional partial differential equations is extended by Lam(1993) to encompass problems with arbitrary geometry. He compared the results of thermal and torsional problems with other solutions and showed reasonably good accuracy.

For solving steady-state heat conduction problems by using the irregular elements of the differential quadrature method is proposed by Chang(1999). He used the mapping technique to transform the governing partial differential equation with the natural transition condition of two adjacent elements and the Neumann boundary condition defined on the irregular physical element into the parent space. Then the DQ technique is used to discretize the transformed relation equation defined on the regular element.

Rapaci(1991) deals, in the one dimensional case, with an inverse problem for the Heat Equation. Such a problem is a two point initial-boundary value problem with boundary conditions. The missing boundary conditions are replaced by a measurement of the temperature in an inner point of the space domain. He obtained the results by using the DQM.

Chawla and Al-Zanaidi(2001) described a locally one-dimensional (LOD) time integration scheme for the diffusion equation in two space dimensions based on the extended trapezoidal formula (ETF). The resulting LOD-ETF scheme is third order in time and is unconditionally stable preventing unwanted oscillations in the solution.

Tanaka and Chen(2001) presents the application of dual reciprocity BEM (DRBEM) and differential quadrature (DQ) method to time-dependent diffusion problems. The DRBEM is employed to discretize the spatial partial derivatives. The time derivative is discretized by using DQM. But the resulting algebraic formulation is the known Lyapunov matrix equations which require a special solution technique.

In this thesis, Differential Quadrature Method is applied to time dependent diffusion problem in two-dimensional space. The diffusion equation is supplemented with Dirichlet or Dirihlet-Neumann type boundary conditions together with an initial condition forming an initial and boundary value problem in two-dimensional space. Time dependent boundary conditions as well as time dependent coefficients in the equation are also discussed and accompanied with sample problems. For simplicity, one dimensional case is chosen to demonstrate the DQM application. Differential Quadrature technique approximates the derivative of a function at a grid point by a linear weighted summation of all the functional values. Derivatives with respect to space variables are discretized using DQM giving a system of ordinary differential equations for the time derivative. We used fourth order explicit Runge-Kutta Method for discretizing time derivatives since its stability region is larger comparing to the other time integration methods and simple for the computations. This DQM and 4th order RK method combination gives very good numerical technique for solving time dependent diffusion problems. Stability criterias are also controlled with several values of time increment Δt and number of grid points N in space region. Implicit RK method is not preferred because of its complexity in the computations.

1.1 Plan of The Thesis

In Chapter 2, first the theory of the Differential Quadrature Method is given in one dimensional case and then the extension to the two dimensional case and the application to our problem is explained. Implementation of Dirichlet and Neumann type boundary conditions to the final system of ordinary differential equations is also presented. Then discretization of the time derivative by using fourth order Runge-Kutta Method is introduced.

Chapter 3 presents application of DQ method and RKM to solve time dependent Diffusion equation. Dirichlet and Dirichlet-Neumann type boundary conditions are used. Time dependent boundary conditions and time dependent coefficients in the equations are also demonstrated with some problems. Choices of Δt and N are discussed in terms of stability requirements.

CHAPTER 2

DIFFERENTIAL QUADRATURE METHOD FOR DIFFUSION EQUATION

This chapter presents the combined application of Differential Quadrature Method (DQM) and The Fourth Order Runge-Kutta Method (RKM) to time-dependent transient diffusion equation.

The equation governing transient diffusion problems can be expressed as

$$\frac{\partial\ u(x,y,t)}{\partial t} - \nabla^2 u(x,y,t) = 0 \qquad (x,y) \in \Omega \tag{2.1}$$

with the initial condition

$$u(x,y,0) = u_0(x,y) \tag{2.2}$$

and the Dirichlet, Neumann and Linear Radiation boundary conditions are given by

$$u(x,y,t) = \bar{u}(x,y,t) \qquad (x,y) \in \Gamma_u \tag{2.3}$$

$$q(x,y,t) = \partial\bar{u}(x,y,t)/\partial n = \bar{q}(x,y,t) \qquad (x,y) \in \Gamma_q \tag{2.4}$$

$$q(x,y,t) = -h(x,y,t)(u(x,y,t) - u_r(x,y,t)) \qquad (x,y) \in \Gamma_r \tag{2.5}$$

where variable domain $\Omega \in R^2$ is bounded by a piecewise smooth boundary $\Gamma = \Gamma_u + \Gamma_q + \Gamma_r$. Here $q = \frac{\partial u}{\partial n}$, n is the unit outward normal, $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the Laplace operator. The functions $\bar{u}(x,y,t)$, $\bar{q}(x,y,t)$, $h(x,y,t)$ and $u_r(x,y,t)$ are given functions of (x,y,t) and the initial condition $u_0(x,y)$ is a given function of space variables (x,y).

The transient diffusion Equation (2.1)

$$\nabla^2 u = \frac{\partial u}{\partial t} \tag{2.6}$$

can be considered in the general form

$$\nabla^2 u = b(x, y, t, u, u_t) \tag{2.7}$$

where (x,y) represents the spatial coordinates, u is the dependent unknown solution, t is the time and u_t denotes the time derivative of u.

Apart from the Differential Quadrature Method, there are many available numerical discretization techniques to solve transient diffusion equation such as the Dual Reciprocity Boundary Element Method, Finite Element Method, Finite Difference Method, Finite Volume Method. As compared to these lower order numerical methods, the Differential Quadrature (DQ) Method can obtain very accurate numerical results using a considerably smaller number of grid points and hence requiring relatively little computational effort. It will be shown in this thesis that the polynomial-based differential quadrature method is an efficient approach to solve the transient diffusion problem.

Differential Quadrature Method will be applied to Equation (2.7) containing Laplace Operator. Since right hand side function b contains also the time derivative of u, the resulting system of equations will contain u_t. Therefore, the time derivatives will be discretized by one of the explicit shemes which is the Fourth Order Runge-Kutta Method in this thesis.

2.1 Differential Quadrature Method

The Differential Quadrature Method was presented by R.E. Bellman and his associates in early 1970's and it is a numerical discretization technique for the approximation of derivatives. In seeking an efficient discretization technique to obtain accurate numerical solutions using a considerably small number of grid points, Belmann (1971,1972) introduced the method of differential quadrature, where a partial derivative of a function with respect to a coordinate direction is expressed as a linear weighted sum of all the functional values at all grid points along that direction. The key to DQ Method is to determine the weighting coefficients for the discretization of a derivative of any order. A major breakthrough in computing the weighting coefficients was made by Shu and Richards (1990) in which all the current methods for determination of the weighting coefficients are generalized under the analysis of a high order polynomial approximation and the analysis of a linear vector space. In Shu's approach, the weighting coefficients of the first order derivative are determined by a simple algebraic formulation without any restriction on the choice of grid (mesh) points, whereas the weighting coefficients of the second and higher order derivatives are determined by a recurrence relationship. Clearly, all the work is based on the polynomial approximation and accordingly, the related DQ method can be considered as the Polynomial-Based Differential Quadrature (PDQ) Method. Recently, Shu and Chew (1997) and Shu and Xue (1997) have developed some simple algebraic formulations to compute the weighting coefficients of the first and second order derivatives in the DQ approach when the function or the solution of a a partial differential equation (PDE) is approximated by a Fourier series expansion. These formulations are different from PDQ and the approach can be termed as the Fourier Expansion-Based Differential Quadrature (FDQ) Method.

Depending on the feature of the problem, the solution of a PDE can be approximated by a polynomial of high degree or by the Fourier series expansion. The FDQ approach is more suitable for problems with harmonic behaviours such as the Helmholtz equation. Since the transient diffusion equation does not have this kind of behaviour, the PDQ Method will be used in this thesis.

For simplicity, the one-dimensional problem is chosen to demonstrate the PDQ Method. When a structured grid is used, the one dimensional results can be directly extended to the multi-dimensional cases and thus to our problem in two dimensions.

2.1.1 One Dimensional Polynomial-Based Differential Quadrature Method

Following the idea of integral quadrature, the Differential Quadrature Method approximates the derivative of a smooth function at a grid point by a linear weighted summation of all the functional values in the whole computational domain (Shu (2000)). For example, the first and second order derivatives of a function $f(x)$ at a point x_i are approximated by

$$f_x(x_i) = \frac{df}{dx}|_{x_i} = \sum_{j=1}^{N} a_{ij} f(x_j), \qquad i = 1, 2...N \tag{2.8}$$

$$f_{xx}(x_i) = \frac{d^2 f}{dx^2}|_{x_i} = \sum_{j=1}^{N} b_{ij} f(x_j), \qquad i = 1, 2...N \tag{2.9}$$

where a_{ij}, b_{ij} are the weighting coefficients and N is the number of grid points in the whole domain. It should be noted that the weighting coefficients a_{ij}(and b_{ij}) are different at different location of x_i since they depend on coordinates of the points. The important procedure in DQ approximation is to determine the weighting coefficients a_{ij} and b_{ij} efficiently.

When the function $f(x)$ is approximated by a high order polynomial, one needs some explicit formulations to compute the weighting coefficients within the scope of a high order polynomial approximation and a linear vector space. In accordance with the Weierstrass polynomial approximation theorem, it is known that the solution of a one dimensional differential equation is approximated by a (N-1)th degree polynomial

$$f(x) = \sum_{k=0}^{N-1} c_k x^k \tag{2.10}$$

where c_k's are constants. The polynomial of degree less than or equal to N-1 constitutes an N dimensional linear vector space V_N with respect to the operation of vector addition and scalar multiplication.

Obviously, in the linear vector space V_N, a set of vectors $1, x, x^2, ..., x^{N-1}$ is linearly independent. Thus,

$$s_k(x) = x^{k-1}, \qquad k = 1, 2, ..., N \tag{2.11}$$

is a basis of V_N.

For the numerical solution of a differential equation, we need to find out the solution at certain discrete points. Now, it is supposed that in a closed interval [a,b], there are N distinct grid points with the coordinates $a = x_1, x_2, ..., x_N = b$ and the functional value at a grid point x_i is $f(x_i)$. Then the constants in Equation (2.10) can be determined from the following system of equations

$$\begin{aligned} c_0 + c_1 x_1 + c_2 x_1^2 + ... + c_{N-1} x_1^{N-1} &= f(x_1) \\ c_0 + c_1 x_2 + c_2 x_2^2 + ... + c_{N-1} x_2^{N-1} &= f(x_2) \\ &\cdots\cdots\cdots\cdots\cdots \\ c_0 + c_1 x_N + c_2 x_N^2 + ... + c_{N-1} x_N^{N-1} &= f(x_N). \end{aligned} \tag{2.12}$$

The matrix of Equation (2.12) is of Vandermonde form which is not singular. Thus the equation can give unique solutions for constants $c_0, c_1, ..., c_{N-1}$. Once these are determined, the approximated polynomial is obtained. However, when N is large, the matrix is highly ill-conditioned and its inversion is very difficult to find. Then it is hard to determine the constants $c_0, c_1, ..., c_N$.

Here, if $r_k(x)$, $k = 1, 2, ..., N$ are the base polynomials in V_N, $f(x)$ can then be expressed by

$$f(x) = \sum_{k=1}^{N} d_k r_k(x). \tag{2.13}$$

Clearly, if all the base polynomials satisfy a linear constrained relationship such as Equation (2.8) or Equation (2.9), so does $f(x)$. In the linear vector space, there may exist several sets of base polynomials. Each set of base polynomials can be expressed uniquely by another set of base polynomials. This means that every set of base polynomials would give the same weighting coefficients. However, the use of different sets of base polynomials will result in different approaches to compute the weighting coefficients. Since there are many sets of base polynomials in the linear vector space, we have many approaches to compute the weighting coefficients.

The property of linear vector space also gives us the ability to apply the weighting coefficients for the discretization of a differential equation. Remembering that the solution of a differential equation is approximated by a polynomial of degree (N-1) which constitutes the N-dimensional vector space, the actual expression of the polynomial contains the unknown constants c_k's which are to be determined. On the other hand, in the linear vector space, the set of base polynomials can be chosen to be independent of the solution. From the property of a linear vector space, if one set of base polynomials satisfies a linear operator so does any polynomial in the space. This indicates that the solution of the partial differential equation also satisfies the linear operator.

Here, for generality, two sets of base polynomials are used to determine the weighting coefficients (Shu, 2000). The first set of base polynomials is chosen as the Lagrange interpolated polinomials,

$$r_k(x) = \frac{M(x)}{(x - x_k)M^{(1)}(x_k)}, \qquad k = 1, 2, ..., N \tag{2.14}$$

where

$$M(x) = (x - x_1)(x - x_2)...(x - x_N) \tag{2.15}$$

and

$$M^{(1)}(x_k) = \prod_{j=1, j\neq k}^{N} (x_k - x_j) \tag{2.16}$$

being the derivative of $M(x)$.

Here $x_1, x_2, ..., x_N$ are the coordinates of grid points and may be chosen arbitrarily but distinct.

For obtaining an efficient procedure to compute the polynomials $r_k(x)$ at discrete points we make use of Kronecker operator as

$$M(x) = N(x, x_k)(x - x_k), \qquad k = 1, 2, ..., N \tag{2.17}$$

with

$$N(x_i, x_j) = M^{(1)}(x_i)\delta_{ij}. \tag{2.18}$$

Using Equation (2.17), Equation (2.14) can be simplified to

$$r_k(x) = \frac{N(x, x_k)}{M^{(1)}(x_k)}, \qquad k = 1, 2, ...N \tag{2.19}$$

and at the point x_i,

$$r_k(x_i) = \frac{N(x_i, x_k)}{M^{(1)}(x_k)}, \qquad i = 1, 2, ...N, \qquad k = 1, 2, ..., N. \tag{2.20}$$

From Equation (2.18), we can obtain the following expression as

$$N(x_i, x_k) = M^{(1)}(x_i)\delta_{ik} = \begin{cases} 0 & \text{if} \quad i \neq k \\ M^{(1)}(x_i) & \text{if} \quad i = k \end{cases} \tag{2.21}$$

giving $r_k(x_i) = 0$ if $i \neq k$ and $r_k(x_i) = 1$ for $i = k$. Using this property of $r_k(x)$ when $i = k$ in the Equation (2.13) at the point x_i, we obtain

$$f(x_i) = \sum_{k=1}^{N} d_k r_k(x_i) = d_i. \tag{2.22}$$

Then $f(x_i)$ takes the form

$$f(x_i) = \sum_{k=1}^{N} f(x_k) r_k(x_i). \tag{2.23}$$

Thus the first and second order derivatives of $f(x)$ with respect to x at the point x_i are

$$f_x(x_i) = \sum_{k=1}^{N} r'_k(x_i) f(x_k) \tag{2.24}$$

$$f_{xx}(x_i) = \sum_{k=1}^{N} r''_k(x_i) f(x_k). \tag{2.25}$$

From Equation (2.8) and Equation (2.9), the coefficients a_{ij} and b_{ij} in the first and second order derivatives of $f(x)$ at the point x_i become

$$r'_k(x_i) = a_{ik} \tag{2.26}$$

$$r''_k(x_i) = b_{ik}. \tag{2.27}$$

Thus the coefficients a_{ij} and b_{ij} can be computed by taking first and second order derivatives of $r_k(x)$ as follows,

$$r'_j(x_i) = \frac{N^{(1)}(x_i, x_j)}{M^{(1)}(x_j)} = a_{ij} \tag{2.28}$$

$$r''_j(x_i) = \frac{N^{(2)}(x_i, x_j)}{M^{(1)}(x_j)} = b_{ij} \tag{2.29}$$

where $N^{(1)}(x, x_j)$ and $N^{(2)}(x, x_j)$ are the first and second order derivatives of the function $N(x, x_j)$.

We successively differentiate Equation (2.17) with respect to x and obtain the following recurrence formulation

$$M^{(m)}(x) = N^{(m)}(x, x_k)(x - x_k) + mN^{(m-1)}(x, x_k)$$

$$for \quad m = 1, 2, ..., N-1, \quad k = 1, 2, ..., N \tag{2.30}$$

where $M^{(m)}(x)$ and $N^{(m)}(x, x_k)$ indicate the m^{th} order derivative of $M(x)$ and $N(x, x_k)$ respectively.

From the Equation (2.30), we can easily obtain

$$N^{(1)}(x_i, x_j) = \frac{M^{(1)}(x_i)}{x_i - x_j}, \qquad i \neq j \tag{2.31}$$

$$N^{(1)}(x_i, x_i) = \frac{M^{(1)}(x_i)}{2} \qquad i = j. \tag{2.32}$$

Similarly, using Equation (2.30) for m=2 gives

$$N^{(2)}(x_i, x_j) = \frac{M^{(2)}(x_i) - 2N^{(1)}(x_i, x_j)}{x_i - x_j}, \quad i \neq j \tag{2.33}$$

$$N^{(2)}(x_i, x_i) = \frac{M^{(3)}(x_i)}{3}. \qquad i = j \tag{2.34}$$

Subsituting Equation (2.31) into Equation (2.28) and Equation (2.32) into Equation (2.29), we finally obtain the coefficients a_{ij} and b_{ij}

$$a_{ij} = \frac{M^{(1)}(x_i)}{(x_i - x_j)M^{(1)}(x_j)}, \qquad i \neq j \tag{2.35}$$

$$a_{ii} = \frac{M^{(2)}(x_i)}{2M^{(1)}(x_i)} \tag{2.36}$$

$$b_{ij} = 2a_{ij}(a_{ii} - \frac{1}{x_i - x_j}) \qquad i \neq j \tag{2.37}$$

$$b_{ii} = \frac{M^{(3)}(x_i)}{3M^{(1)}(x_i)}. \tag{2.38}$$

It is observed that, if x_i is given, it is easy to compute $M^{(1)}(x_i)$ from Equation (2.16) and hence a_{ij}, b_{ij} for $i \neq j$. However, the computation of a_{ii} (Equation (2.36)) and b_{ii} (Equation (2.38)) involve the computation of the second order derivative $M^{(2)}(x_i)$ and the third order derivative $M^{(3)}(x_i)$ which are not easy to compute. This difficulty can be eliminated by the property of the linear vector space.

According to the theory of a linear vector space, one set of base polynomials can be expressed uniquely by another set of base polynomials. Thus if one set of base polynomials satisfies a linear operator, say Equation (2.8) or Equation (2.9), so does another set of base polynomials. As a consequence, the Equations (2.8) and (2.9) should also be satisfied by the second set of base polynomials $x^k, k = 0, 1, 2, ..., N-1$. When k=0 this set of base polynomials gives

$$\sum_{j=1}^{N} a_{ij} = 0 \qquad or \qquad a_{ii} = -\sum_{j=1, j\neq i}^{N} a_{ij} \tag{2.39}$$

and

$$\sum_{j=1}^{N} b_{ij} = 0 \qquad or \qquad b_{ii} = -\sum_{j=1, j\neq i}^{N} b_{ij} \tag{2.40}$$

which are practical to compute once a_{ij}, b_{ij} $(i \neq j)$ are known.

Now, the PDQ method is going to be applied to transient diffusion problem in two-dimensions and the discretization for time derivative will be performed by using Runge-Kutta Method. The choice of the grid points is going to be given in a later section.

2.1.2 Two Dimensional Differential Quadrature Method For Time-Dependent Diffusion Equation

In previous Section Differential Quadrature Method was explained for one-dimensional case. The time dependent diffusion equation in two-dimensions has second order space derivatives and first order time derivative. The PDQ Method is applied for the discretization of space derivatives of the unknown function u, discretization with respect to time will be performed by using fourth order Runge-Kutta Method. The diffusion equation in two dimensions,

$$\frac{\partial u(x,y,t)}{\partial t} = \frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{\partial^2 u(x,y,t)}{\partial y^2} \tag{2.41}$$

can be discretized by using first PDQ method as

$$\sum_{k=1}^{N} w_{ik}^{(2)} u_{kj} + \sum_{k=1}^{M} \bar{w}_{jk}^{(2)} u_{ik} = \frac{\partial u(x_i, y_j, t)}{\partial t} \tag{2.42}$$

$$for \quad i = 1,2,...,N \quad and \quad j = 1,2,...,M$$

where N and M are the number of grid points in the x and y directions respectively. $w_{ik}^{(2)}$ and $\bar{w}_{jk}^{(2)}$ are the DQ weighting coefficients of the second order derivatives of u with respect to x and y respectively. Those coefficients $w_{ik}^{(2)}$, $\bar{w}_{jk}^{(2)}$ are computed analogous to the coefficients b_{ij} given in Equations (2.37), (2.38) for one dimensional case. For example,

$$w_{ik}^{(2)} = 2w_{ik}^{(1)}(w_{ii}^{(1)} - \frac{1}{x_i - x_k}) \qquad i,k = 1,2,...,N \tag{2.43}$$

$$\bar{w}_{jk}^{(2)} = 2\bar{w}_{jk}^{(1)}(\bar{w}_{jj}^{(1)} - \frac{1}{y_j - y_k}) \qquad j,k = 1,2,...,M \tag{2.44}$$

where $w_{ik}^{(1)}, w_{ii}^{(1)}, \bar{w}_{jk}^{(1)}$ and $\bar{w}_{jj}^{(1)}$ are also analogous to the coefficients a_{ij} of the first order derivatives of u with respect to x and y (Equations (2.35),(2.36))

$$w_{ik}^{(1)} = a_{ik} = \frac{M^{(1)}(x_i)}{(x_i - x_k)M^{(1)}(x_k)} \qquad i \neq k \tag{2.45}$$

$$w_{ii}^{(1)} = a_{ii} = \frac{M^{(2)}(x_i)}{2M^{(1)}(x_i)} \tag{2.46}$$

$$\bar{w}_{jk}^{(1)} = \bar{a}_{jk} = \frac{M^{(1)}(y_j)}{(y_j - y_k)M^{(1)}(y_k)} \qquad j \neq k \tag{2.47}$$

$$\bar{w}_{jj}^{(1)} = \bar{a}_{jj} = \frac{M^{(2)}(y_j)}{2M^{(1)}(y_j)}. \tag{2.48}$$

For the computation of diagonal entries $w_{ii}^{(1)}$, $\bar{w}_{jj}^{(1)}$, $w_{ii}^{(2)}$, $\bar{w}_{jj}^{(2)}$ again Equations (2.39) and (2.40) are made use of giving

$$w_{ii}^{(1)} = -\sum_{j=1, i\neq j}^{N} w_{ij}^{(1)}, \qquad w_{ii}^{(2)} = -\sum_{j=1, i\neq j}^{N} w_{ij}^{(2)} \quad i = 1, 2, ..., N, \tag{2.49}$$

$$\bar{w}_{jj}^{(1)} = -\sum_{i=1, i\neq j}^{M} \bar{w}_{ji}^{(1)}, \qquad \bar{w}_{jj}^{(2)} = -\sum_{i=1, i\neq j}^{M} \bar{w}_{ji}^{(2)} \quad j = 1, 2, ..., M. \tag{2.50}$$

Now, $\frac{\partial u}{\partial t}$ is also considered discretized as $\frac{\partial u_{ij}}{\partial t}$, thus Equation (2.42) is a set of DQ algebraic equations which can be written in a matrix form

$$[A]\{u\} = \{b\} \tag{2.51}$$

where $\{u\}$ is a vector of unknown NM functional values at all discretized points of the region, $[A]$ is the $NM \times NM$ coefficient matrix containing the weighting coefficients $w_{ik}^{(2)}$, $\bar{w}_{jk}^{(2)}$ and the right hand side vector $\{b\}$ of size $NM \times 1$ contains first order time derivatives of the function u at the same discretized points. Therefore a numerical scheme is necessary for handling these time derivatives. Equation (2.51) can be solved by several time integration schemes such as Euler, Modified Euler, Runge-Kutta Methods. Here, Runge-Kutta Method is going to be used since it is a one step method obtained from the Taylor series expansion of u up to and including the terms involving $(\Delta t)^4$ where Δt is the step size with respect to time.

2.1.3 Choice of Grid Points

Since the weighting coefficients $w_{ik}^{(2)}$, $\bar{w}_{jk}^{(2)}$ and $w_{ik}^{(1)}$, $\bar{w}_{jk}^{(1)}$ corresponding to the discretization of the second and first order derivatives respectively, contain grid points x_i, y_j's, the choice of these grid points becomes quite important. Equally spaced grid points, due to their obvious convenience, have been in use by most investigators. However, unequally spaced grid points especially the zeros of orthogonal polynomials like Legendre and Chebyshev polynomials usually give more accurate solutions then the equally spaced grid points.

The natural choice for the grid points is the equally spaced points which is given by

$$x_i = \frac{i-1}{N-1}a \ ; \qquad i = 1, 2, ..., N \tag{2.52}$$

and

$$y_j = \frac{j-1}{M-1}b \ ; \qquad j = 1, 2, ..., M \tag{2.53}$$

in the x and y directions, respectively for a region $[0, a] \times [0, b]$. For this uniform grid (equally spaced) with step sizes Δx and Δy in x and y directions respectively, one can obtain

$$x_k - x_i = (k-i)\Delta x, \qquad y_k - y_j = (k-j)\Delta y$$

$$M^{(1)}(x_i) = (-1)^{N-1}(\Delta x)^{N-1}(i-1)!(N-i)! \quad i = 1, 2, ..., N$$

$$M^{(1)}(y_j) = (-1)^{M-1}(\Delta y)^{M-1}(j-1)!(M-j)! \quad j = 1, 2, ..., M$$

and the coefficients for the first order derivatives reduce to

$$w_{ij}^{(1)} = (-1)^{i+j}\frac{(i-1)!(N-i)!}{\Delta x(i-j)(j-1)!(N-j)!} \quad i, j = 1, 2, ..., N, \ i \neq j$$

$$\bar{w}_{ij}^{(1)} = (-1)^{i+j}\frac{(i-1)!(M-i)!}{\Delta y(i-j)(j-1)!(M-j)!} \quad i, j = 1, 2, ..., M, \ i \neq j.$$

The so-called Chebyshev-Gauss-Lobatto point distribution offer a better choice and have been found consistently better than the equally spaced, Legendre and Chebyshev points in a variety of problems (Bert and Malik, 1996).

These points are the Chebyshev collacation points which are the roots of $|T_N(x)| = 1$ and given by (Shu,(2000))

$$x_k = Cos(\frac{k-1}{N-1}\pi) \qquad 1 \leq k \leq N$$

for an interval [-1,1]. For a region on [a,b]

$$x_i = \frac{1}{2}[1 - \cos{(\frac{i-1}{N-1})}\pi]a, \qquad i = 1, 2, ..., N \tag{2.54}$$

and

$$y_j = \frac{1}{2}[1 - \cos{(\frac{j-1}{M-1})}\pi]b, \qquad i = 1, 2, ..., M \tag{2.55}$$

in the x and y directions, respectively.

For the Chebyshev-Gauss-Lobatto points we have (in x direction)

$$M^{(1)}(x_i) = (-1)^{i+1}N^2$$

$$M^{(2)}(x_i) = (-1)^i N^2 \frac{x_i}{1-x^2}$$

and the corresponding weighting coefficients simplify greatly.

In this thesis for discretization of the intervals, the Chebyshev-Gauss-Lobatto point distribution is going to be used since it gives real part of all the PDQ matrix eigenvalues strictly negative. This is the condition of obtaining stable solutions (Shu,(2000)) for the system of ordinary differential equations.

2.1.4 Implementation of Boundary Conditions

Proper implementation of boundary conditions is very important for the accurate solution. The insertion of Dirichlet type boundary conditions is straightforward since these known values contribute to the right hand side vector $\{b\}$ in the system (2.51). If the boundary condition involves normal derivatives of the unknown function u then these derivatives can also be approximated by the Differential Quadrature Method.

Dirichlet Type Boundary Condition:

For imposing the Dirichlet type boundary conditions, the Equation (2.42) should only be applied at the interior points since the solution at the boundary grid points is known. Thus, the Equation (2.42) can be rewritten as

$$\sum_{k=2}^{N-1} w_{ik}^{(2)} u_{kj} + \sum_{k=2}^{M-1} \bar{w}_{jk}^{(2)} u_{ik} = \frac{\partial u_{ij}}{\partial t} - s_{ij} \tag{2.56}$$

where $2 \leq i \leq N-1, \;\; 2 \leq j \leq M-1$ and

$$s_{ij} = (w_{i1}^{(2)} u_{1j} + w_{iN}^{(2)} u_{Nj} + \bar{w}_{j1}^{(2)} u_{i1} + \bar{w}_{jM}^{(2)} u_{iM}).$$

Equation (2.56) is a set of DQ algebraic equations which can be written in matrix form

$$[A]\{u\} = \{b\} - \{s\} \tag{2.57}$$

where $\{u\}$ is a vector of unknown functional values at all the interior points given by

$$\{u\} = \{u_{22}, u_{23}, ..., u_{2,M-1}, u_{32}, ..., u_{3,M-1}, ..., u_{N-1,2}, ..., u_{N-1,M-1}\}^T$$

and {b} is a vector still containing discretized time derivatives of u and {s} vector contains known values of u at the boundary grid points. The size of the matrix $[A]$ is (N-2)(M-2) by (N-2)(M-2). Equation (2.57) can be solved by using iterative methods for discrete time derivative values of u after the time derivative is discretized. As mentioned before the fourth order Runge-Kutta Method is going to be used in this thesis for discretizing the time derivative $\dot{u}$.

Neumann Type Boundary Condition:

For the Neumann conditions the normal derivatives on the boundary should also be discretized by polynomial-based differential quadrature method.

The normal derivative of u can be written as

$$\frac{\partial u}{\partial n} = \frac{\partial u}{\partial x} n_x + \frac{\partial u}{\partial y} n_y \tag{2.58}$$

and $\partial u/\partial x$ and $\partial u/\partial y$ are discretized by using PDQ Method.

Now,

$$\frac{\partial u_{ij}}{\partial x} = \sum_{k=1}^{N} w_{ik}^{(1)} u_{kj}, \qquad i = 1, 2, ..., N \tag{2.59}$$

and

$$\frac{\partial u_{ij}}{\partial y} = \sum_{k=1}^{M} w_{jk}^{(1)} u_{ik}, \qquad j = 1, 2, ..., M \tag{2.60}$$

where $w_{ik}^{(1)}$ and $\bar{w}_{jk}^{(1)}$ are the weighting coefficients with respect to x and y directions and obtained analogously in the one dimensional case (Equation (2.35)).

Thus,

$$w_{ik}^{(1)} = \frac{M^{(1)}(x_i)}{(x_i - x_k) M^{(1)}(x_k)}, \qquad i \neq k \tag{2.61}$$

$$w_{ii}^{(1)} = \frac{M^{(2)}(x_i)}{2M^{(1)}(x_i)} \tag{2.62}$$

$$\bar{w}_{jk}^{(1)} = \frac{M^{(1)}(y_j)}{(y_j - y_k) M^{(1)}(y_k)} \qquad j \neq k \tag{2.63}$$

$$\bar{w}_{jj}^{(1)} = \frac{M^{(2)}(y_j)}{2M^{(1)}(y_j)}. \tag{2.64}$$

Assuming $\partial u_{Nj}/\partial x = c_j \;\; (j = 1, 2, ..., M)$ and $\partial u_{Nj}/\partial y = 0$ are given on one part of the boundary, we can write

$$\frac{\partial u_{Nj}}{\partial x} = \sum_{k=1}^{N} w_{Nk}^{(1)} u_{kj} = c_j. \tag{2.65}$$

Rewriting Equation (2.65) as

$$w_{NN}^{(1)}u_{Nj} + \sum_{k=1}^{N-1} w_{Nk}^{(1)}u_{kj} = c_j \tag{2.66}$$

u_{Nj} is easily obtained as a value on the boundary

$$u_{Nj} = \frac{1}{w_{NN}^{(1)}}(c_j - \sum_{k=1}^{N-1} w_{Nk}^{(1)}u_{kj}) \qquad j = 1, 2, ...M. \tag{2.67}$$

These M equations for the unknowns u_{Nj}, $(j = 1, 2, ..., M)$ are going to be added to the DQ system of equations (2.42) which is written for $i \neq N$, $j = 1, 2, ..., M$ for the case of Neumann type of boundary conditions $\frac{\partial u_{Nj}}{\partial x} = c_j$ on $x = x_N$, (i=N case). When normal boundary conditions are implemented at all the related grid points, the final system of DQ equations will be again in the form of Equation (2.57)

$$[A]\{u\} = \{b\} - \{s\}$$

where the vectors $\{b\}$ and $\{s\}$ still contains the discretized time derivatives of u and known values of u at the boundary grid points respectively.

Thus, equations found by discretizing the normal derivatives of u on the boundary are updated using interior u values which are not known yet. This problem is handled during the iterations performed in the Runge-Kutta Method since the initial u values are given.

The mixed type boundary conditions which are combinations of the Dirichlet and Neumann conditions are implemented in a similar fashion.

2.2 Runge-Kutta Method

The fourth order explicit Runge-Kutta Method (RKM) is applied to discretize time derivatives in the resulting system of algebraic equations (2.57). RKM is a one step method for solving initial value problems. Our time dependent diffusion problem is an initial and boundary value problem. Therefore the resulting algebraic system of equations Equation (2.57) can originally be considered as an initial value problem in the form (a set of ordinary differential equations in time)

$$\{b\} = \frac{d\{u\}}{dt} = [A]\{u\} + \{s\} \tag{2.68}$$

where $\{u\}$ is the unknown vector of values at the interior grid points, $\{s\}$ is the vector containing known boundary conditions and $[A]$ is the weighting coefficient matrix.

The fourth order explicit RKM is going to be explained for a ordinary differential equation (initial value problem) in the form

$$\dot{u} = f(t, u)$$

$$u(t_0) = u_0$$

and then modified for our system of equations.

One step methods for solving $\dot{u} = f(t, u)$ require only a knowledge of the numerical solution u_n in order to compute the next value u_{n+1}. This has advantages over the p-step multistep methods that use several past values $u_1, ..., u_p$ that have to be calculated by another method.

The best known one-setep methods are the Runge-Kutta Methods. They are fairly simple to program and their truncation error can be controlled more than for the multistep methods.(Atkinson,(1978)).

The most simple one-step method is based on Taylor series expansion.

Assume $U(x, y, t)$ is the solution of initial value problem

$$\dot{u} = f(t, u), \qquad u(x, y, t_0) = U_0. \tag{2.69}$$

Expanding $U(x, y, t_1)$ about t_0 using Taylor's theorem

$$U(x, y, t_1) = U(x, y, t_0) + \Delta t \dot{U}(x, y, t_0) + \ldots + \frac{(\Delta t)^r}{r!} U^{(r)}(x, y, t_0)$$
$$+ \frac{(\Delta t)^{r+1}}{(r+1)!} U^{(r+1)}(\xi) \tag{2.70}$$

for some $t_0 \leq \xi_0 \leq t_1$, and dropping the remainder term, we have an approximation for $U(x, y, t_1)$ provided that we can calculate $\ddot{U}(x, y, t_0), \ldots, U^{(r)}(x, y, t_0)$.

Differentiate $\dot{U}(x, y, t) = f(t, U)$ to obtain

$$\ddot{U}(x, y, t) = f_t(t, U(x, y, t)) + f_U(t, U(x, y, t)).\dot{U}(x, y, t)$$

$$\ddot{U} = f_t + f_U f$$

and proceed similarly to obtain the higher order derivatives $U^{(r)}(x, y, t_0)$ of $U(x, y, t)$.

Although the Taylor series method can give excellent results, the derivatives can be very difficult to calculate and very time-consuming to evaluate. To avoid the differentiation of $f(t, u)$, one can use the Runge-Kutta formulas.

Runge-Kutta methods are closely related to the Taylor series expansion of $U(x, y, t)$ in Equation (2.70), but differentiation of f are not necessary in this method. All Runge-Kutta Methods will be written in the form

$$u_{n+1} = u_n + \Delta t F(t_n, u_n, \Delta t; f) \qquad n \geq 0. \tag{2.71}$$

We want

$$F(t, U(x, y, t), \Delta t; f) \approx \dot{U}(x, y, t) = f(t, U(x, y, t))$$

for all small values of Δt. The truncation error for Equation (2.71) is defined as

$$T_{n+1}(U) = U(x, y, t_{n+1}) - U(x, y, t_n)$$
$$-\Delta t F(t_n, U(x, y, t_n), \Delta t; f) \quad n \geq 0. \qquad (2.72)$$

The derivation of a formula for the truncation error is linked to the derivation of these methods and this will also be true when considering Runge-Kutta Methods of a higher order. The derivation of Runge-Kutta Method will be introduced by deriving a family of second order formulas (Butcher, (2003)). We suppose F has the general form

$$F(t, u, \Delta t; f) = \gamma_1 f(t, u) + \gamma_2 f(t + \alpha\Delta t, u + \alpha\Delta t f(t, u)) \qquad (2.73)$$

in which the three constants $\gamma_1, \gamma_2, \alpha$ are to be determined.

When the Taylor's theorem for functions of two variables is used to expand the second term on the right hand side of Equation (2.73), through the second derivative terms, we obtain

$$F(t, u, \Delta t; f) = \gamma_1 f(t, u) + \gamma_2[f(t, u) + \Delta t(\alpha f_t + \alpha f f_u)$$
$$+(\Delta t)^2(\frac{1}{2}\alpha^2 f_{tt} f + \alpha f_{tu} f + \frac{1}{2}\alpha^2 f^2 f_{uu})] + O((\Delta t)^3). \qquad (2.74)$$

Also, one needs derivatives of $\dot{U}(x, y, t) = f(t, U(x, y, t))$

$$\ddot{U} = f_t + f_u f$$

$$U^{(3)} = f_{tt} + 2f_{tu}f^2 + f_{uu}f^2 + f_u f_t + f_u^2 f \qquad (2.75)$$

for the truncation error,

$$T_{n+1}(U) = U(x,y,t_{n+1}) - U(x,y,t_n) - hF(t_n, U(x,y,t_n), \Delta t; f)$$

$$= \Delta t\dot{U}(x,y,t_n) + \frac{(\Delta t)^2}{2}\ddot{U}(x,y,t_n) + \frac{(\Delta t)^3}{6}U^{(3)}(x,y,t_n)$$
$$+O(\Delta t)^4 - \Delta t F(t_n, U(x,y,t_n), \Delta t; f).$$

Substituting from Equation (2.74) and Equation (2.75) and collocating together, we obtain

$$T_{n+1}(U) = \Delta t[1-\gamma_1-\gamma_2]f + (\Delta t)^2[(\frac{1}{2}-\gamma_2\alpha)f_t + (\frac{1}{2}-\gamma_2\alpha)f_u f]$$
$$+(\Delta t)^3[(\frac{1}{6}-\frac{1}{2}\gamma_2\alpha^2)f_{tt} + (\frac{1}{3}-\gamma_2\alpha^2)f_{tu}f + (\frac{1}{6}-\frac{1}{2}\gamma_2\alpha^2)f_u f_u^2$$
$$+\frac{1}{6}f_u f_t + \frac{1}{6}f_u^2 f] + O((\Delta t)^4) \tag{2.76}$$

where all derivatives are evaluated at (t_n, U_n).

We want to make the truncation error converge to zero as rapidly as possible. The coefficient of $(\Delta t)^3$ cannot be zero in general, if f is allowed to vary arbitrarily. The requirement that the coefficients of Δt and $(\Delta t)^2$ be zero leads to

$$\gamma_1 + \gamma_2 = 1 \qquad \gamma_2\alpha = 1/2 \tag{2.77}$$

We observe that Equation (2.77) reduces to $\gamma_2 = 1/2\alpha$ and $\gamma_1(1-(1/2\alpha))$. That there are infinitely many solutions for the parameters is a peculiarity of Runge-Kutta Methods. For example, one choice is $\alpha = 1/2$, giving the sequence

$$y_{n+1} = y_n + \Delta t f(t_n + \frac{\Delta t}{2}, u_n + \frac{\Delta t}{2}f(t_n, u_n)) \quad n \geq 0$$

which is usually called the modified Euler's Method. Another choice is $\alpha = 2/3$, which leads to second order Runge-Kutta Method

$$y_{n+1} = y_n + \frac{\Delta t}{4}[f(t_n, u_n) + 3f(t_n + \frac{2}{3}\Delta t, u_n + \frac{2}{3}\Delta t f(t_n, u_n))] \quad n \geq 0. \tag{2.78}$$

Higher order formulas can be created, although the algebra becomes very complicated. Assume a formula for $F(t, u, \Delta t; f)$ of the form

$$F(t, u, \Delta t; f) = \sum_{j=1}^{m} \gamma_j K_j$$

$$K_1 = f(t, u)$$

$$K_j = f(t + \alpha_j \Delta t, u + \Delta t \sum_{i=1}^{j-1} \beta_{ji} K_i) \tag{2.79}$$

where $0 \leq \alpha_j \leq 1$ and $\alpha_j = \sum_{i=1}^{j-1} \beta_{ji}$.

These coefficients can be chosen to make the leading terms in the truncation error equal to zero, just as was done with Equation (2.76) and Equation (2.77). Using Equation (2.79), one can obtain m-stage Runge-Kutta Method to solve $\dot{u} = f(t, u)$. Higher order Runge-Kutta Methods are more accurate than lower order, but it is known that m and order k must satisfy $m \geq k$ for an m-stage k*th* order method.

For a given value of m, there are many choices for the weights γ_j in Equation (2.79) and for the parameters α_j and β_{jr}. For $m = 4$, a very popular four-stage, $4th$ order Runge-Kutta Method is

$$u_{n+1} = u_n + \frac{\Delta t}{6}[K_1 + 2K_2 + 2K_3 + K_4] \tag{2.80}$$

where

$$K_1 = f(t_n, u_n)$$

$$K_2 = f(t_n + \frac{1}{2}\Delta t, u_n + \frac{1}{2}\Delta t K_1)$$

$$K_3 = f(t_n + \frac{1}{2}\Delta t, u_n + \frac{1}{2}\Delta t K_2)$$

$$K_4 = f(t_n + \Delta t, u_n + \Delta t K_3).$$

For the transient diffusion equation, considered in this thesis, the DQ discretized set of ordinary differential equations (Equation 2.57)

$$[A]\{u\} + \{s\} = \{b\} = \frac{d\{u\}}{dt} \tag{2.81}$$

are solved now by using $4th$ order RKM (Equation (2.80)). Thus, we can easily write by taking [A]{u}+{s} as the vector function {f(t,{u})} in the sample initial value problem $\dot{u} = f(t, u)$. So,

$$\{f(t, \{u\})\} = [A]\{u\} + \{s\}. \tag{2.82}$$

Thus the fourth order RKM gives for the transient diffusion equation the following vector equation

$$\{u_{n+1}\} = \{u_n\} + \frac{\Delta t}{6}[\{K_1\} + 2\{K_2\} + 2\{K_3\} + \{K_4\}] \tag{2.83}$$

where

$$\{K_1\} = [A]\{u_n\} + \{s\}$$

$$\{K_2\} = [A]\{u_n + \frac{\Delta t}{2}\{K_1\}\} + \{s\}$$

$$\{K_3\} = [A]\{u_n + \frac{\Delta t}{2}\{K_2\}\} + \{s\}$$

$$\{K_4\} = [A]\{u_n + \Delta t\{K_3\}\} + \{s\}.$$

Now, the stability of the numerical scheme (2.83) in applying the system (2.57) has to be considered for obtaining converged numerical solution to the transient diffusion equation.

2.3 Stability of Discretized Differential Quadrature Equations

From the application of PDQ method to the transient diffusion equation we obtained the set of ordinary differential equations (2.57)

$$[A]\{u\} = \{b\} - \{s\}.$$

The stability analysis of this equation is based on the eigenvalue distribution of the DQ discretization matrix [A]. If [A] has eigenvalues λ_i and corres-ponding eigenvector ξ_i, (i=1,2,...,K) K being the size of the matrix [A], the similarity transformation reduces the system of the form

$$\frac{d\{U\}}{dt} = [D]\{U\} + \{S\}. \tag{2.84}$$

Here the diagonal matrix [D] is formed from the eigenvalues and from a nonsingular matrix [P] containing the eigenvectors as columns

$$[D] = [P]^{-1}[A][P] \tag{2.85}$$

$$\{U\} = [P]^{-1}\{u\} \tag{2.86}$$

$$\{S\} = -[P]^{-1}\{s\} \tag{2.87}$$

Since [D] is a diagonal matrix Equation (2.84) is an uncoupled set of ordinary differential equations.

Considering the ith equation of (2.84)

$$\frac{dU_i}{dt} = \lambda_i U_i + S_i \tag{2.88}$$

the solution is given by

$$U_i = (U_i(0) + \frac{S_i}{\lambda_i})e^{\lambda_i t} - \frac{S_i}{\lambda_i}. \tag{2.89}$$

Thus, the solution {u} of the system (2.57) can be obtained as

$$\{u\} = [P]\{U\} = \sum_{i=1}^{K} U_i\xi_i = \sum_{i=1}^{K}[U_i(0)e^{\lambda_i t} + \frac{S_i}{\lambda_i}(e^{\lambda_i t} - 1)]\xi_i. \tag{2.90}$$

Clearly, the stable solution of {u} when $t \to \infty$ requires

$$Re(\lambda_i) < 0 \qquad for\ \ all\ \ i \tag{2.91}$$

where Re(λ_i) denotes the real part of λ_i. This is the stability condition for the system (2.57).

When single step method is applied to the system of ordinary differential equations, it is absolutely stable if

$$|E_i(\lambda_i \Delta t)| < 1 \qquad i = 1, 2, ..., K$$

Where $Re(\lambda_i) < 0$ and $E_i(\lambda_i \Delta t)$ denotes the growth factor for each diagonal element of $E(D\Delta t)$ for approximating the matrix $exp(D\Delta t)$.

Fourth order RKM has the region of absolute stability (Butcher, (2003))

$$-2.78 < \lambda_i \Delta t < 0 \qquad for\ \ real\ \ \lambda_i \tag{2.92}$$

$$0 < |\lambda_i \Delta t| < 2^{(3/2)} \quad for\ \ pure\ \ imaginary\ \ \lambda_i. \tag{2.93}$$

CHAPTER 3

PROBLEMS AND RESULTS

In this Chapter, application of Differential Quadrature Method for Diffusion type problems described in Chapter 2 as Equation (2.1) is given. First four problems are diffusion equations with Dirichlet type boundary conditions and the other two problems are diffusion equation with Dirichlet-Neumann type boundary conditions together with an initial condition for each of the problems considered. Two problems are also added with time dependent boundary conditions and time dependent coefficients of second order derivatives in the equations. Results are presented in terms of potential (solution of diffusion equation) contours in (x, y) domain for certain time levels comparing with the exact solution. Since the fourth order Runge-Kutta Method used in this thesis is an explicit procedure, choosing the right time step Δt for each problem to obtain stable solution is an important feature of this study. Use of Chebyshev-Gauss-Lobatto grid distribution guaranties the stability condition for the resulting system of PDQ ordinary differential equations giving the eigenvalues of the weighting coefficient matrix as real and negative. Choices of the time step Δt and grid steps Δx, Δy are made according to the absolute stability region of the fourth order RKM. Comparison for different numbers of grid points is also carried. Computer programs are written in Fortran Language and run in PC platform. All the graphics are obtained by using MATLAB package.

Problem 1

The Dirichlet problem is defined as

$$\frac{\partial\ u(x,y,t)}{\partial t} - \nabla^2 u(x,y,t) = 0 \qquad (x,y) \in (-1,1) \tag{3.1}$$

with the initial condition

$$u(x,y,0) = u_0(x,y) = 1 \tag{3.2}$$

and the Dirichlet boundary conditions are given by

$$u(x,y,t) = 0 \qquad x,y = (-1,1). \tag{3.3}$$

The analytical solution is (Tanaka and Chen (2001))

$$u(x,y,t) = v(x,t)v(y,t) \tag{3.4}$$

where

$$v(z,t) = \frac{4}{\pi}\sum_{i=0}^{\infty}\frac{(-1)^i}{2i+1}\cos\left(\frac{2i+1}{2}\pi z\right)e^{\left[-(2i+1)^2\frac{\pi^2 t}{4}\right]}. \tag{3.5}$$

This problem is solved by using several numbers of grid points and compared with the analytical solution for finding the suitable number of grid points. We employed $N = 5$, $N = 7$, $N = 13$ and $N = 14$ Chebyshev-Gauss-Lobatto grid points in one direction in the Differential Quadrature Method for comparing with the analytical solution. The number of grid points N_x and N_y respectively for x and y directions are taken as equal to N and grid step sizes $\Delta x = \Delta y = h$.

Since we use Chebyshev-Gauss-Lobatto grid points the resulting PDQ set of ordinary differential equations (Equation 2.57)) are stable in the sense that all the eigenvalues of weighting coefficients matrix [A] are real and negative for both Dirichlet and Neumann boundary conditions.

Also absolute stability region for the fourth order RKM is $-2.78 < \lambda_i \Delta t < 0$ i=1,2,...K where K is the size of the matrix [A]. Therefore choice of h and Δt play an important role for the stability of the overall numerical scheme employed for solving transient diffusion equation. While the refinement of h improves the results for PDQ part (discretization of Laplace operator part), Δt in RKM must also be decreased since the eigenvalues are getting large in magnitude for small h.

For the number of grid points $N = 5, 7$, the time step Δt was taken to be 0.01 first and the approximated solution agrees with the exact solution at the time level $t = 1.0$ as can be seen from Figures 3.1 and 3.2. When the number of grid points N is increased (h is decreased) to get smoother contours of the potential at the same time level, we need smaller time steps. Thus for $N = 9, 10$ and 13, we have taken $\Delta t = 0.001$ since the maximum eigenvalue in magnitude is about 20×10^3 for N=10 (Shu,(2000)) and obtained very well agreement with the exact solution. Figures 3.3, 3.4, 3.5 show potential values at $t = 1.0$ from both DQ Method and exact solution. Figure 3.6 represents both solutions at the time level $t = 2.0$ and Figure 3.7 shows the very well agreement of the potential with the exact solution for several values of Δt. Smaller Δt values provide the same accuracy but since it requires more computational time, $\Delta t = 0.001$ or $\Delta t = 0.0001$ values are suitable for most of the Dirichlet problems.

Since the solution obtained with $N = 13$ is already very accurate there is no need for this problem to increase N. Increasing N will increase the size of the coefficient matrix and this will be costly in terms of computation since in the Runge-Kutta Method we have matrix vector multiplications four times in each time step.

For the coming problems the number of points in one direction is going to be taken around $N = 13$ since it is a suitable number of grid points in one direction for obtaining accurate results and for drawing the contours. Time step size Δt is going to be taken as 10^{-3} or 10^{-4} which gives $-2.78 < \lambda_i \Delta t < 0$ for some.

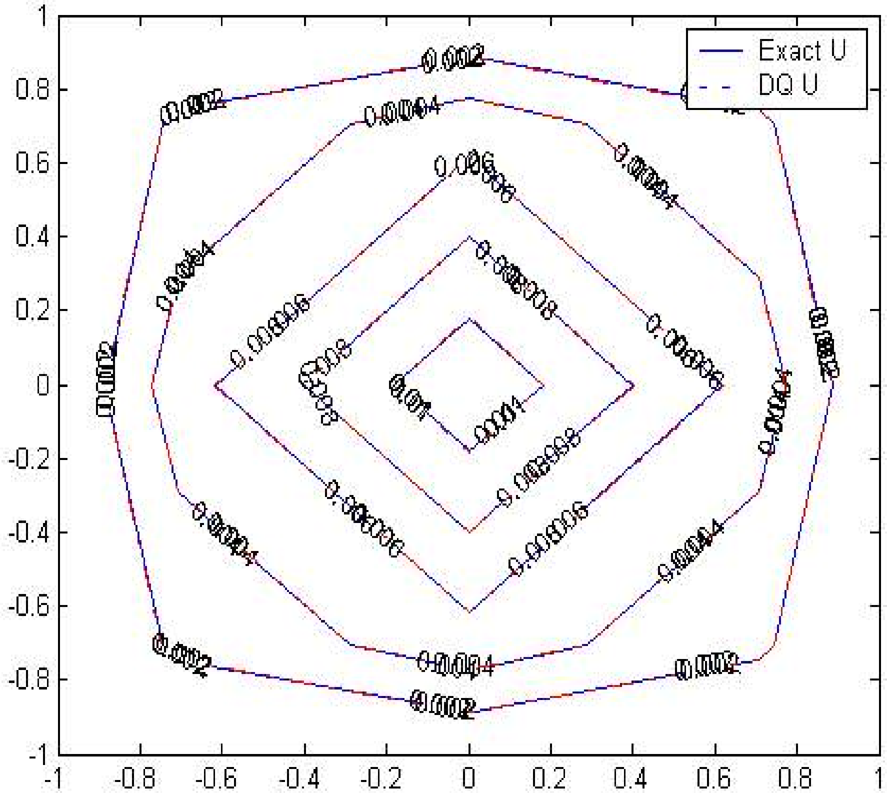


Figure 3.1: At t=1.0 with $N = 5$ grid points, $\Delta t = 0.01$.

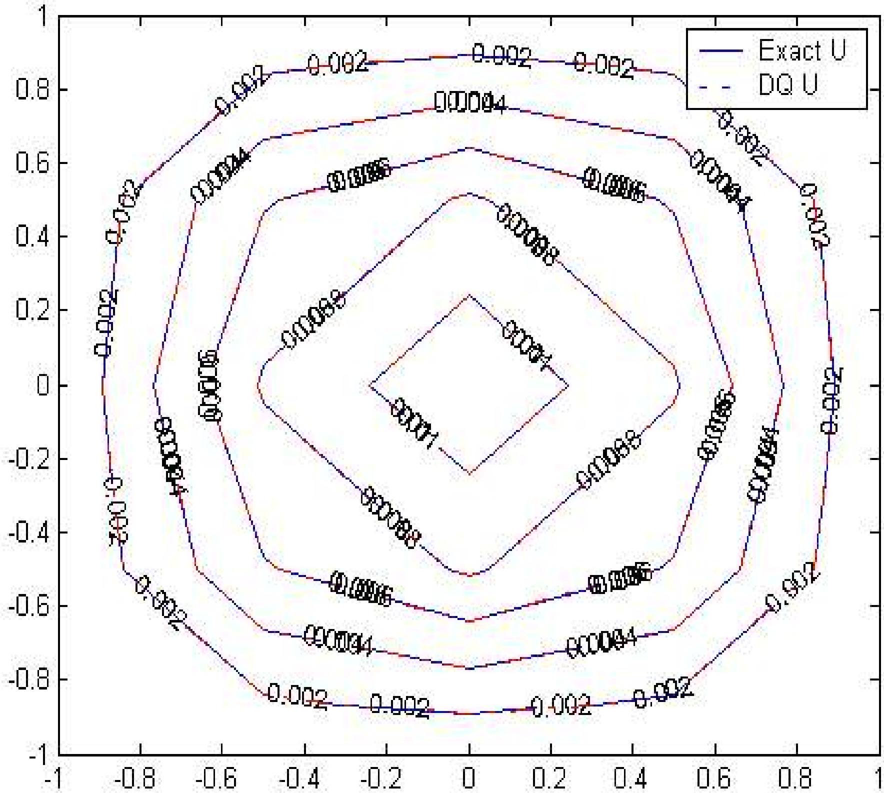


Figure 3.2: At t=1.0 with $N = 7$ grid points,$\Delta t = 0.01$.

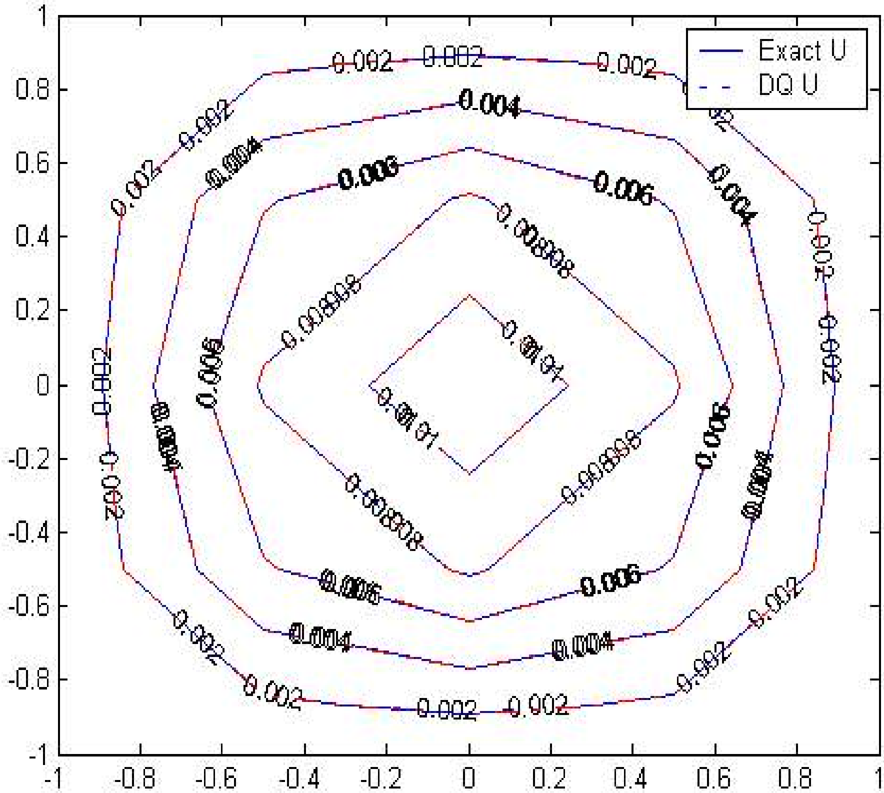


Figure 3.3: At t=1.0 with $N = 7$ grid points, $\Delta t = 0.001$.

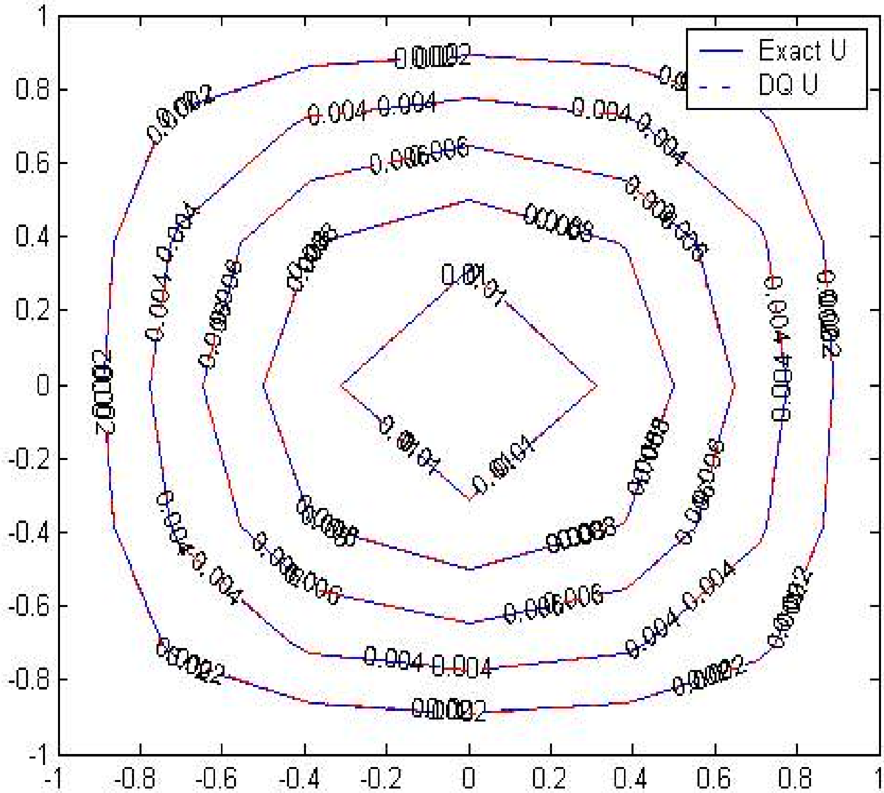
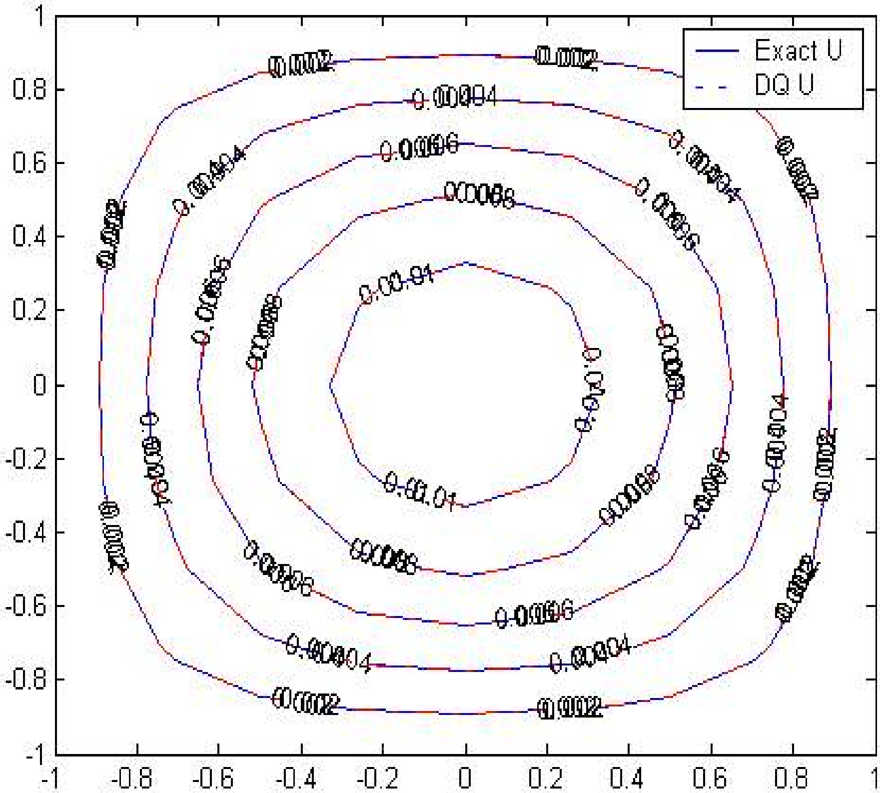


Figure 3.4: At t=1.0 with $N = 9$ grid points, $\Delta t = 0.001$.



,

Figure 3.5: At t=1.0 with $N = 13$ grid points, $\Delta t = 0.001$

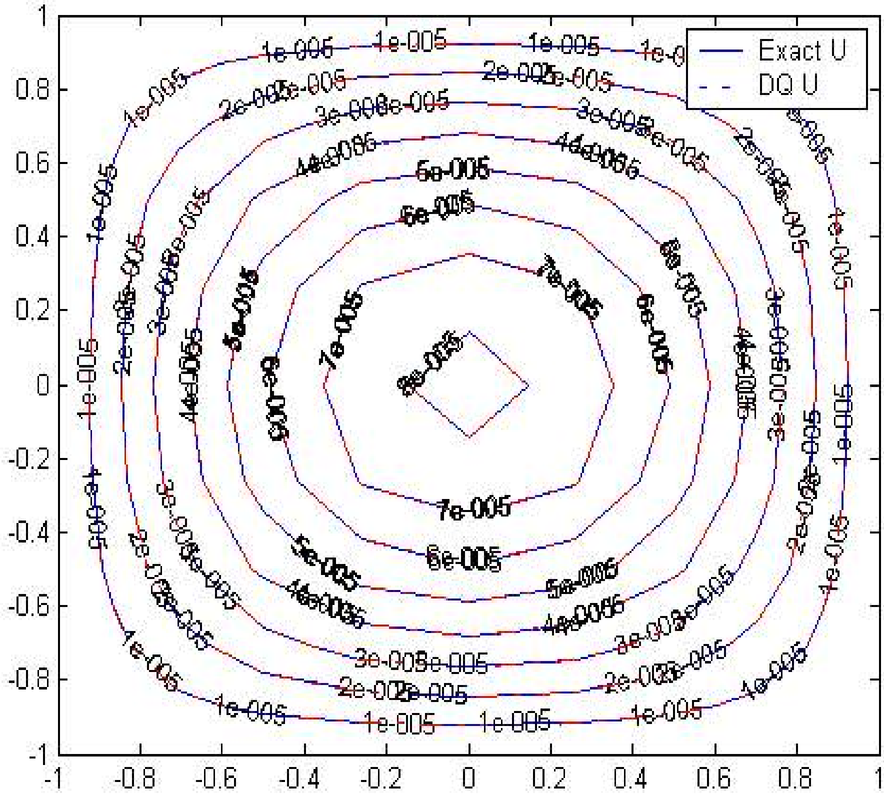


Figure 3.6: At t=2.0 with $N = 13$ grid points, $\Delta t = 0.001$

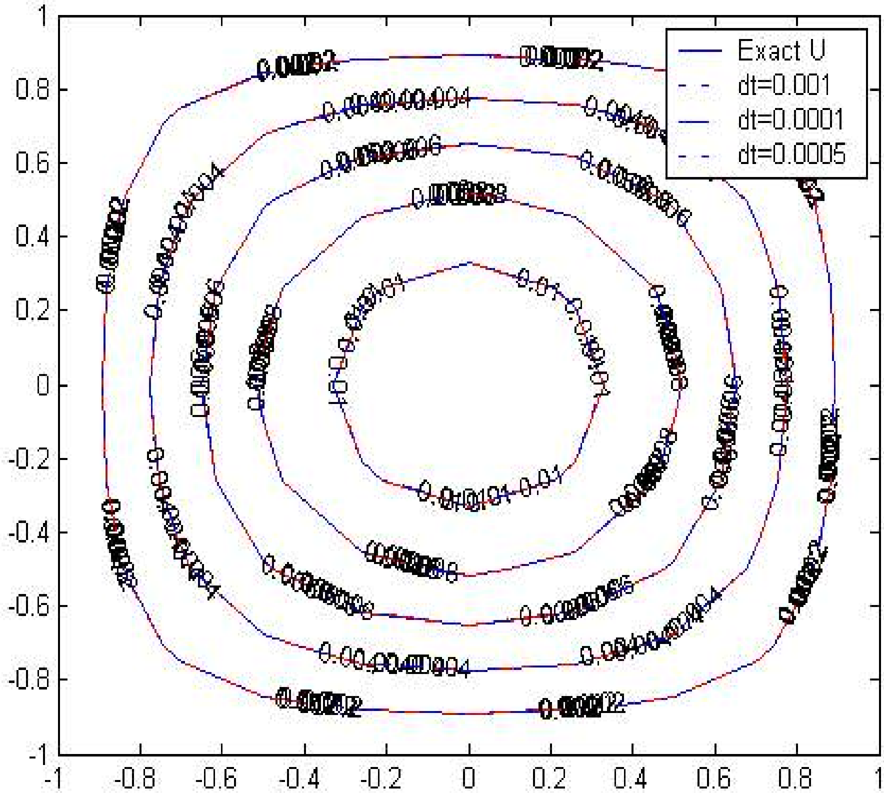


Figure 3.7: At t=1.0 with $N = 13$ grid points

Problem 2

The Dirichlet problem is defined as

$$\frac{\partial\ u(x,y,t)}{\partial t} - \nabla^2 u(x,y,t) = 0 \qquad (x,y) \in (0,1) \tag{3.6}$$

with the initial condition

$$u(x,y,0) = u_0(x,y) = \sin(\pi x)\sin(2\pi y) \tag{3.7}$$

and the Dirichlet boundary conditions are given by

$$u(x,y,t) = 0 \qquad x,y = 0,1. \tag{3.8}$$

Exact solution is given by the equation

$$u(x,y,t) = e^{-5\pi^2 t}\sin(\pi x)\sin(2\pi y). \tag{3.9}$$

This Dirichlet problem differs from the Dirichlet problem 1 only in the initial condition which is a function of x and y. The region of the problem is still a square $0 \leq x \leq 1$, $0 \leq y \leq 1$. In this problem $N = 13$ and $\Delta t = 0.0001$ were used for drawing potential contours at the time levels $t = 0.01$ and $t = 0.1$ in Figures 3.8 and 3.9. Figure 3.10 gives much smoother potential contours at $t = 0.1$ with $\Delta t = 0.0001$ by using $N = 17$ which is expected since more points are involved for both PDQ computations and drawing graphs.

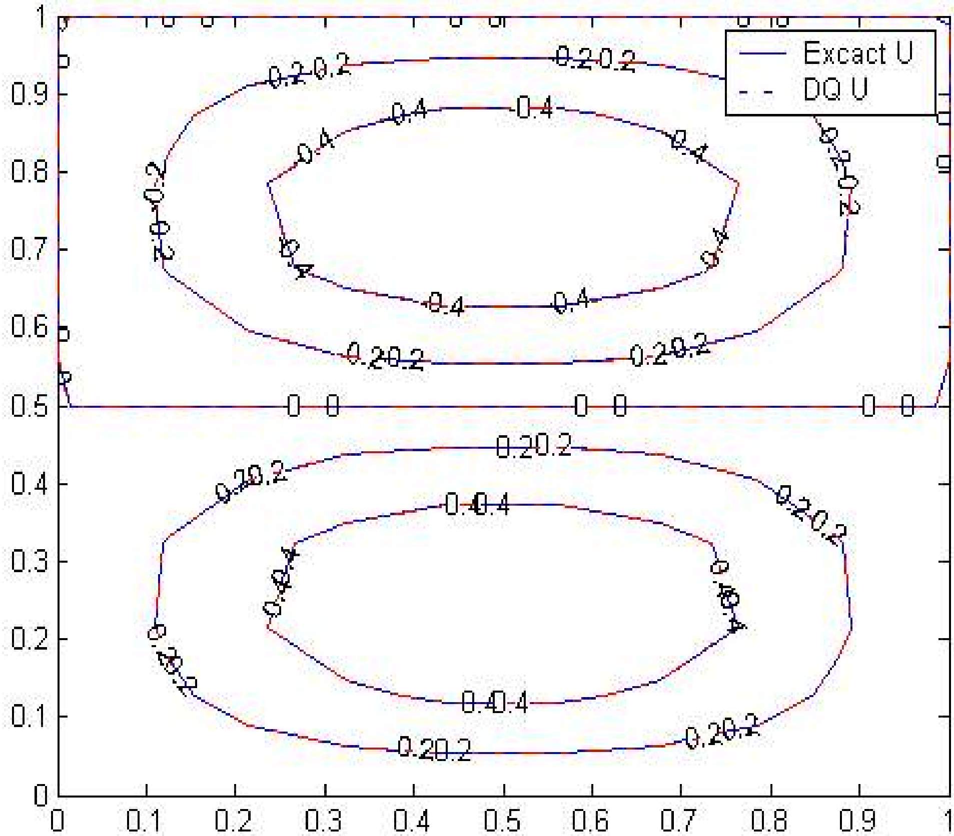


Figure 3.8: At t=0.01 with $N = 13$ grid points, $\Delta t = 0.0001$

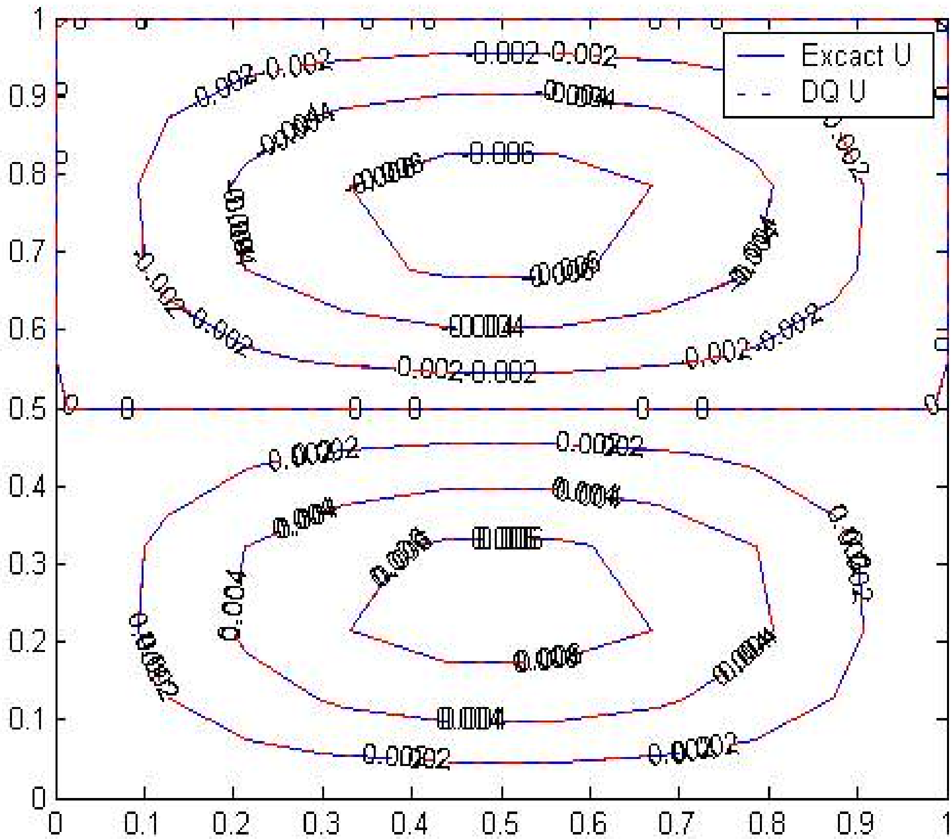


Figure 3.9: At t=0.1 with $N = 13$ grid points, $\Delta t = 0.0001$

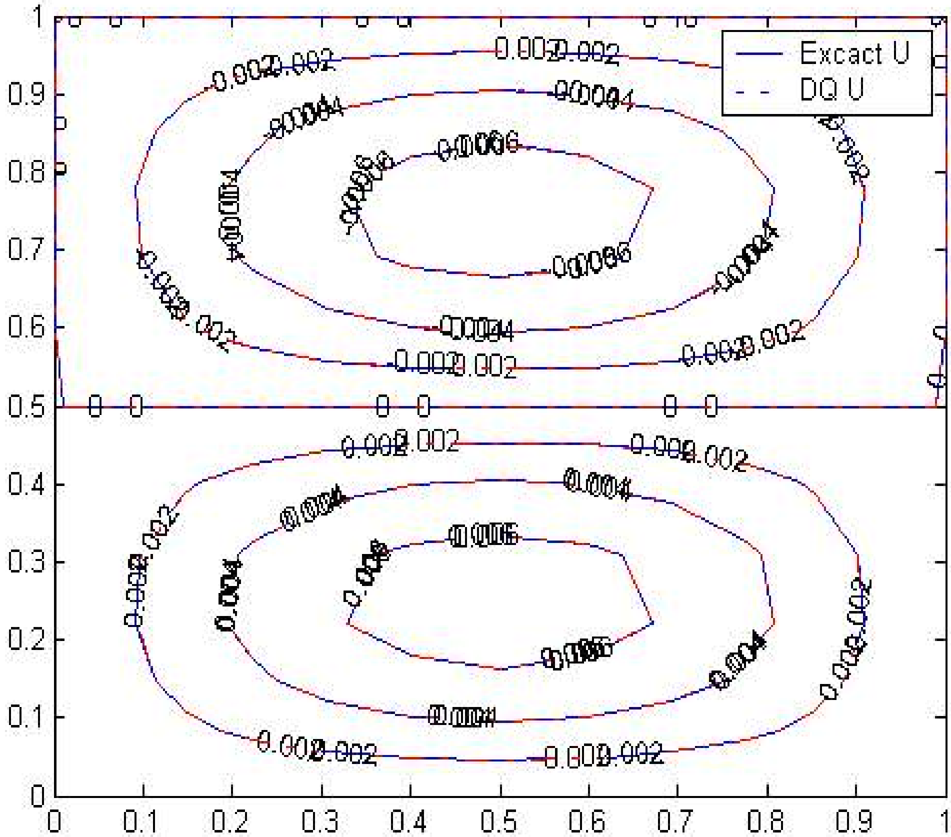


Figure 3.10: At t=0.1 with $N = 17$ grid points, $\Delta t = 0.0001$

Problem 3

The Dirichlet problem considered is

$$\frac{\partial\, u(x,y,t)}{\partial t} - \nabla^2 u(x,y,t) = 0 \qquad (0 < x, y < 1; t > 0) \tag{3.10}$$

with the initial condition

$$u(x,y,0) = u_0(x,y) = 20 + 80\Big[y - \sin(\frac{\pi x}{2})\sin(\frac{\pi y}{2})\Big] \qquad (0 \leq x, y \leq 1) \tag{3.11}$$

subject to the boundary conditions

$$u(x,0,t) = 20, \quad u(x,1,t) = 20 + 80\Big[y - e^{-0.5\pi^2 t}\sin(0.5\pi x)\Big] \tag{3.12}$$

$$u(0,y,t) = 20 + 80y, \quad u(1,y,t) = 20 + 80\Big[y - e^{-0.5\pi^2 t}\sin(0.5\pi y)\Big]. \tag{3.13}$$

The analytical solution is given by

$$u(x,y,t) = 20 + 80\Big[y - e^{-0.5\pi^2 t}\sin(0.5\pi x)\sin(0.5\pi y)\Big]. \tag{3.14}$$

In this Dirichlet problem, both initial and boundary conditions are functions of x and y. Results are obtained with $N = 13$ and $\Delta t = 0.0001$ at the time levels $t = 0.1$, $t = 0.5$, $t = 1.0$ and $t = 10$ as presented in Figures 3.11, 3.12, 3.13 and 3.14 respectively. As can be seen from these Figures the steady-state case is obtained after the time level $t = 0.5$ giving exactly the same contours.

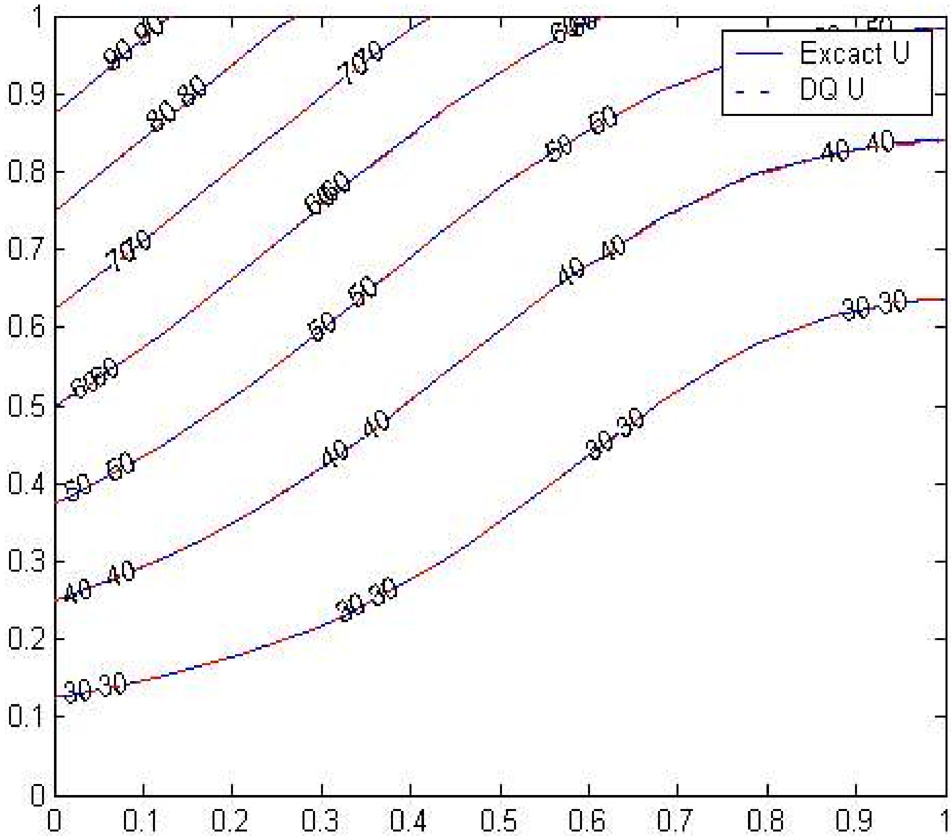


Figure 3.11: At t=0.1 with $N = 13$ grid points, $\Delta t = 0.0001$

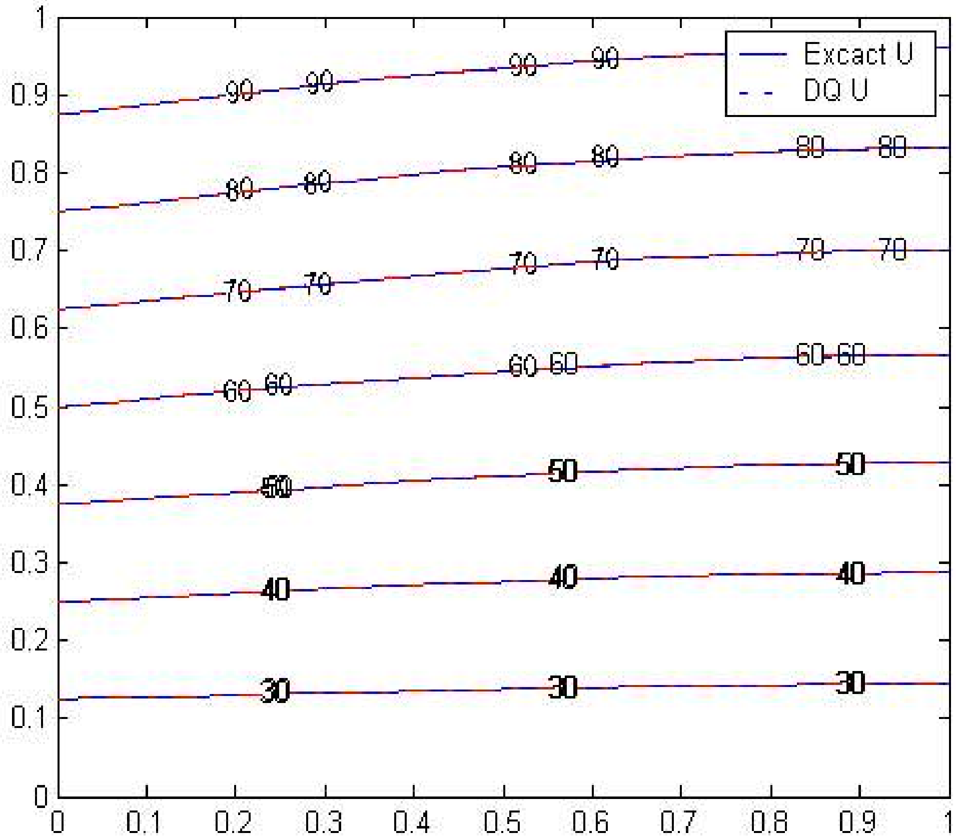


Figure 3.12: At t=0.5 with $N = 13$ grid points, $\Delta t = 0.0001$

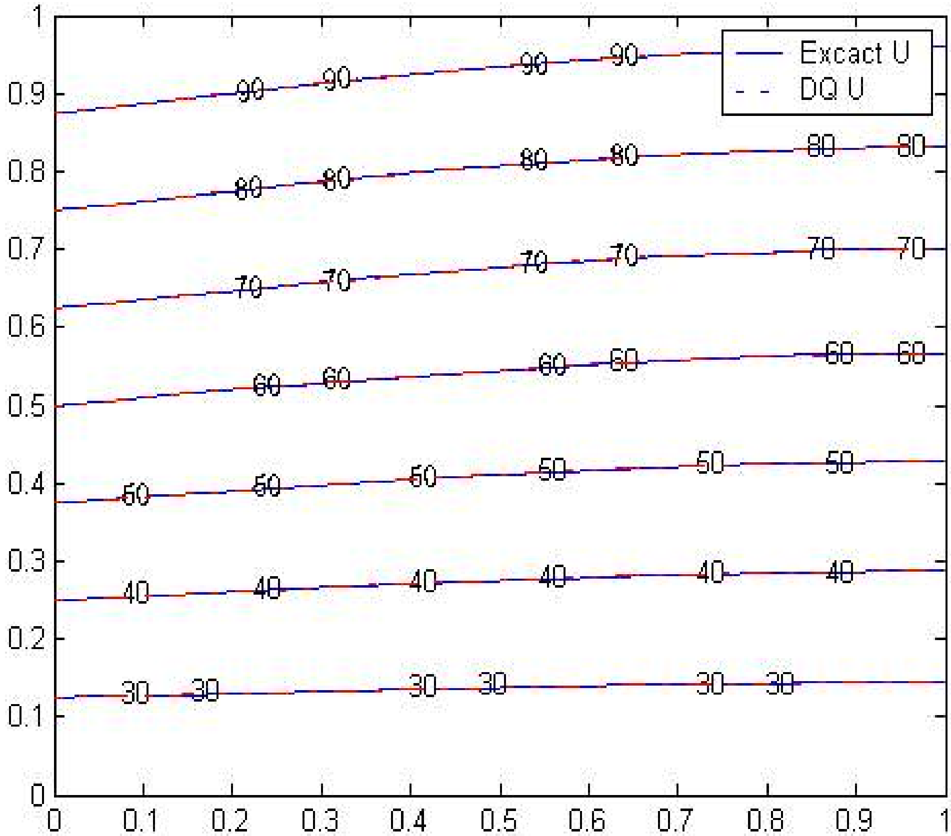


Figure 3.13: At t=1.0 with $N = 13$ grid points, $\Delta t = 0.0001$

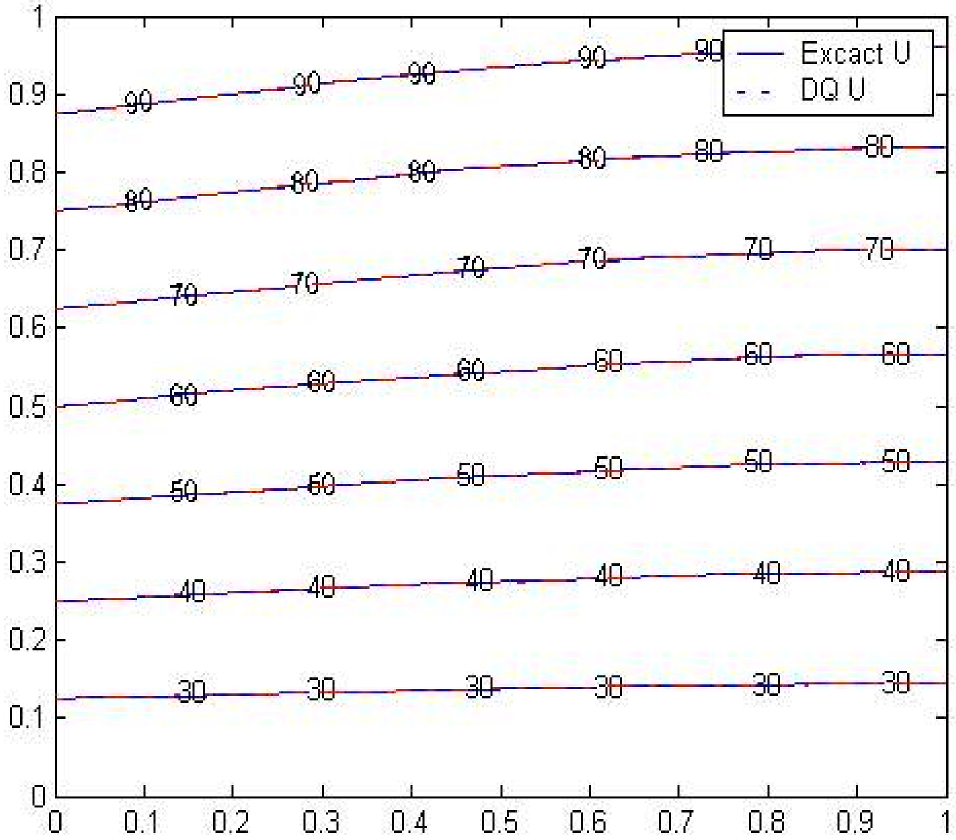


Figure 3.14: At t=10 with $N = 13$ grid points, $\Delta t = 0.0001$

Problem 4

The last Dirichlet problem is defined as

$$\frac{\partial\, u(x,y,t)}{\partial t} - \nabla^2 u(x,y,t) = 0 \qquad (0 < x, y < \pi; t > 0) \tag{3.15}$$

with the initial condition

$$u(x,y,0) = u_0(x,y) = \sin(yx\pi) \qquad (0 \leq x, y \leq \pi) \tag{3.16}$$

subject to the boundary conditions

$$u(x,0,t) = u(x,\pi,t) = 0 \qquad (0 \leq x \leq \pi) \tag{3.17}$$

$$u(0,y,t) = 0 \qquad (0 \leq y \leq \pi) \tag{3.18}$$

and

$$u(\pi,y,t) = \sin y \qquad (0 \leq y \leq \pi). \tag{3.19}$$

The analytical solution is given by

$$u(x,y,t) = \frac{\sinh x}{\sinh \pi} \sinh y + \frac{2}{\pi} \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k(1+k^2)^{1/2}} e^{-(1+k^2)^{1/2} t} \sin kx \sin y. \tag{3.20}$$

In this problem, initial and some of the boundary conditions are also functions of x and y as in the previous problem. The region of the problem is a square $0 \leq x \leq \pi$, $0 \leq y \leq \pi$. The potential contours are obtained by using $N = 13$ and $\Delta t = 0.001$ at the time levels $t = 0.1$ and $t = 0.01$ in Figures 3.15 and 3.16. It can be seen that the Figures show the very well agreement of the approximate solution (differential quadrature solution) with the exact solution.

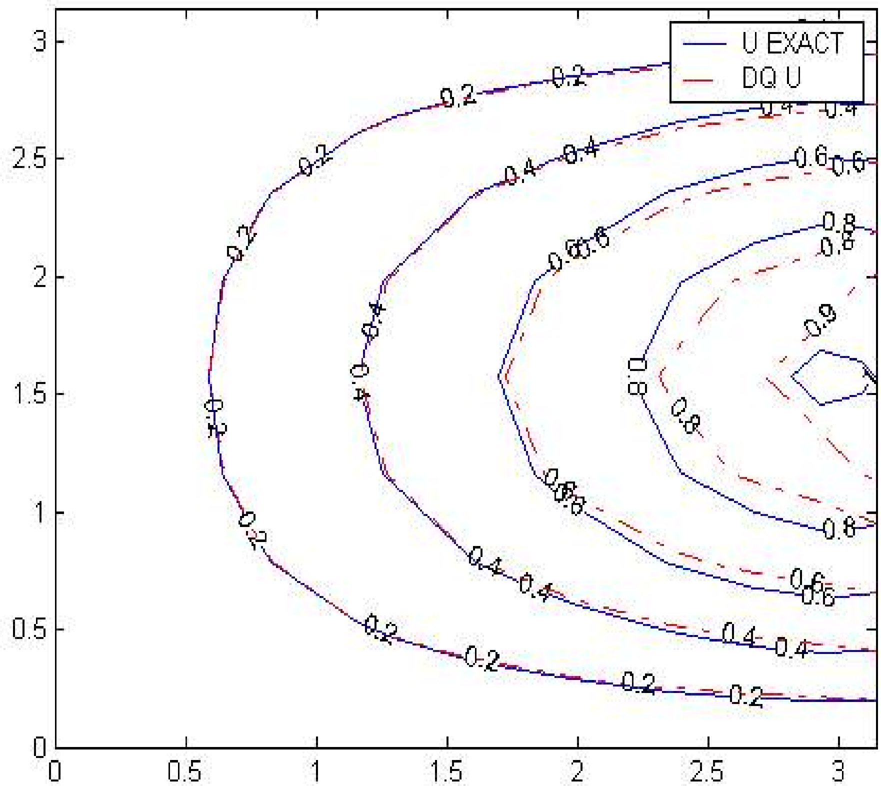


Figure 3.15: At t=0.1 with $N = 13$ grid points, $\Delta t = 0.001$

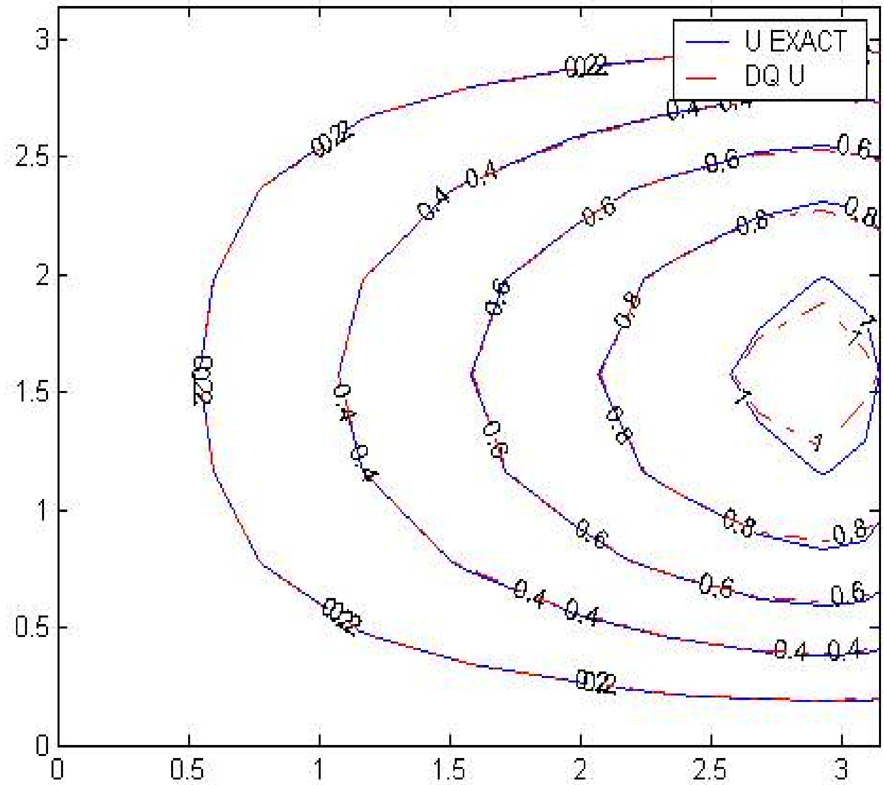


Figure 3.16: At t=0.01 with $N = 13$ grid points, $\Delta t = 0.001$

The following two problems are Dirichlet-Neumann problems that on some boundaries normal derivative q of the potential u is given and on the others still Dirichlet conditions are specified.

Problem 5

This case is a reduced form of the Dirichlet problem 1 and thus holds the same analytical solution formulas. The problem is a Dirichlet-Neumann problem

$$\frac{\partial\, u(x,y,t)}{\partial t} - \nabla^2 u(x,y,t) = 0 \qquad\qquad (x,y) \in (0,1) \tag{3.21}$$

with the initial condition

$$u(x,y,0) = u_0(x,y) = 1. \tag{3.22}$$

The Dirichlet boundary conditions are given by

$$u(1,y,t) = 0, \quad u(x,1,t) = 0 \tag{3.23}$$

and the Neumann boundary conditions are specified as

$$q(0,y,t) = 0, \quad q(x,0,t) = 0. \tag{3.24}$$

The analytical solution is given by the equations (3.4), (3.5) in problem 1 as

$$u(x,y,t) = v(x,t)v(y,t) \tag{3.25}$$

where

$$v(z,t) = \frac{4}{\pi}\sum_{i=0}^{\infty}\frac{(-1)^i}{2i+1}\cos\left(\frac{2i+1}{2}\pi z\right)e^{\left[-(2i+1)^2\frac{\pi^2 t}{4}\right]}. \tag{3.26}$$

For Dirichlet-Neumann problem defined in the region $(x, y)\epsilon[0, 1] \times [0, 1]$ has two homogenous Dirichlet and two homogenous Neumann type boundary conditions with unity initial condition.

For this problem Neumann conditions are also discretized to the final system of equations. The number of discretization points $N = 10$ was suitable for obtaining potential contours. Figures 3.17 gives potential at $t = 0.1$ with $\Delta t = 0.001$ obtained by using $N = 13$ grid points. Figures 3.18 and 3.19 present solution of the problem at the time level $t = 0.1$ with $\Delta t = 0.001$ and $\Delta t = 0.0001$ respectively and by using $N = 10$. As can be seen smaller Δt gives better accuracy for this Dirichlet-Neumann problem. This may be due to the behaviour of mixed boundary conditions since the maximum eigenvalue in magnitude also vary with varying h. We continue with $N = 10$ grid points since it gives almost the same accuracy with $N = 13$ for the other computations in solving Dirichlet-Neumann diffusion problems for decreasing computational cost.

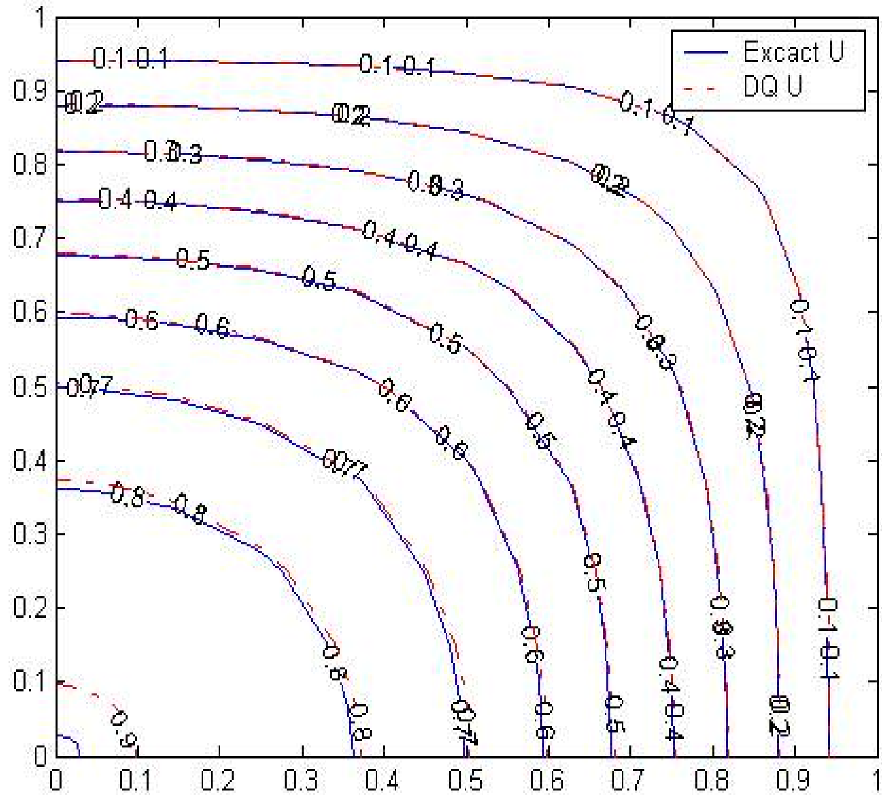


Figure 3.17: At t=0.1 with $N = 13$ grid points, $\Delta t = 0.001$

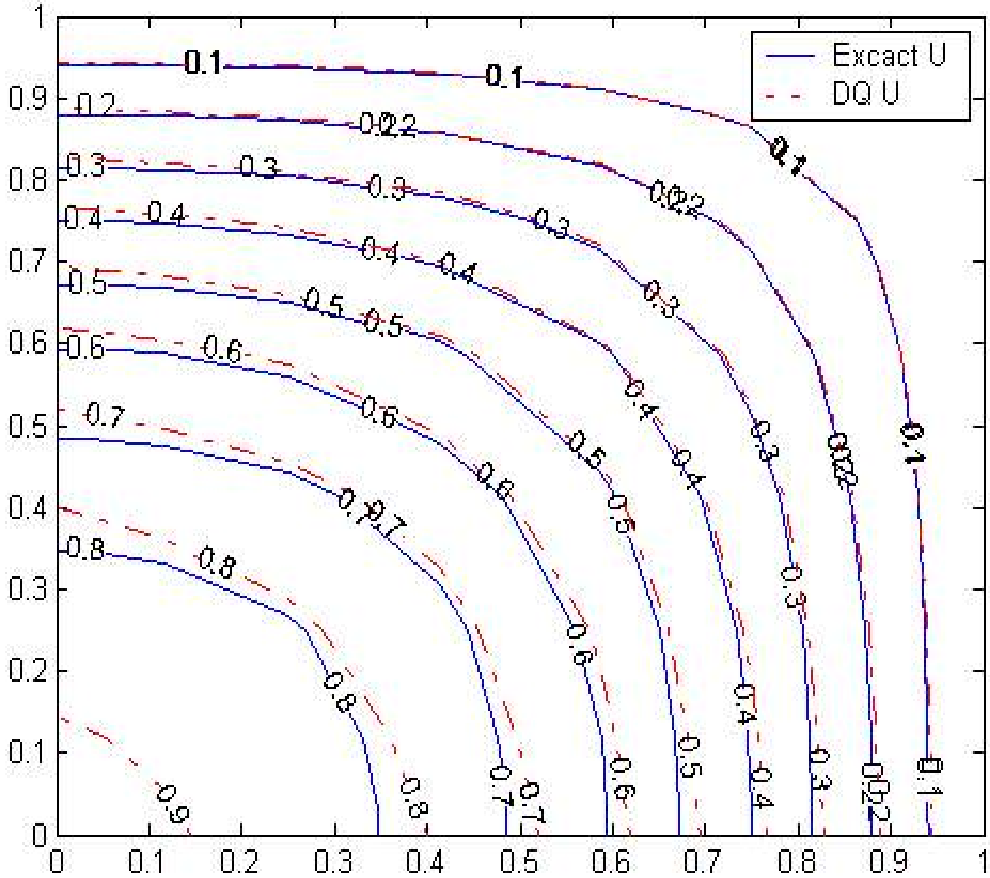


Figure 3.18: At t=0.1 with $N = 10$ grid points, $\Delta t = 0.001$

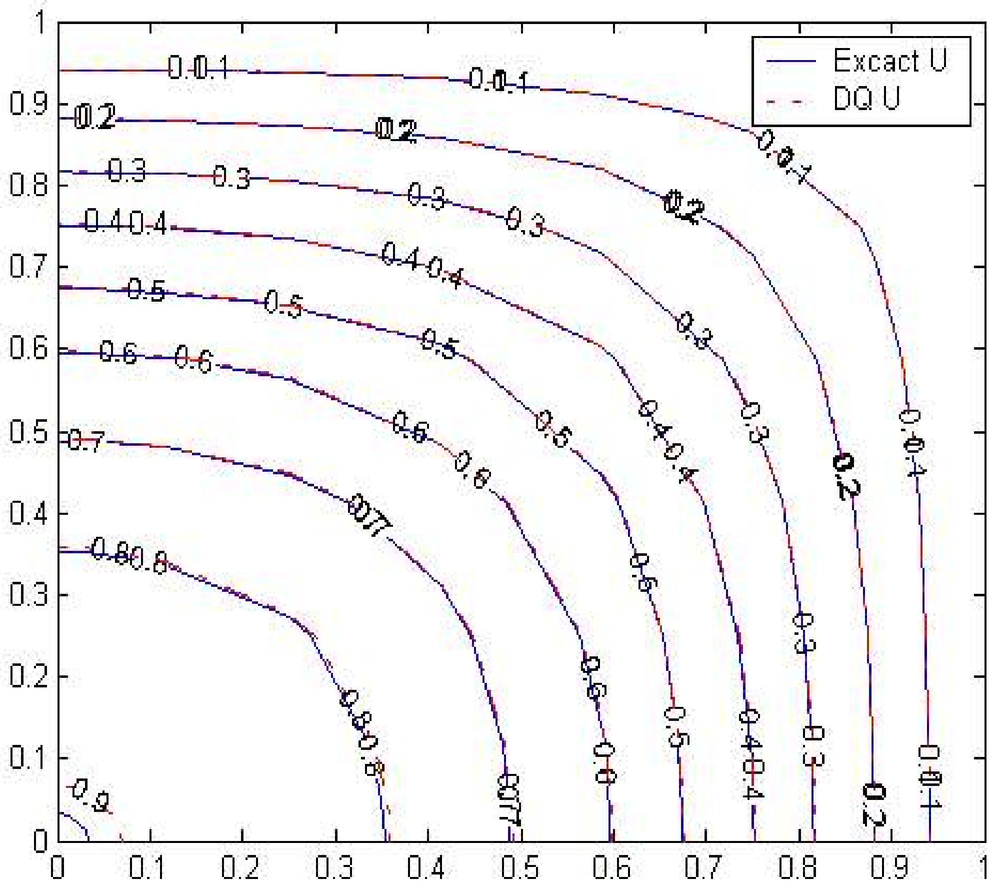


Figure 3.19: At t=0.1 with $N = 10$ grid points, $\Delta t = 0.0001$

Problem 6

The Dirichlet-Neumann problem considered is

$$\frac{\partial\, u(x,y,t)}{\partial t} - \nabla^2 u(x,y,t) = 0 \qquad (0 < x, y < 1; t > 0) \tag{3.27}$$

with the initial condition

$$u(x,y,0) = u_0(x,y) = 20 + 80\Big[y - \sin(\frac{\pi x}{2})\sin(\frac{\pi y}{2})\Big] \qquad (0 \le x, y \le 1) \tag{3.28}$$

subject to the boundary conditions

$$u(x,0,t) = 20 \qquad q(x,1,t) = 80 \tag{3.29}$$

$$u(0,y,t) = 20 + 80y \qquad q(1,y,t) = 0. \tag{3.30}$$

The theoretical solution is given by the equation

$$u(x,y,t) = 20 + 80\Big[y - e^{-0.5\pi^2 t}\sin(0.5\pi x)\sin(0.5\pi y)\Big]. \tag{3.31}$$

This problem is the modified form of problem 3 in boundary conditions. Two of the Dirichlet type boundary conditions are converted to Neumann conditions. In this way normal derivatives on the respective boundaries are included to the final system of equations through PDQ formulations. As expected the same potential contours are obtained as in problem 3 at the time levels $t = 0.1$, $t = 0.5$, $t = 1.0$ and $t = 10$ respectively in Figures 3.20, 3.21, 3.22, 3.23 with $\Delta t = 0.0001$ but by using $N = 10$. Around with $t = 0.5$ the steady state case is reached.

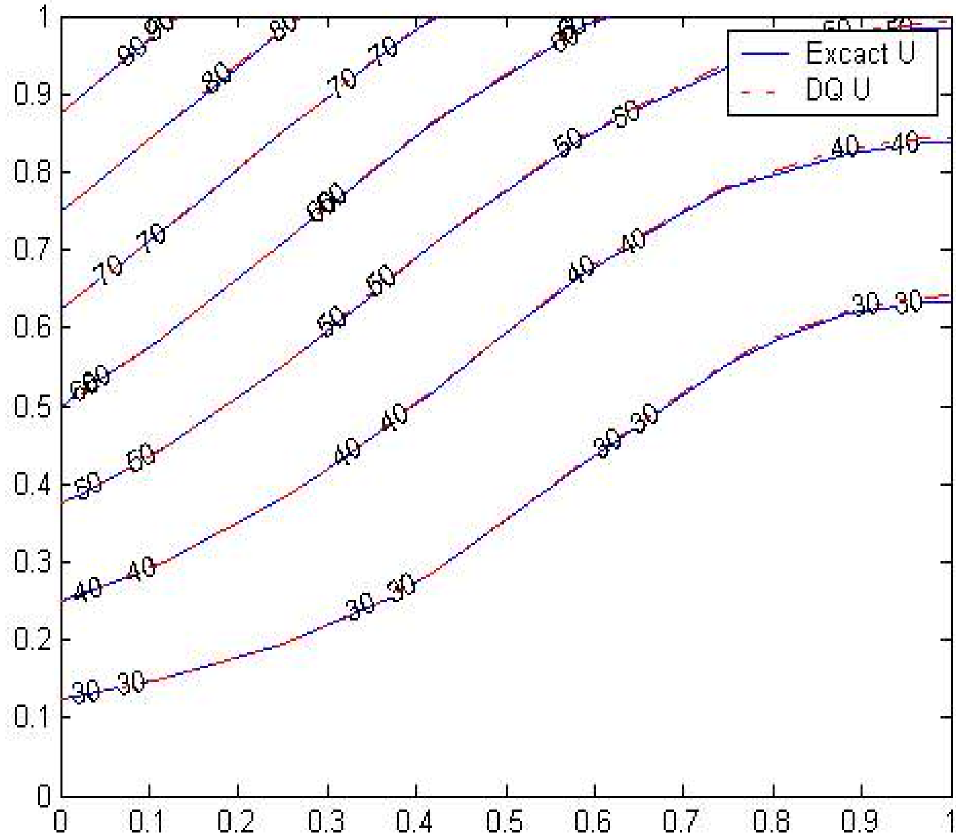


Figure 3.20: At t=0.1 with $N = 10$ grid points, $\Delta t = 0.0001$

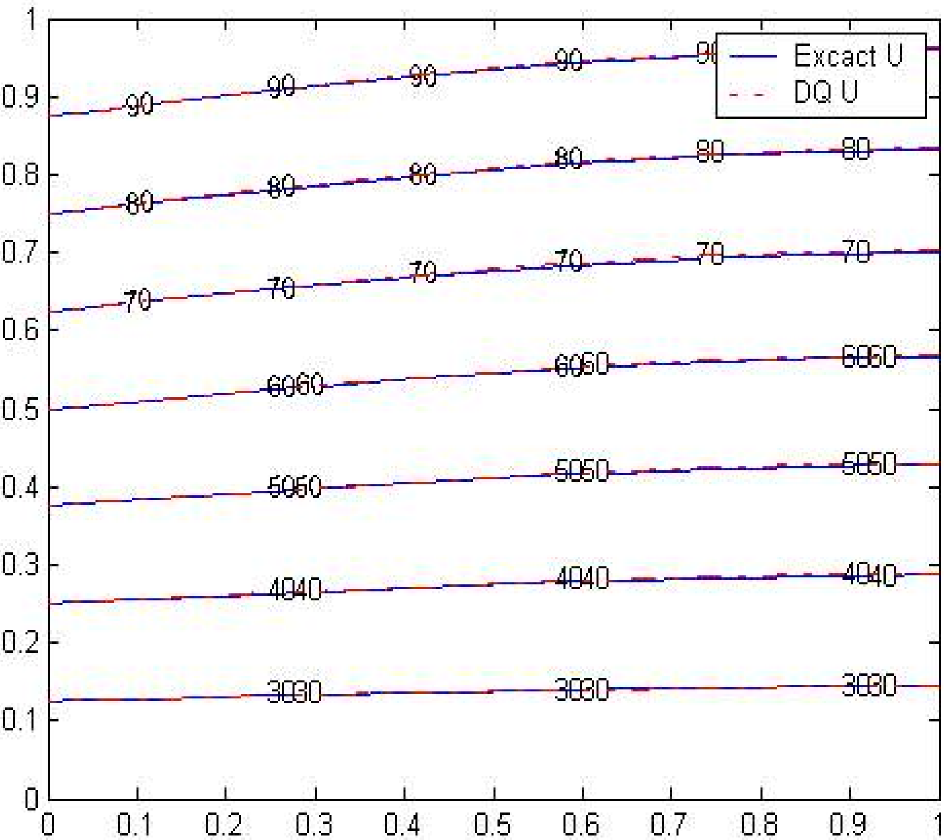


Figure 3.21: At t=0.5 with $N = 10$ grid points, $\Delta t = 0.0001$

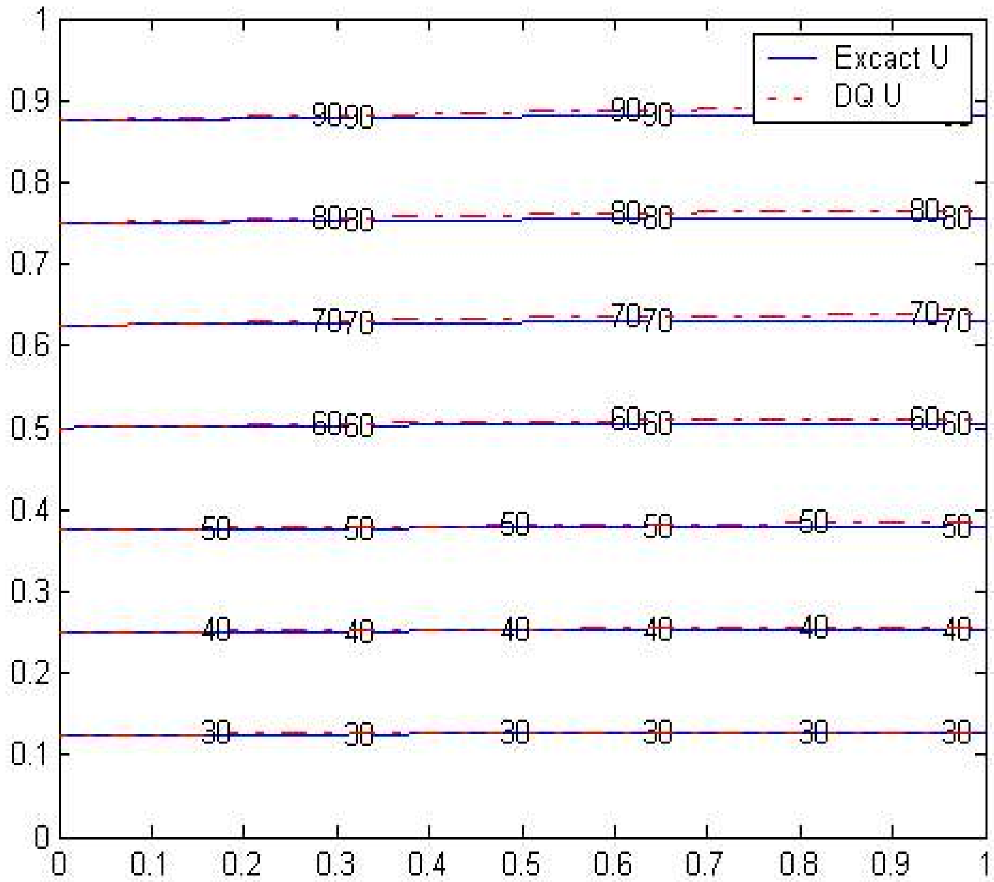


Figure 3.22: At t=1.0 with $N = 10$ grid points, $\Delta t = 0.001$

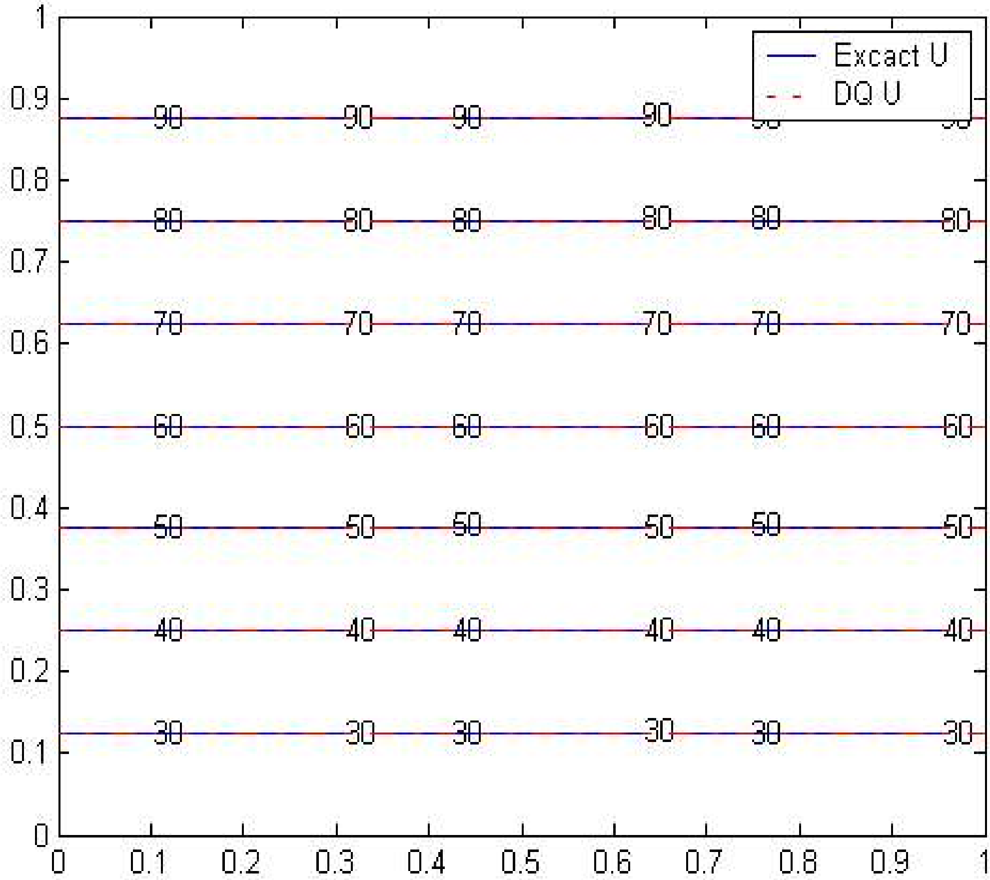


Figure 3.23: At t=10 with $N = 10$ grid points, $\Delta t = 0.0001$

Problems 7 and 8 are different in nature from the previous Dirichlet type problems. In these problems boundary conditions are not only functions of x and y but also the time t. Also second order derivatives have coefficients containing space variables in Problem 7 and space together with time variables in Problem 8.

In these problems $N = 13$ is used as in the other Dirichlet problems.

Problem 7

The Dirichlet problem is defined as

$$\frac{\partial u(x,y,t)}{\partial t} = \frac{1}{4}(1-x^2)\frac{\partial^2 u}{\partial x^2} + \frac{1}{4}(1-y^2)\frac{\partial^2 u}{\partial y^2} \qquad 0 \le x, y \le 0.9, t > 0 \quad (3.32)$$

with the initial condition

$$u(x,y,0) = (1-x^2)(1-y^2) \quad (3.33)$$

and the Dirichlet boundary conditions are given by

$$u(x,0,t) = (1-x^2)e^{(-t)} \qquad u(x,0.9,t) = 0.19e^{(-t)}(1-x^2) \quad (3.34)$$

$$u(0,y,t) = (1-y^2)e^{(-t)} \qquad u(0.9,y,t) = 0.19e^{(-t)}(1-y^2). \quad (3.35)$$

The analytical solution is given by

$$u(x,y,t) = (1-x^2)(1-y^2)e^{(-t)}. \quad (3.36)$$

Potential contours are given in Figures 3.24 and 3.25 at the time level $t = 1.0$ by using $N = 13$, $\Delta t = 0.001$ and $\Delta t = 0.0001$ respectively. Agreement with the exact solution is perfect for this problem. In Figure 3.26 behavior of potential is presented at $t = 0.1$ with the same parameters $N = 13$ and $\Delta t = 0.0001$.

One can see that time dependent boundary conditions are easily inserted in DQM.

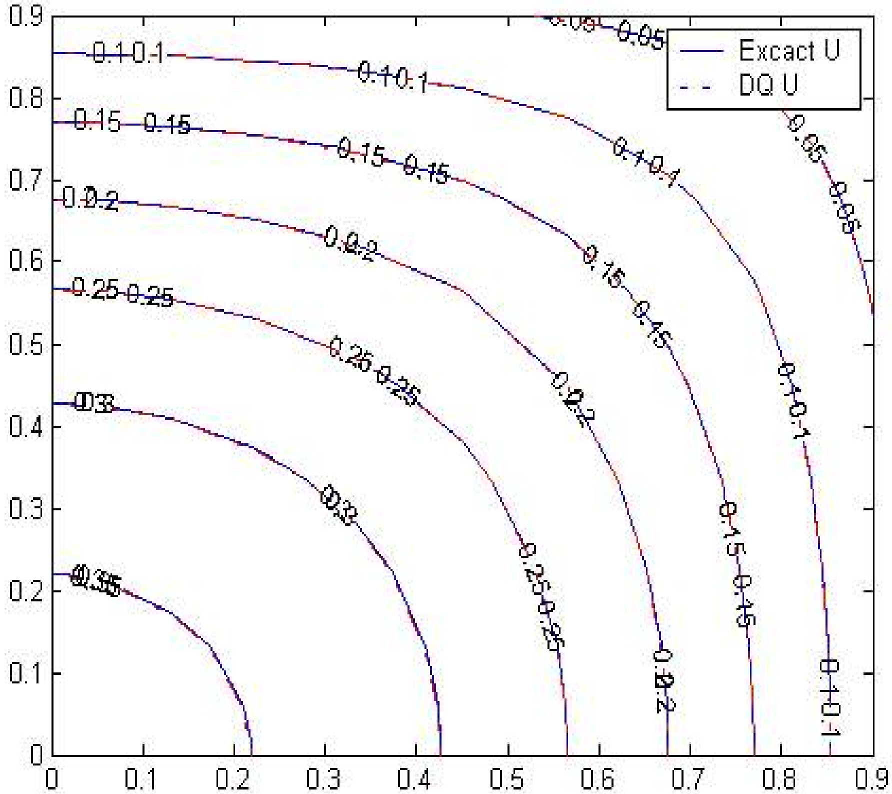


Figure 3.24: At t=1.0 with $N = 13$ grid points, $\Delta t = 0.001$

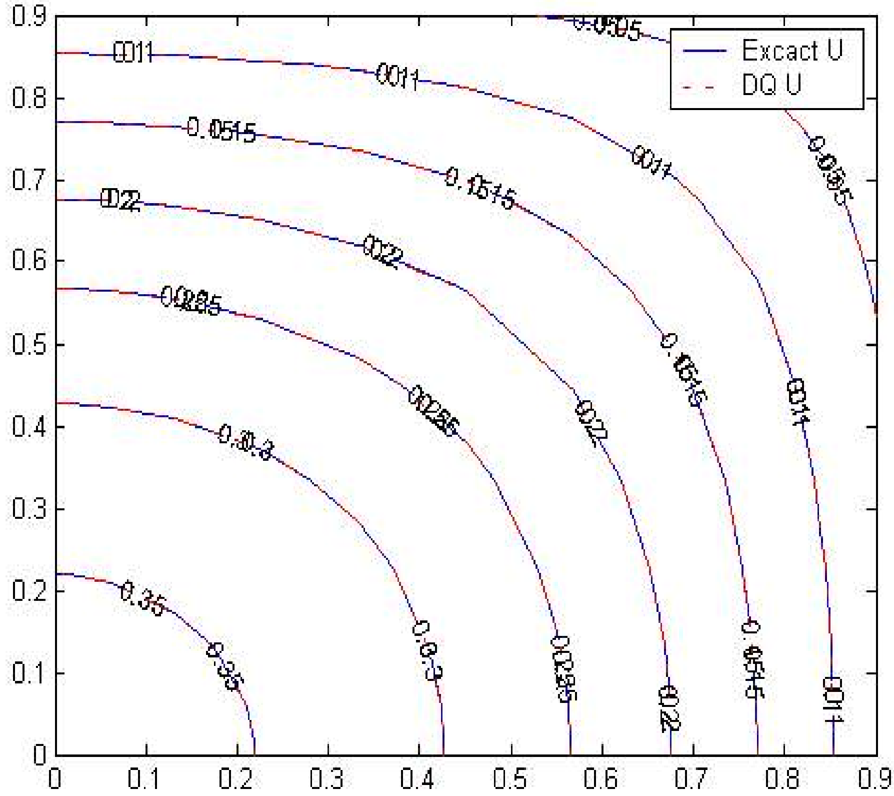


Figure 3.25: At t=1.0 with $N = 13$ grid points, $\Delta t = 0.0001$

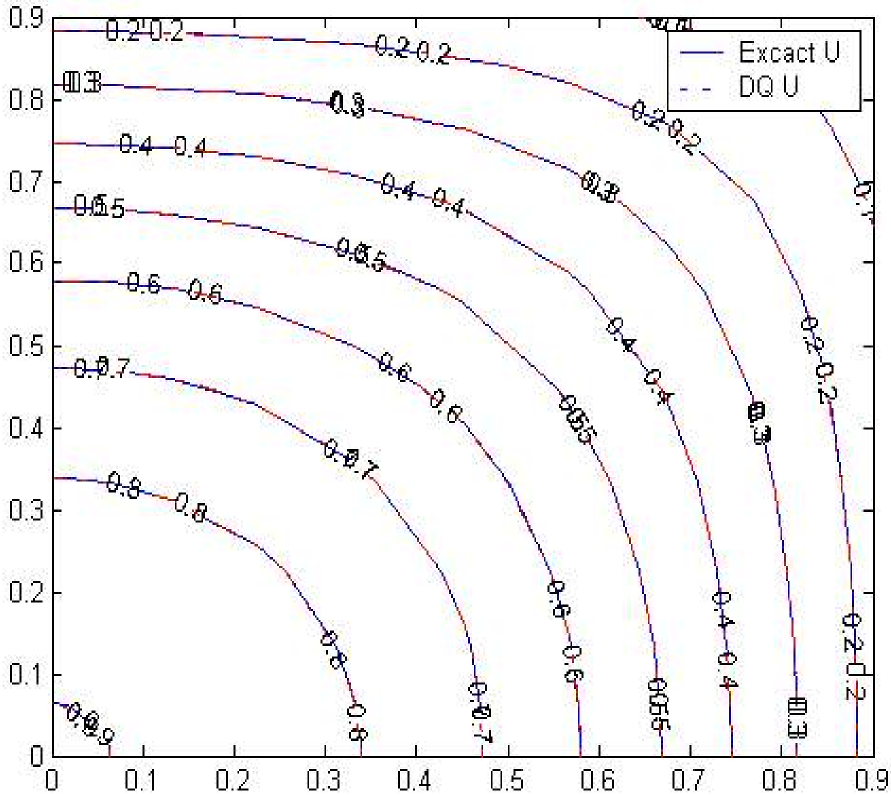


Figure 3.26: At t=0.1 with $N = 13$ grid points, $\Delta t = 0.001$

Problem 8

In this problem coefficients of the second order derivatives are functions of x, y and t.

$$\frac{\partial u(x,y,t)}{\partial t} = \frac{x+1}{2(1+y)(1+t)^2}\frac{\partial^2 u}{\partial x^2} + \frac{1+y}{2(1+x)(1+t)^2}\frac{\partial^2 u}{\partial y^2} \qquad 0 \le x, y \le 1, t > 0 \tag{3.37}$$

with the initial condition

$$u(x,y,0) = e^{(1+x)(1+y)} \tag{3.38}$$

and the Dirichlet boundary conditions are given by

$$u(x,0,t) = e^{(1+x)(1+t)} \qquad u(x,1,t) = e^{2(1+x)(1+t)} \tag{3.39}$$

$$u(0,y,t) = e^{(1+y)(1+t)} \qquad u(1,y,t) = e^{2(1+y)(1+t)}. \tag{3.40}$$

The analytical solution is

$$u(x,y,t) = e^{(1+x)(1+y)(1+t)}. \tag{3.41}$$

The weighting coefficients are recomputed for each time level and therefore the matrix [A] is changed from time level to time level. Since the size of the matrix is small this recomputations are not time consuming contrary to FDM and FEM.

Here, $\Delta t = 0.0001$ is used for drawing potential contours at the time level $t = 0.1$ and $t = 0.01$. Figures 3.27 and 3.28 show very well agreement between computed solution and exact solution.

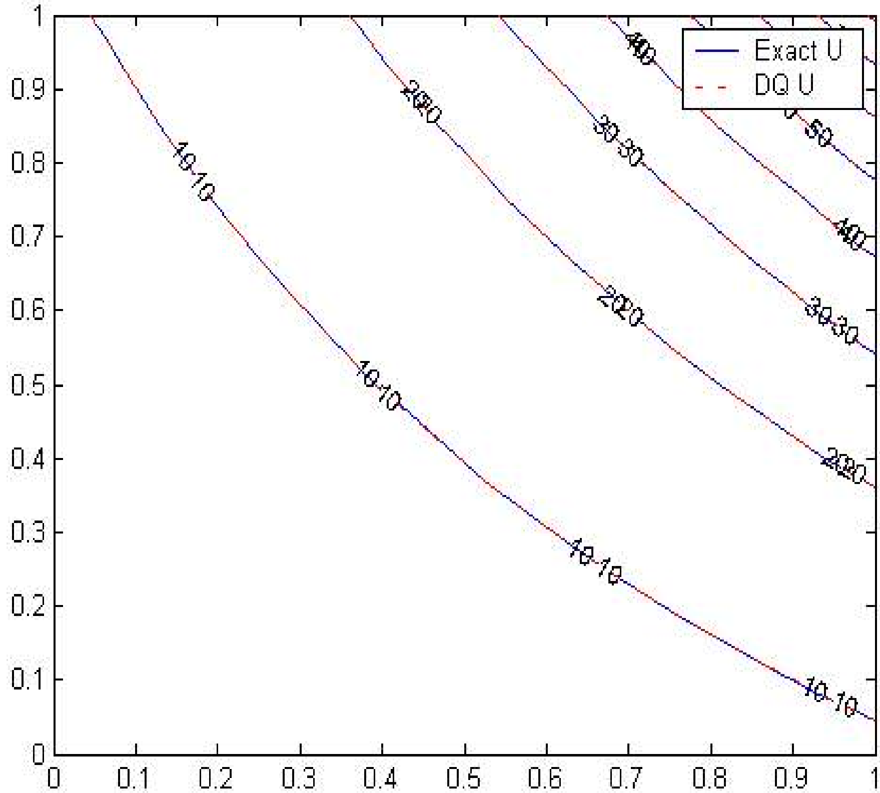


Figure 3.27: At t=0.1 with $N = 13$ grid points, $\Delta t = 0.0001$

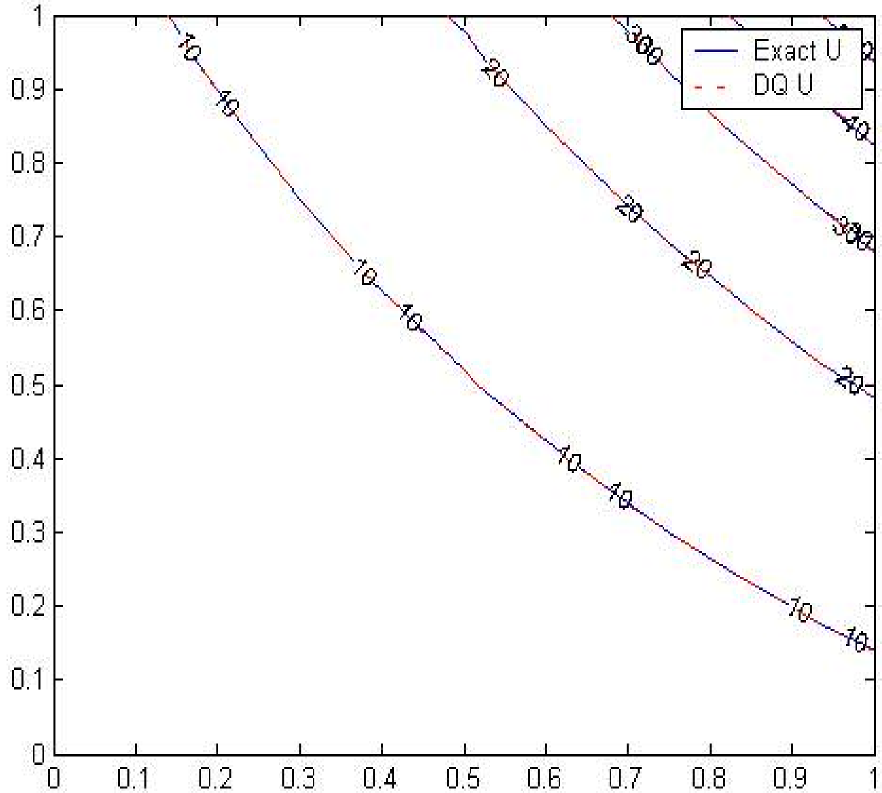


Figure 3.28: At t=0.01 with $N = 13$ grid points, $\Delta t = 0.0001$

CHAPTER 4

CONCLUSION

In this thesis, the Differential Quadrature Method is accompanied with the fourth order Runge-Kutta Method for solving time dependent diffusion equation. Derivatives in space directions are discretized with DQM while time derivative is discretized by using 4th order RKM. The combination of these two methods performs very well since the resulting system of ordinary differential equations is stable for Dirichlet and Dirichlet-Neumann type of boundary conditions and the explicit 4th order RKM has quite a large stability region as well as practical to use.

The numerical procedure presented is applied to several Dirichlet Dirichlet-Neumann type problems. The solutions depend on the choices of step sizes Δt and h in time and space directions respectively since they have to satisfy the stability criteria of RKM. Thus, results are obtained and presented in terms of contours for several values of Δt and h (for several values of number of grid points)

Time dependent coefficients in the equation and time dependent boundary conditions are also implemented in the method and applied to two problems.

When the results are compared with the exact solutions, the high accuracy, efficiency and the stability of the method is observed.

For the DQM discretization, we did not require more than $N = 13$ number of grid points in space directions since with $N = 13$ we obtained very well accuracy. Thus, DQM has the advantage that it gives very accurate numerical results using a considerably smaller number of grid points as compared to the other numerical methods.

REFERENCES

Atkinson, E.K. *An introduction to numerical analysis*, Jhon Wiley, (1978).

Bellman, R. and Casti, J. Differential quadrature and long-term integration, *J Math Anal Appl*, 34, 235-238, (1971).

Bellman, R., Kashef, B.G., and Casti, J. Differential quadrature: A technique for the rapid solution of nonlinear partial differential equations, *J Comput Phys*, 10, 40-52, (1972).

Bellman, R., Roth, R.S. *Method of approximation: tecchnics for mathematical modeling*, D. Reidel Publishing Company, Dordrecht, Holland, (1986).

Bert, W. and Malik, M. Differential quadrature method in computational mechanics: A review, *Appl Mech Rev*, 49, (1996).

Butcher, J.C. *Numerical methods for ordinary differential equations*, Jhon Wiley, (2003).

Chang, C. The development of irregular elements for differential quadrature element method steady-state heat conduction analysis, *Comput Meth in Appl Mech and Eng*, 170, 1-14, (1999).

Chawla, M. M. and Al-Zanaidi, M. A. An Extended Trapezoidal Formula for the Diffusion Equation in two space dimensions, *Comput and Math with Appl*, 42, 157-168, (2001).

Chen, W., Yu, Y. and Wang, X. Reducing the computational of Differential Quadrature Method, *Num Meth for PDE*, 12(5), 565-577, (1996).

Civan, F. Rapid and accurati solution of reactor models by the quadrature method, *Comput in Chemical Eng*, 18, 1005-1009, (1994).

Fung, T. C. Solving initial problems by differential quadrature method-Part 1: first-order equations, *Comput Meth in Appl Mech and Eng*, 50(6), 1411-1427, (2001).

Fung, T. C. Solving initial problems by differential quadrature method-Part 2: second and higher-order equations, *Comput Meth in Appl Mech and Eng*, 50(6), 1429-1454, (2001).

Fung, T. C. Imposition of boundary conditions by modifying the weighting coefficient matrices in the differential quadrature method, *Int J for Num Meth in Eng*, 56(3), 405-432, (2003).

Jain, M.K. *Numerical Solution of Differential Equations* (1984).

Lam S.S.E. Application of the differential quadrature method to two dimensional problems with arbitrary geometry, *Comput and Struct*, 47(3), 459-464, (1993).

Quan J.R. and Chang C. T. New insights in solving distributed system equations by the quadrature methods, *I. Comput Chem Eng*, 13, 779-788, (1989a).

Quan J.R. and Chang C. T. New insights in solving distributed system equations by the quadrature methods, *II. Comput Chem Eng*, 13, 1017-1024, (1989b).

Repaci A. A nonlinear invers heat-transfer problem, *Comput Math Appl*, 21(11-12), 139-143, (1991).

Shu, C. and Chew, Y.T. Fourier expansion-based differential quadrature and its application to Helmholtz eigenvalue problems, *Commun Numer Methods Eng*, 13(8), 643-653, (1997).

Shu, C. and Richards B. E. High resolution of natural convection in a square cavity by generalized differential quadrature, *Proc of 3rd Conf on Adv in Num Meth in Eng: Theory and Appl*, 2, 978-985, (1990).

Shu, C. and Xue, H. Explicit computation of weighting coefficients in the harmonic differential quadrature, *J Sound Vib*, 204(3), 549-555, (1997).

Shu, C. and Xue, H. Solution of Helmhotz equation by differential quadrature method, *Comput Meth Appl Mech Eng*, 175, 203-212, (1999).

Shu, C. *Differential quadrature and its application in engineering*, Springer-Verlag, London, (2000).

Tanaka, M. and Chen, W. Coupling dual reciprocity boundary element method and differential quadrature method for time dependent diffusion problems, *Appl Math Modelling*, 25, 257-268, (2001).

Wu, T. Y. and Liu, G. R. A differential quadrature as a numerical method to solve differential equations, *Comput Mech*, 24(3), 197-205, (1999).