Mathematical Theory and Modeling
ISSN 2224-5804 (Paper) ISSN 2225-0522(Online)
Vol 3, No.12, 2013
www.iiste.org

An improved ADI-DQM based on Bernstein polynomial for solving two-dimensional convection-diffusion equations

A.S.J. Al- Saif [1] and Firas Amer Al-Saadawi [2]

[1]Department Mathematics, Education College for pure science, Basrah University, Basrah, Iraq

[2]Directorate of Education in Basrah, Basrah, Iraq

E-mail:sattaralsaif@yahoo.com, firasamer519@yahoo.com

Abstract

In this article, we presented an improved formulations based on Bernstein polynomial in calculate the weighting coefficients of DQM and alternating direction implicit-differential quadrature method (ADI-DQM) that is presented by (Al-Saif and Al-Kanani (2012-2013)), for solving convection-diffusion equations with appropriate initial and boundary conditions. Using the exact same proof for stability analysis as in (Al-Saif and Al-Kanani 2012-2013), the new scheme has reasonable stability. The improved ADI-DQM is then tested by numerical examples. Results show that the convergence of the new scheme is faster and the solutions are much more accurate than those obtained in literature.

Keywords: Differential quadrature method, Convection-diffusion, Bernstein polynomial, ADI, Accuracy.

1- Introduction

The convection-diffusion equations are widely used in various fields such as petroleum reservoir simulation, subsurface contaminant remediation, heat conduction, shock waves, acoustic waves, gas dynamics, elasticity [2,13,16]. The studies conducted for solving the convection-diffusion equations in the last half century are still in an active area of research to develop some better numerical methods to approximate its solution. Extensive researches have been carried out to handle different types of convection-diffusion equation by different numerical methods [2,4,9,11,12,19,28]. The best oldest known approximation techniques are the finite difference (FD) and finite element (FE) methods. A relatively new numerical technique is the differential quadrature method (DQM). Despite being a domain discretization method, the differential quadrature method gives accurate results using less discretization points than all the above mentioned methods (FD&FE). DQM depends on the idea of integral quadrature and approximate a spatial partial derivatives as a linear weighted sum of all functional values of the solution at all mesh points [16]. This method was proposed by Bellman and Casti [5] in 1971. One of important keys to DQM lies in the determination of weighting coefficients for the discretization of a spatial derivative of any order, where it play the important role in the accuracy of numerical solutions. Initially, Bellman et al.[6](1972), suggested two methods to determine the weighting coefficients of the first order derivative. The first method solves an algebraic equation system. The second use a simple algebraic formulation, but with the coordinates of grid points chosen as the roots of the Legendre polynomial. Quan and Chang [20] (1989a) and Shu and Richards [25] (1992), derived a recursive formula to obtain these coefficients directly and irrespective of the number and positions of the sampling points. In their approach, they used the Lagrange polynomials as the trial functions and found a simple recurrence formula for the weighting coefficients, and used by Meral [16] (2013) and Jiwari [13] (2013). Bert et al.[7](1993) and Striz et al.[27] (1995) developed the differential quadrature method, which uses harmonic functions instead of polynomial as test function in the quadrature method to handle periodic problems efficiently, and also circumvented the limitation for the number of grid point in the conventional DQM based on polynomial test function. Their study shows that the proper test functions are essential for the computational efficiency and reliability of the DQM. Shu et al.[23] (2001) presented a numerical study of natural convection in an eccentric annulus between a square outer cylinder and a circular inner cylinder using DQM, by using Fourier series expansion as the trial functions to compute weighting coefficients. Krowiak[15] (2008) studied the methods that based on the differential quadrature in vibration analysis of plates, and using the spline functions as the trial functions to compute weighting coefficients. Korkmaz et al.[14] (2011) used the quartic B-spline differential quadrature method, and applied it on the one-dimensional Burger’s equation, by using the quartic B-spline functions as the trial functions to compute weighting coefficients.

The motivations to introduce the current work are: Firstly, Attempt of all the above researchers to develop the DQM by using different test functions in computing weighted coefficients of DQM motivating us to research about polynomial has good properties and suitable with application of DQM. Therefore, we noticed that Bernstein polynomials are incredibly useful mathematical tools as they are simply defined. They can be calculated quickly on computer systems and represent a tremendous variety of functions. They can be differentiated and integrated easily, and can be pieced together to form spline curves that can approximate any function to any accuracy desired. One of important properties to Bernstein polynomials is surely convergence. Secondly, Most recently, Al-Saif and Al-Kanani [3,4] (2012-2013) proposed a new improvement for DQM that is resulting from applied ADI into DQM for convection-diffusion problems, and the results of ADI-DQM with Lagrange polynomial and Fourier series expansion as the test functions to computing the weighted coefficients show the efficiency of the proposed method to handle the problems under consideration. Moreover, we wanted to be extending the application of our new suggestion in [2] to improve ADI-DQM.

Depending on these reasons and according to our humble knowledge that the Bernstein polynomials are not yet used to calculate weighting coefficients, in this article, we suggest Bernstein polynomials as test functions to compute the weighting coefficients of the spatial derivatives, in order to introduce a new development to the differential quadrature method that is called alternating direction implicit formulation of the Bernstein differential quadrature method (ADI-BDQM). Using the ADI-BDQM for solving convection-diffusion problems excellent numerical results are obtained. ADI-DQM is then tested by numerical examples. Results show that the convergence of the new scheme is faster and the solutions are much more accurate than those in literature [3,4,11,12] and has reasonable stability.

2- Bernstein Differential quadrature method (BDQM)

The differential quadrature is a numerical technique used to solve the initial and boundary value problems. This method was proposed by Bellman and Casti[5] in (1971). The DQM is based on the idea that the partial derivative of a field variable at the i^{th}discrete points in the computational domain is approximated by a weighted linear sum of the values of the field variable along the line that passes though that point, which is parallel with coordinate direction of the derivative as following[2]:

$$\frac{\partial^m u}{\partial x^m}\Big|_{x=x_i} = \sum_{k=1}^{N} w_{ik}^{(m)}\, u(x_k), \ \ i = 1,2,\dots,N \ \ ,m = 1,2,\dots,N-1 \tag{1}$$

where x_i are the discrete points in the variable, $u_x^{(m)}$ is the m^{th} order derivative of the function, $u(x_i)$ are the function values at these points, and $w_{ik}^{(m)}$ are the weighting coefficients for the m^{th} order derivative of the function with respect to x and N is the number of the grid points. There are two key points in the successful application of the DQM: how the weighting coefficients are determined and how the grid points are selected [17]. Many researchers have obtained weighting coefficients implicitly or explicitly using various test functions [2,3,13,14,21,23].

Here, we use the exact same manner in (Quan and Chang [20] (1989a) and Shu and Richards[25] (1992)) to determine the weighting coefficients, but with employing the Bernstein polynomials as the test function. A Bernstein polynomial, named after Sergei Natanovich Bernstein, is a polynomial in the Bernstein form, that is a linear combination of Bernstein basis polynomials. The Bernstein basis polynomials of n^{th}-degree are defined on the interval $[0,1]$ by Singh et al.[26]:

$$B_{k,n}(x) = \binom{n}{k} x^k (1-x)^{n-k}, \quad 0 \le k \le n \tag{2}$$

The general form of Bernstein polynomials ofn^{th}-degree that used to solve differential equation [8,18] are defined on the interval $[0, L]$ as:

$$B_{k,n}(x) = \binom{n}{k} \frac{x^k (L-x)^{n-k}}{L^n}, \quad 0 \le k \le n \tag{3}$$

were binomial coefficients are given by :

$$\binom{n}{k} = \frac{n!}{k!\ (n-k)!} \tag{4}$$

There are $n+1$, n^{th}-degree Bernstein polynomials. For mathematical convenience, we usually set, $B_{k,n}(x) = 0$, if $k < 0$ or $k > n$. These polynomials are quite easy to write down the coefficients that can be obtained from Pascal's triangle. It can easily be shown that each of the Bernstein polynomials is positive and also the sum of all the Bernstein polynomials is unity for all real $x \in [0,1]$, i.e

$$\sum_{k=0}^{n} B_{k,n}(x) = 1 \quad \forall\, x \in [0,1]$$

The Bernstein polynomials can be written to any interval as following[26]:

$$b(x) \cong B_n(f)(x) = \sum_{k=0}^{n} f\left(\frac{k}{n}\right) B_{k,n}(x) \tag{5}$$

where $f\left(\frac{k}{n}\right)$ is arbitrary function, for $k = 0,1,\dots,n$, $n \geq 1$. Similar to Lagrange differential quadrature method LDQM to determined weighting coefficients, we can derive the explicit formulationto compute the weighting coefficients $w_{ik}^{(1)}$ by using Bernstein polynomial as a test functions, which are listed below:

$$w_{ik}^{(1)} = \begin{cases} \frac{1}{L} \frac{b^{(1)}(x_i)}{(x_i - x_k)\, b^{(1)}(x_k)} & for\ i \neq k, \\ -\sum_{k=1,i\neq k}^{N} w_{ik}^{(1)}\ , & for\ i = k \end{cases} \quad ; i,k = 1,2,\dots,N \tag{6}$$

where L is length interval $[0, L]$ and

$$b(x) = \sum_{k=0}^{n} f\left(\frac{k}{n}\right) \binom{n}{k} \frac{x^k (L - x)^{n-k}}{L^n} \ and\ b^{(1)}(x_i) = \sum_{k=0}^{n} f\left(\frac{k}{n}\right) \binom{n}{k} \frac{(x_i)^k (L - x_i)^{n-k}}{L^n}$$

The weighted coefficients of the second order derivative by using Bernstein polynomial as a test functions can be obtained as

$$[w_{ik}^{(2)}] = [w_{ik}^{(1)}][w_{ik}^{(1)}] = [w_{ik}^{(1)}]^2 \tag{7}$$

The same technique can be used to obtained the weighting coefficients $w_{ik}^{(m)}$.

3- Alternating direction implicit technique - BDQM

To illustrate the application of the technique of ADI to the formula of DQM, we consider the following partial differential equation in two dimensions as;

$$\frac{\partial u}{\partial t} = \ell_x u + \ell_y u \tag{8}$$

where $\ell_x\ and\ \ell_y$ are the differential operators with respect to $x\ and\ y$ respectively.

The alternating direction implicit technique was introduced in the mid-50s by Peaceman and Rachford for solving equations, which result from finite difference discretization of partial differential equations (PDEs). From iterative method's perspective, ADI method can be considered as special relaxation method, where a big system is simplified into a number of smaller systems such that each of them can be solved efficiently and the solution of the whole system is got from the solutions of the sub-systems in an iterative way. Using alternating direction implicit method to approximate equation (8), we get the systems of algebraic equations in the form [3,4]:

$$\frac{u_{ij}^{n+\frac{1}{2}} - u_{ij}^{n}}{\frac{\Delta t}{2}} = \sum_{k=1}^{N} ((w_{dqx})_{ik}) u_{ik}^{n+\frac{1}{2}} + \sum_{l=1}^{M} ((w_{dqy})_{jl})\, u_{jl}^{n} \tag{9}$$

$$\frac{u_{ij}^{n+1} - u_{ij}^{n+\frac{1}{2}}}{\frac{\Delta t}{2}} = \sum_{k=1}^{N} ((w_{dqx})_{ik}) u_{ik}^{n+\frac{1}{2}} + \sum_{l=1}^{M} ((w_{dqy})_{jl})\, u_{jl}^{n+1} \tag{10}$$

where $(w_{dqx})_{ik}$ and $(w_{dqy})_{jl}$ are the BDQM quantities that including the weighting coefficients for the differential operators ℓ_x and ℓ_y, respectively.

Equation (9) is used to compute function values at all interval mesh points along rows and known as horizontal traverse or x −sweep. While, Equation (10) is used to compute function values at all interval mesh points along columns and known as vertical traverse or y −sweep.

4- Numerical examples and discussion

In this section, we apply ADI-BDQM on two test problems to demonstrate the efficiency of the ADI-BDQM. These examples are chosen such that their exact solutions are known.

Problem 1 (Dehghan and Mohebbi[12])

Consider the unsteady state linear two-dimensional convection-diffusion equation:

$$\frac{\partial u}{\partial t}+\beta_x\frac{\partial u}{\partial x}+\beta_y\frac{\partial u}{\partial y}-\alpha_x\frac{\partial^2 u}{\partial x^2}-\alpha_y\frac{\partial^2 u}{\partial y^2}=0\,,(x,y,t)\in[0,L]\times[0,L]\times[0,T] \tag{11}$$

where $u(x,y,t)$ is a transported variable, β_x and β_y are arbitrary constants which show the speed of convection and the diffusion coefficients α_x and α_y are positive constants. The initial condition of Equation (11) has the following form:

$$u(x,y,0)=a(e^{-x\,c_x}+e^{-y\,c_y}) \quad , \quad 0\le x,y\le 1 \tag{12}$$

where $c_x=\frac{-\beta_x\pm\sqrt{\beta_x^2+4b\alpha_x}}{2\alpha_x}>0 \quad , \quad c_y=\frac{-\beta_y\pm\sqrt{\beta_y^2+4b\alpha_y}}{2\alpha_y}>0$

and the boundary conditions are given by:

$$\left.\begin{array}{l} u(0,y,t)=ae^{bt}(1+e^{-y\,c_y})\,,u(1,y,t)=ae^{bt}(e^{-c_x}+e^{-y\,c_y}) \\ u(x,0,t)=ae^{bt}(e^{-x\,c_x}+1)\,,u(x,1,t)=ae^{bt}(e^{-x\,c_x}+e^{-c_y}) \end{array}\right., t>0 \tag{13}$$

The exact solution is given as:

$$u(x,y,t)=ae^{bt}(e^{-x\,c_x}+e^{-y\,c_y}) \quad , \quad 0\le x,y\le 1\,, \quad t>0 \tag{14}$$

Equation (11) can be approximated by using ADI-BDQM, equations (9 and 10), such that $(w_{dqx})_{ik}=\alpha_x w_{ik}^{(2)}-\beta_x w_{ik}^{(1)}$ and $(w_{dqy})_{jl}=\alpha_y\widetilde{w}_{jl}^{(2)}-\beta_y\widetilde{w}_{jl}^{(1)}$, where the $w_{ik}^{(1)},w_{ik}^{(2)},\widetilde{w}_{jl}^{(1)}$ and $\widetilde{w}_{jl}^{(2)}$ are the weighted coefficients of the first and the second order derivatives with respect to x and y, respectively. In this problem, we take $\beta_x=\beta_y=-1$, $L=1$, $a=1$ and $b=0.1$, and use equally spaced grid points. Tables 1 and 2 are shows the errors obtained from solving problem 1 by using LDQM, ADI-LDQM and ADI-BDQM at $t=0.1$, $\Delta t=0.0001$ and for different values of $h=\Delta x=\Delta y$. Fig. 1 clarifies a comparison between exact solution and numerical solutions for $\alpha_x=\alpha_y=0.01$ and 0.1 respectively. The results confirm that the ADI-BDQM is more accurate and less CPU time than the LDQM and ADI-LDQM.

Table 1 Errors obtained for problem 1 with $t=0.1,\ \alpha_x=\alpha_y=0.01$

h	Max |error| of LDQM	CPU	Max |error| of ADI-LDQM	CPU	Max |error| of ADI-BDQM	CPU
0.25	2.912711E-08	0.514	4.879633E-09	0.548	7.335061E-10	0.522
0.17	2.108340E-08	0.615	2.073938E-09	0.668	6.280813E-10	0.644
0.125	1.371521E-08	0.770	1.262915E-09	0.960	5.741963E-10	0.907
0.1	7.726773E-09	1.105	9.595748E-10	1.522	4.953427E-10	1.443

Table 2 Errors obtained for problem 1 with $t=0.1,\ \alpha_x=\alpha_y=0.1$

h	Max |error| of LDQM	CPU	Max |error| of ADI-LDQM	CPU	Max |error| of ADI-BDQM	CPU
0.25	2.572117E-06	0.509	2.844328E-06	0.540	1.832912E-07	0.528
0.17	1.379575E-05	0.619	1.437375E-05	0.624	2.033508E-06	0.595
0.125	3.223357E-05	0.772	3.292829E-05	0.874	3.652773E-06	0.859
0.1	5.348801E-05	1.047	5.434319E-05	1.415	4.038135E-06	1.282

We choose the arbitrary function $f\left(\frac{k}{n}\right) = exp\left(s * \frac{k}{n} * h\right), k = 0,1, \dots, n$ and s is the arbitrary constant. In this problem, we take $s = 88, 179, 339$ and 526 for Table 2. and $s = -50, -35, -7$ and 116 for Table 3. respectively at the number of grid points $N = M = 5, 7, 9$ and 11. Notice that in the next example, we choose the same above arbitrary function with different values of s.

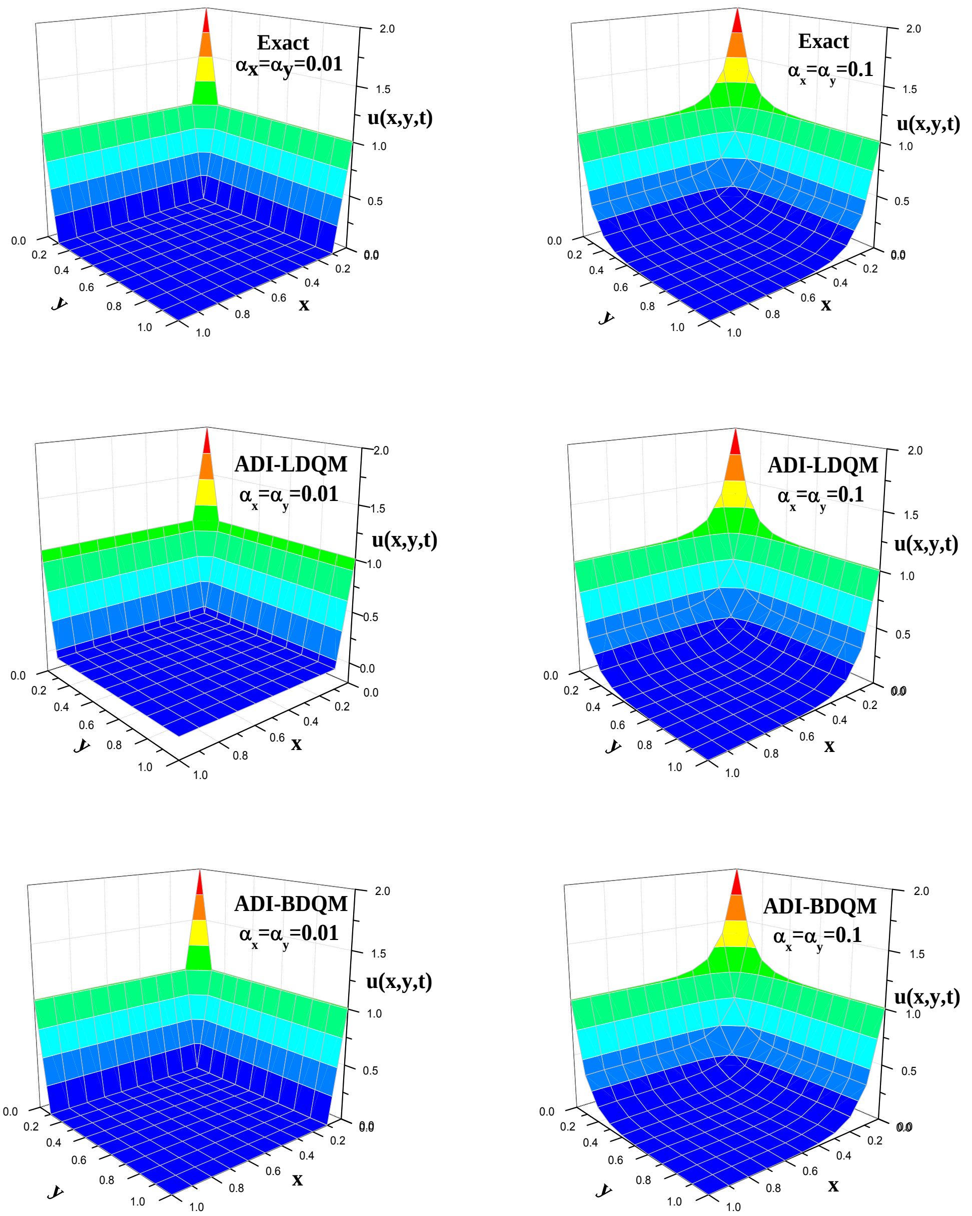

Fig. 1 Exact and approximate solutions of the problem 1 with, $t = 0.1$, $\alpha_x = \alpha_y = 0.01$ and 0.1.

Problem 2 (Al-Saif and Al-kanani[4])

Consider the nonlinear two-dimensional convection-diffusion equations that are called Burger's Equations:

$$\frac{\partial u}{\partial t} + \alpha u \frac{\partial u}{\partial x} + \alpha v \frac{\partial u}{\partial y} - \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = 0 \qquad (15)$$

$$\frac{\partial v}{\partial t} + \alpha u \frac{\partial v}{\partial x} + \alpha v \frac{\partial v}{\partial y} - \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) = 0 \qquad (16)$$

where $u(x,y,t)$ and $v(x,y,t)$ are the velocity components to be determined, α is constant and Re is the Reynolds number, $(x,y,t) \in [0,L] \times [0,L] \times [0,T]$.

In this problem, we take $\alpha = -2$, $Re = 1$, $L = 1$, and the initial conditions of Equations (15) and (16) have the following form:

$$u(x,y,0) = \frac{1}{2} - \frac{x+y}{1+x+y}, \qquad v(x,y,0) = \frac{1}{2} + \frac{x+y}{1+x+y} \tag{17}$$

The exact solutions are given by:

$$u(x,y,t) = \frac{1}{2} - \frac{x+y+t}{1+x+y+t}, \qquad v(x,y,t) = \frac{1}{2} + \frac{x+y+t}{1+x+y+t} \tag{18}$$

The boundary conditions can be achieved easily from (18) by using $x, y = 0,1$.

Equations (15-16) can be approximated by using ADI-BDQM ((Equations(9 and 10)), such that $\left(w_{dqx}\right)_{ik} = \frac{1}{Re} w_{ik}^{(2)} - \alpha u_{ij}^{n} w_{ik}^{(1)}$ and $\left(w_{dqy}\right)_{jl} = \frac{1}{Re} \widetilde{w}_{jl}^{(2)} - \alpha v_{ij}^{n} \widetilde{w}_{jl}^{(1)}$, where the $w_{ik}^{(1)}, w_{ik}^{(2)}, \widetilde{w}_{jl}^{(1)}$ and $\widetilde{w}_{jl}^{(2)}$ are the weighted coefficients of the first and the second order derivatives with respect to x and y, respectively. For the above problem, we found numerical solutions for u and v and use equally spaced grid points. Tables 3 and 4 are shows the errors obtained from solving problem 2 by LDQM, ADI-LDQM and ADI-BDQM at $t = 0.01$ and 0.1, $\Delta t = 0.0001$ and x and $y \in [0,1]$ for different values of $h = \Delta x = \Delta y$. Fig. 2 clarifies a comparison between exact solution and numerical solutions of the problem 2. The results show that the ADI-BDQM has a high accuracy, good convergence and less CPU time compared with the LDQM and ADI-LDQM.

Table 3 Errors obtained ADI-LDQM and ADI-BDQM for problem 2 with $t = 0.01, Re = 1$ of u

h	Max |error|of LDQM	CPU	Max |error|of ADI-LDQM	CPU	Max |error| of ADI-BDQM	CPU
0.25	1.943390E-05	0.098	1.441773E-05	0.102	9.541987E-06	0.097
0.17	3.109294E-05	0.106	1.970786E-05	0.115	1.386325E-05	0.100
0.125	4.022089E-05	0.122	2.122300E-05	0.138	1.610371E-05	0.119
0.1	4.733133E-05	0.151	2.021087E-05	0.163	6.885246E-06	0.126

Table 4 Errors obtained ADI-LDQM and ADI-BDQM for problem 2 with $t = 0.01, Re = 1$ of v

h	Max |error|of LDQM	CPU	Max |error| of ADI-LDQM	CPU	Max |error| of ADI-BDQM	CPU
0.25	1.943416E-05	0.100	9.237709E-06	0.105	3.091749E-08	0.100
0.17	3.109260E-05	0.110	9.286324E-06	0.124	7.134462E-07	0.101
0.125	4.022089E-05	0.122	6.145820E-06	0.135	1.674942E-06	0.122
0.1	4.733010E-05	0.137	2.454387E-06	0.160	5.610121E-06	0.136

In this problem, we take $s = 7, 42, 275$ and 375 respectively at the number of grid $N = M = 5$, 7 , 9 and 11.

5- Comparison with the other methods

We compare the numerical results of ADI-BDQM for problems 1 and 2 with the results of other numerical methods such as ADI-LDQM, High-order compact boundary value method (HOCBVM)[12]and Radial basis function based meshless method(RBFBMM)[11]. The error measurements resulted from the ADI-BDQM is more accurate than the methods, HOCBVM[12], RBFBMM[11] and ADI-LDQM. Moreover, the number of grid points by using ADI-BDQM and ADI-LDQM are less than the other methods.

6- Stability analysis of BDQM

The stability of numerical schemes is closely related to numerical error. A solution is said to be unstable if errors appear at some stage in the calculations (for example, from erroneous initial conditions or local truncation or round-off errors) are propagated without bound throughout subsequent calculations. Thus a method is stable if small changes in the initial data produce correspondingly small changes in the final results, that is, the difference between the theoretical and numerical solutions remains bounded at a given time t, as time and space steps tend to zero or time step remains fixed at every level and $t \to \infty$[1]. So stability, means that the numerical solution must be close to the exact solution, meaning that whenever was the error a little the deviation in derivatives, however, this error may accumulate at each time step and affects to the stability of the solution.

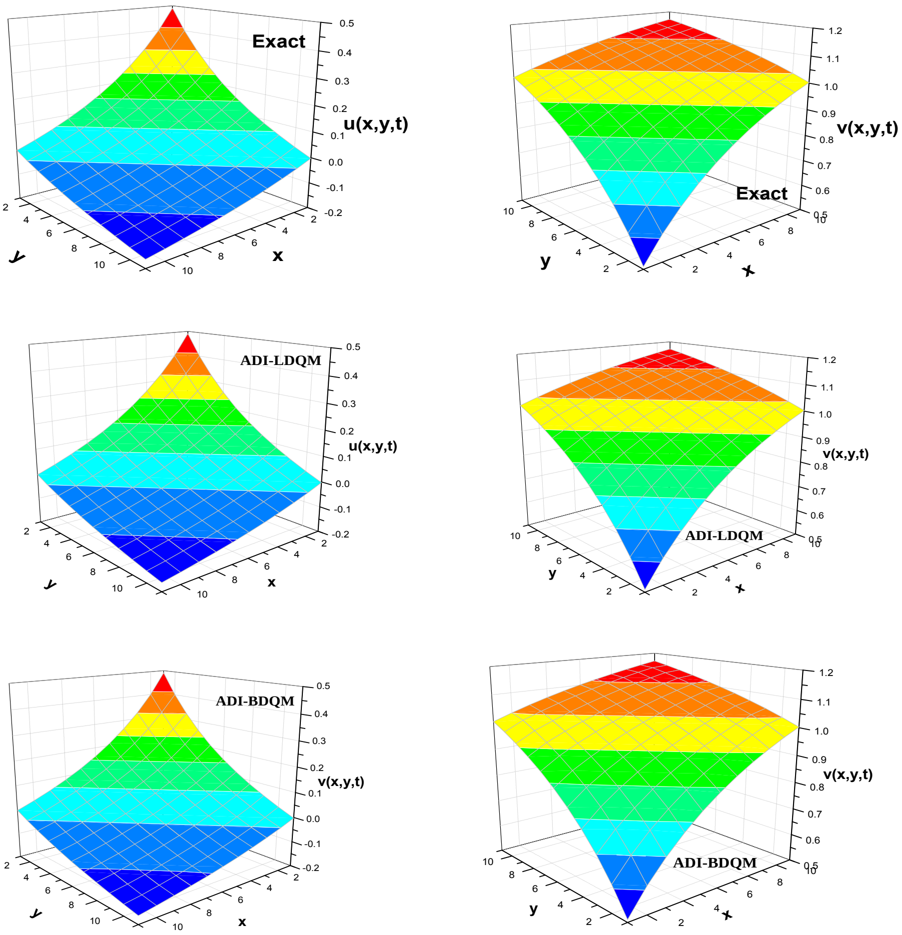

Fig. 2 Exact and approximate solutions of the problem 2 with, t=0.01 and Δt=0.0001

Table 5. Comparison of the numerical results of the problem 1 for different methods at $\Delta t = 0.0001, t = 0.1$.

Method	Number grid points	Max$\|error\|$of u, $\alpha_x = \alpha_y = 0.01$	Max$\|error\|$of u, $\alpha_x = \alpha_y = 0.1$
ADI-BDQM	11×11	4.953427E-10	4.083296E-06
ADI-LDQM	11×11	9.595748E-10	5.434319E-05
HOCBVM[12]	25×25	1.1826E-01	2.4170E-05
RBFBMM[11]	41×41	4.97E-02	4.25E-02

Table 6. Comparison of the numerical results of the problem 2 for different methods at $\Delta t = 0.0001, t = 0.01, Re = 1$.

Method	Number grid points	Max$\|error\|$ of	Max$\|error\|$ of
ADI-BDQM	5×5	9.486005E-06	1.230419E-08
ADI-LDQM	5×5	1.532977E-05	9.154925E-06
BDQM	5×5	1.877131E-05	1.877144E-05
LDQM	5×5	1.943390E-05	1.943416E-05

Theorem[22]

The system of ODE $\hat{u} = Au$ with a constant coefficient matrix A is,

(1) Stable if the roots of the characteristic polynomial are purely imaginary.

(2) Asymptotically stable if the roots have negative real parts.

(3) Unstable if a root has positive real part.

From application of BDQM to the any convection-diffusion equation in this work, we obtained the set ordinary differential equations:

$$[A]\,\{u\} = \{b\} - \{r\} \tag{19}$$

where $\{u\}$ is a vector of unknown functional values at all the interior points given by $\{u\} = \left[u_{2,2}, u_{2,3}, \dots, u_{2,M-1}, u_{3,2}, \dots, u_{3,M-1}, \dots, u_{N-1,2}, \dots, u_{N-1,M-1}\right]^T$

and $\{r\}$ is a known vector which is made up of the non-homogeneous part and the boundary conditions given by

$$\{r\} = \left[r_{2,2}, r_{2,3}, \dots, r_{2,M-1}, r_{3,2}, \dots, r_{3,M-1}, \dots, r_{N-1,2}, \dots, r_{N-1,M-1}\right]^T$$

and $[A]$ is the coefficient matrix containing the weighting coefficients, the dimension of the matrix $[A]$ is $(N-2)(M-2)$ by $(N-2)(M-2)$. For the multi-dimensional case, the matrix $[A]$ contains many zero elements, which are irregularly distributed in the matrix.

The stability analysis of the Equation (19) is based on the eigenvalue distribution of the BDQM discretization matrix $[A]$. If $[A]$has eigenvalues λ_i and corresponding eigenvector ξ_i,$(i = 1,2,\dots,K)$ K being the size of the matrix$[A]$, the similarity transformation reduces the system (19) of the form[10,24].

$$\frac{d\{U\}}{dt} = [D]\{U\} + \{R\} \tag{20}$$

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

$$[D] = [P]^{-1}[A][P] \tag{21}$$

Pre-multiplying by the matrix $[P]^{-1}$ on the both sides Equation (20) and setting

$$\{U\} = [P]^{-1}[u] \tag{22}$$

$$\{R\} = [P]^{-1}[r] \tag{23}$$

Since $[D]$ is a diagonal matrix, Equation (20) is an uncoupled set of ordinary differential equations.

Considering the i^{th} equation of (20)

$$\frac{dU_i}{dt} = \lambda_i U_i + R_i \tag{24}$$

If R_i is time-independent, then the solution of Equation (24) can be written as

$$U_i = \left(U_i(0) + \frac{R_i}{\lambda_i}\right) e^{\lambda_i t} - \frac{R_i}{\lambda_i} \tag{25}$$

For this case, using Equations (22) and (23), the solution $\{u\}$ can be obtained as

$$\{u\} = [P][U] = \sum_{i=1}^{N} U_i\,\xi_i = \sum_{i=1}^{N} \left[U_i(0)e^{\lambda_i t} + \frac{R_i}{\lambda_i}(e^{\lambda_i t} - 1)\right] \xi_i \tag{26}$$

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

$$Real(\lambda_i) \le 0 \quad \text{for all} \quad i \tag{27}$$

where $Real(\lambda_i)$ denotes the real part of λ_i. This is the stability condition

for the system (19).

In this section, we can applied the stability condition (27) on the problems that mentioned in the previous section by using ADI-BDQM.

Problem 1.

From the application of ADI-BDQM to the Equation (11) and using $\beta_x = \beta_y = -1$, and $N = M = 5$, Equation (11) can be rewritten as:

$$\sum_{k=2}^{4} (\alpha_x w_{ik}^{(2)} - \beta_x w_{ik}^{(1)}) u_{kj}^n + \sum_{l=2}^{4} (\alpha_y \widetilde{w}_{jl}^{(2)} - \beta_y \widetilde{w}_{jl}^{(1)}) u_{il}^n = \{b\} - \{r\} \quad (28)$$

where $2 \le i \le 4$, $2 \le j \le 4$

From Equation (28), we can obtain a system of algebraic equations (19).

This system has the solution (26), and this solution $\{u\}$ is stable as $t \to \infty$ and the real parts of the eigenvalues of the matrix $[A]$ for $\alpha_x = \alpha_y = 0.01, 0.1$ respectively, are:

$Real(\lambda_1) = -0.231296, Real(\lambda_2) = -0.231296, Real(\lambda_3) = 0.025635,$

$Real(\lambda_4) = -0.102832, Real(\lambda_5) = -0.102832, Real(\lambda_6) = -0.102832,$

$Real(\lambda_7) = -0.102832, Real(\lambda_8) = -0.231296, Real(\lambda_9) = -0.231296.$

and

$Real(\lambda_1) = -15.583721, Real(\lambda_2) = -15.583721, Real(\lambda_3) = -4.524857,$

$Real(\lambda_4) = -15.027137, Real(\lambda_5) = -15.027137, Real(\lambda_6) = -8.901465,$

$Real(\lambda_7) = -8.901465, Real(\lambda_8) = -10.327329, Real(\lambda_9) = -10.327329.$

This means that the stability condition (27) is hold.

Problem 2.

From the application of ADI-BDQM to the Equation (15) and using $Re = 1$, $\alpha = -2$ and $N = M = 5$ Equation (15) can be rewritten as:

$$\sum_{k=2}^{4} \left(\frac{1}{Re} w_{ik}^{(2)} - \alpha u_{ij}^n w_{ik}^{(1)}\right) u_{kj}^n + \sum_{l=2}^{4} \left(\frac{1}{Re} \widetilde{w}_{jl}^{(2)} - \alpha v_{ij}^n \widetilde{w}_{jl}^{(1)}\right) u_{il}^n = \{b\} - \{r\} \quad (29)$$

where $2 \le i \le 4$, $2 \le j \le 4$

From Equation (29), we can obtain a system of algebraic equations (19).

This system has the solution (26), and this solution $\{u\}$ is stable as $t \to \infty$ and the real parts of the eigenvalues of the matrix $[A]$are:

$Real(\lambda_1) = -22.813375, Real(\lambda_2) = -99.583607, Real(\lambda_3) = -99.583607,$

$Real(\lambda_4) = -78.223693, Real(\lambda_5) = -78.223693, Real(\lambda_6) = -46.002054,$

$Real(\lambda_7) = -54.995694, Real(\lambda_8) = -54.995694, Real(\lambda_9) = -67.664862.$

This means that the stability condition (27) is hold. When using the equation (22), we will find the same eigenvalues mentioned above of the matrix [A].

Finally, the numerical results of the above problems confirm that the newly developed method ADI-BDQM is stable for the grid points $N = M = 5$. In this work, with the help of symbolic computation software Maple 13, the eigenvalues are computed.

7- Conclusions

In this work, we employed a new technique ADI-BDQM to solve convection-diffusion equations successfully. The weighting coefficients for spatial derivatives are computing by use Bernstein polynomials as test functions. The numerical results show that the new method has higher accuracy, good convergence and reasonable stability as well as a less computation workload by using few grid points. Moreover, the results show that the application of our new suggestion in [2] to improve ADI-DQM is successful and can be applied on more general problems.

References

[1] A. Ali, Mesh free collocation method for numerical solution of initial-boundary- value problems using radial basis functions, *Ph.D. thesis*, Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Pakistan, (2009).

[2] A. S. J. Al-Saif and Firas A. Al-Saadawi, Bernstein differential quadrature method for solving the unsteady state convection-diffusion equation, *Indian J. of Appl. Reasearch*, 3(9)(2013),20-26.

[3] A. S. J. Al-Saif and M. Al-kanani, Solution of nonlinear initial-value problems by the alternating direction implicit formulation on differential quadrature method, *Int. J. of Pure and Appl. Research in Engin. and Technology*, 1(9)(2013), 57-69.

[4] A. S. J. Al-Saif and M. J. Al-kanani, Alternating direction implicit formulation of the differential quadrature method for solving Burger equations, *Inter. J. Modern Math. Sc.* 30(1)(2012), 1-11.

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

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

[7] C. W. Bert, X. Wang and A. G. Striz, Differential quadrature for static and free vibrational analysis of anisotropic plates, *Int. J. Solids Structures*, 30(1993),1737-1744.

[8] M. I. Bhatti and P. Bracken, Solution of differential equations in a Bernstein polynomial basis, *J. Comput. Appl. Math.*, 205(2007), 272-280.

[9] J. Chan and J. A. Evans, A minimum-residual finite element method for the convection-diffusion equation, *Institute for Comput. Engineering and Sciences ICES*, 13(13)(2013), 1-19.

[10] W. Chen, Differential quadrature method and its applications in engineering , *Ph. D. thesis*, Shanghai JiaoTong University, China, (1996).

[11] P. P. Chinchapatnam, Radial basis function based meshless methods for fluid flow problems, *Ph.D. thesis*, Southampton University, Faculty of Engineering Science and Mathematics, UK, (2006).

[12] M. Dehghan and A. Mohebbi, High-order compact boundary value method for the solution of unsteady convection-diffusion problems, *Math. Comput. Simul.*,79(2008), 683-699.

[13] R. Jiwari, R. Mittal and K. K. Sharma, A numerical scheme based on weighted average differential quadrature method for the numerical solution of Burger's equation, *Appl. Math. Comput.*, 219(2013), 6680-6691.

[14] A. Korkmaz,A. M. Aksoy and I. Dag, Quartic B-spline differential quadrature method, *International J. of Nonlinear Sc.*, 11(2011),403-411.

[15] A. Krowiak, methods based on the differential quadrature in vibration analysis of plates, *J. Theor. Appl. Mech.*, 46(1)(2008), 123-139.

[16] G. Meral, Differential quadrature solution of heat-and mass-transfer equations, *Appl. Math. Modelling*, 37(2013),4350-3459.

[17] M. Mozaffari , A. A. Atai and N. Mostafa, Large deformation and mechanics of flexible isotropic membrane ballooning in three dimensions by differential quadrature method, *J. Mech. Sc. Tech.*, 23(2009), 2921-2927.

[18] Y. Ordokhani and S. D. Far, Application of the Bernstein polynomials for solving the nonlinear Fredholm Integro-differential equations, *J. Appl. Math. And Bioinformatics*,1(2011), 13-31.

[19] F. U. Prieto, J.B. Muñaz and L.G. Corvinos, Application of the generalized finite difference method to solve the advection–diffusion equation, *J. of Comp. and App. Math.*, 235(2011), 1849–1855.

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

[21] M. Ramezani, M. Shahrezaee, H. Kharazi and L. H. Kashany, Numerical solutions of differential algebraic equation by differential quadrature method, *J. Basic. Appl. Sc. Res*, 2(11)(2012), 11821-11828.

[22] D. A. Sanchez, *Ordinary Differential Equations and Stability Theory*: An Introduction, W.H. Freeman and Company, USA, (1968).

[23] C. Shu , H. Xue and Y. D. Zhu, Numerical study of natural convection in an eccentric annulus between a square outer cylinder and a circular inner cylinder using DQ method, *Int. J. Heat and Mass Transfer*, 44(2001), 3321-3333.

[24] C. Shu, *Differential Quadrature and Its Application in Engineering*, springer-verlag, London, (2000).

[25] C. Shu and B. Richards, Application of generalized differential quadrature to solve two-dimensional incompressible Navier-Stokes equations, *Int. J. Numer. Meth. In Fluids*,15(1992), 791-798.

[26] A. K. Singh, V. K. Singh and O. P. Singh, The Bernstein operational matrix of integration, *Appl. Math. Sc.*, 3(49)(2009), 2427-2436.

[27] A. G. Striz, X. Wang and C. W. Bert, Harmonic differential method and applications to structural components, *Acta Mechanica*, 111(1995), 85-94.

[28] H. Zhu, H. Shu and M. Ding, Numerical solutions of two-dimensional Burger's equations by discrete Adomian decomposition method, *Comput. Math. Appl.*, 60(2010), 840-848.