

A Novel Approach for Numerical Computation of Fokker-Planck Equation

Pramod Kumar¹, Brajesh Kumar Singh^{2,*}, Shobh Nath Rai³

1,2,3. Department of Applied Mathematics, School for Physical Sciences, Babasaheb Bhimrao Ambedkar University, Lucknow – 226025, INDIA.

Received: March 14, 2016 / Accepted: April 12, 2016 / Published: July 25, 2016.

Abstract: This paper present an implementation of “*modified cubic B-spline differential quadrature method (MCB-DQM)*” proposed by Arora & Singh (Applied Mathematics and Computation Vol. 224(1) (2013) 161-177) for numerical computation of Fokker-Planck equations. The modified cubic B-splines are used as set of basis functions in the differential quadrature to compute the weighting coefficients for the spatial derivatives, which reduces Fokker-Planck equation into system of first-order ordinary differential equations (ODEs), in time. The well known SSP-RK43 scheme is then applied to solve the resulting system of ODEs. The efficiency of proposed method has been confirmed by three examples having their exact solutions. This shows that MCB-DQM results are capable of achieving high accuracy. Advantage of the scheme is that it can be applied very smoothly to solve the linear or nonlinear physical problems, and a very less storage space is required which causes less accumulation of numerical errors.

Key words: Fokker-Planck equation; modified cubic B-spline; MCB-DQM; SSP-RK43; Thomas algorithm

1. Introduction

Fokker-Planck equation (FPE) (1) was introduced by Risken [1] to describe the Brownian motion of particles, has broad application in engineering and natural science such as solid-state physics, theoretical biology, circuit theory, quantum theory, chemical physics etc.

$$\frac{\partial W}{\partial t} = \gamma \frac{\partial v W}{\partial v} + \gamma \frac{KT}{m} \frac{\partial^2 W}{\partial^2 v}, \quad (1)$$

where $W(x, t)$, v are the distribution function and velocity for the particle of mass m at time t , respectively. $\gamma \rightarrow$ Fraction, $K \rightarrow$ Boltzmann's constant and $T \rightarrow$ temperature of fluid. FPE in more general form can be expressed as [1] :

$$\frac{\partial u}{\partial t} = \left[-\frac{\partial}{\partial x} A(x) + \frac{\partial^2}{\partial x^2} B(x) \right] u, \quad (2)$$

with the initial condition

$$u(x, 0) = \psi(x), x \in D, \quad (3)$$

and boundary conditions

$$u(x, t) = \xi(x), \quad \forall (x, t) \in \partial D \times (0, T], \quad (4)$$

Where $D = \{x : a < x < b\}$ and ∂D is its boundary, ψ , ξ are known, and $u(x, t)$ is unknown. In Eq. (2): $A(x) \rightarrow$ drift coefficient and $(B(x) > 0) \rightarrow$ diffusion coefficient. These coefficients may be function of time t as:

$$\frac{\partial u}{\partial t} = \left[-\frac{\partial}{\partial x} A(x, t) + \frac{\partial^2}{\partial x^2} B(x, t) \right] u. \quad (5)$$

Eq. (1) is a special case of the FPE, where $A(x)$ is linear and $B(x)$ is constant. Eq. (2) represent the motion for the distribution function, $u(x, t)$ is referred to as forward Kolmogorov equation. Similar nonlinear equation has been introduced by Risken [1] as:

*Corresponding author: bksingh0584@gmail.com

$$\frac{\partial u}{\partial t} = \left[-A(x, t) \frac{\partial}{\partial x} + B(x, t) \frac{\partial^2}{\partial x^2} \right] u, \quad (6)$$

Eq. (6) is backward Kolmogorov equation, a more general form of FPE. Nonlinear FPE appeared in several areas such as plasma physics, population dynamics, biophysics, engineering, neurosciences, nonlinear hydrodynamics, polymer physics, laser physics, pattern formation, psychology and marketing [2]. The nonlinear FPE has the following form:

$$\frac{\partial u}{\partial t} = \left[-\frac{\partial}{\partial x} A(x, t, u) + \frac{\partial^2}{\partial x^2} B(x, t, u) \right] u. \quad (7)$$

The numerical and analytical approximate solutions of the FPE have been intensively studied since the work of Risken [2]. The fundamental aspects of nonlinear FPE have been studied by Frank [2]. Harrison [3] has been obtained numerical solution of the Fokker Planck equation for the probability density function of a stochastic process by using finite element method. Recently, numerical solutions of fractional FPE have been solved by iterative Laplace Transform method [4].

Lakestani and Dehghan [5] has been applied cubic B-spline method to solve FPE and found that implementation of cubic B-spline method is easy and gives very accurate results. Palleschi et al. [6] has been used a fast and accurate algorithm to obtain the numerical solution of FPE, and he also studied the stability and convergence of the algorithm analytically. Further, a fast and reliable algorithm for the numerical solution of unidimensional Fokker-Planck equations has been proposed by Palleschi & Rosa [7]. Kazem et al. [8] have been applied two meshless methods to obtain the numerical solutions FPE based on radial basis functions using collocation method. The first method is based on the Kansa's technique while another technique is based on Hermite interpolation.

Sadighi et al. [9] have been introduced homotopy-perturbation method (HPM) and variational iteration method (VIM) to obtain the exact solutions of FPE. Tatari et al. [10] used Adomian

decomposition method while Hesama et al. [11] used differential transform method to find the analytic solutions of FPE.

In this paper, we implement MCB-DQM [12, 22, 23] for numerical computation of Fokker-Planck equations, and Kolmogorov equations. The efficacy of the method has been confirmed by taking three examples having their exact solutions, which shows that MCB-DQM results are acceptable and in good agreement with earlier studies available in literature.

The paper is organized into five more sections which follows this introduction. Sect. 2: describes MCB-DQM. Sect. 3: implementation procedure is illustrated. Stability of the method is studied in Sect. 4. Numerical examples are given to establish the applicability and accuracy of the method in Sect. 4. Conclusion of the article is given in Section 5.

2. Description of MCB-DQM

Differential quadrature method (DQM) was introduced by Bellman et al. [13] to approximate the spatial derivatives of a function using the weighted sum of the functional values at certain discrete points. In DQM, the weighting coefficients are computed by using several kinds of test functions: spline function, sinc function, Lagrange interpolation polynomials and Legendre polynomials [14-20] etc. This section redescribes MCB-DQM [13] to complete our problem. It is assumed that the N grid points $a = x_1 < x_2, \dots < x_N = b$ are distributed uniformly with $x_{i+1} - x_i = h$. The first & second order spatial derivatives [14] of $u(x, t)$ at x_i are defined by

$$\frac{\partial u(x_i, t)}{\partial x} = \sum_{j=1}^N w_{ij}^{(1)} u(x_j, t), \quad i \in \Delta_N \quad (8)$$

$$\frac{\partial^2 u(x_i, t)}{\partial x^2} = \sum_{j=1}^N w_{ij}^{(2)} u(x_j, t), \quad i \in \Delta_N \quad (9)$$

Where $\Delta_N = 1, 2, \dots, N$; $w_{ij}^{(r)} (r=1, 2)$ are the weighting coefficients of r th-order derivative.

The cubic B-spline [14] is defined as follows:

$$\varphi_j(x) = \frac{1}{h^3} \begin{cases} (x-x_{j-2})^3, & x \in (x_{j-2}, x_{j-1}) \\ (x-x_{j-2})^3 - 4(x-x_{j-1})^3, & x \in (x_{j-1}, x_j) \\ (x_{j+2}-x)^3 - 4(x_{j+1}-x)^3, & x \in (x_j, x_{j+1}) \\ (x_{j+2}-x)^3, & x \in (x_{j+1}, x_{j+2}) \\ 0, & \text{else.} \end{cases} \quad (10)$$

where $\{\varphi_0, \varphi_1, \dots, \varphi_N, \varphi_{N+1}\}$ is a basis over $[a, b]$.

The values of $\varphi_j, \varphi'_j, \varphi''_j$ at the nodal points are given in Table 1.

Modified cubic B-splines [12] are defined as:

$$\left. \begin{aligned} \phi_1(x) &= \varphi_1(x) + 2\varphi_0(x) \\ \phi_2(x) &= \varphi_2(x) - \varphi_0(x) \\ \phi_j(x) &= \varphi_j(x) \text{ for } j = 3, \dots, N-2 \\ \phi_{N-1}(x) &= \varphi_{N-1}(x) - \varphi_{N+1}(x) \\ \phi_N(x) &= \varphi_N(x) + 2\varphi_{N+1}(x) \end{aligned} \right\}, \quad (11)$$

where $\{\phi_0, \phi_1, \dots, \phi_N\}$ forms a basis over $[a, b]$.

Approximate value of $\frac{\partial u(x_i, t)}{\partial x}$ at i^{th} node is

$$\phi'_k(x_i) = \sum_{j=1}^N w_{ij}^1 \phi_k(x_j), \quad k \in \Delta_N \quad (12)$$

From Eq. (12) and Table 1, Eq. (10) is reduced into a tridiagonal system of equations as

$$A \bar{w}^{(1)}[i] = \bar{R}[i], \quad \forall i \in \Delta_N, \quad (13)$$

where A is the coefficient matrix is given as:

$$A = \begin{bmatrix} 6 & 1 & & & & \\ 0 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 4 & 1 \\ & & & & 1 & 4 & 0 \\ & & & & & 1 & 6 \end{bmatrix}, \quad (14)$$

$\bar{w}^{(1)}[i] = [w_{i1}^{(1)}, w_{i2}^{(1)}, \dots, w_{iN}^{(1)}]^T$ is the weighting coefficient vector corresponding to x_i , and the coefficient vector $\bar{R}[i]$ corresponding to $x_i, i \in \Delta_N$ is computed as

$$\bar{R}[1] = \begin{bmatrix} -6/h \\ 6/h \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix}, \bar{R}[2] = \begin{bmatrix} -3/h \\ 0 \\ 3/h \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \dots, \bar{R}[N-1] = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -3/h \\ 0 \\ 3/h \end{bmatrix}, \bar{R}[N] = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -6/h \\ 6/h \end{bmatrix}. \quad (15)$$

The system (13) is solved by using *Thomas algorithm* for $w_{ij}^{(1)}, i, j \in \Delta_N$. $w_{ij}^{(2)}, i, j \in \Delta_N$, are determined as in [16]:

$$w_{ij}^{(2)} = 2w_{ij}^{(1)} \left(w_{ij}^{(1)} - \frac{1}{x_i - x_j} \right), \quad \text{if } i \neq j \quad (16)$$

$$w_{ii}^{(2)} = - \sum_{i=1, i \neq j}^N w_{ij}^{(2)}, \quad \text{if } i = j$$

Table 1 Let $\varphi_j = \varphi_j(x)$, then $\varphi_j, \varphi'_j, \varphi''_j$ at x_j

	x_{j-2}	x_{j-1}	x_j	x_{j+1}	x_{j+2}
$\varphi_j(x)$	0	1	4	1	0
$\varphi'_j(x)$	0	$3/h$	0	$-3/h$	0
$\varphi''_j(x)$	0	$6/h^2$	$-12/h^2$	$6/h^2$	0

3. Implementation of Method

On substituting the approximate values of the spatial derivatives computed in Section 3, FPE (7) can be re-written as

$$\frac{\partial u(x_i, t)}{\partial t} = - \left[A_{ij} \sum_{j=1}^N w_{ij}^{(1)} u(x_j) + u_i \sum_{j=1}^N w_{ij}^{(1)} A_{ij} \right] + \left[B_{ij} \sum_{j=1}^N w_{ij}^{(2)} u(x_j) + u_i \sum_{j=1}^N w_{ij}^{(2)} B_{ij} \right], \quad i = 1, 2, \dots, N \quad (17)$$

where $A_{ij} = A(x_i, t, u_i)$, $B_{ij} = B(x_i, t, u_i)$. Eq. (17) reduced into a set of first order ODEs:

$$\frac{du_i}{dt} = L(u_i), \quad i \in \Delta_N, \quad (18)$$

with initial condition (3), where L is a nonlinear differential operator. We use the well known SSP-RK43 scheme [21] to solve the resulting system (18) through the following steps:

$$\left. \begin{aligned} u^{(1)} &= u^m + \frac{\Delta t}{2} L(u^m), \quad u^{(2)} = u^{(1)} + \frac{\Delta t}{2} L(u^{(1)}) \\ u^{(3)} &= \frac{2}{3} u^m + \frac{u^{(2)}}{3} + \frac{\Delta t}{6} L(u^{(2)}) \\ u^{m+1} &= u^{(3)} + \frac{\Delta t}{2} L(u^{(3)}) \end{aligned} \right\} \quad (19)$$

4. Numerical Experiments and Discussions

In this section, three examples of FPE having their exact solutions considered. The accuracy/efficiency of

MCB-DQM is measured by in terms of L_2 and L_∞ error norms:

$$L_\infty = \max_{j \in \Delta_N} |u_j^e - u_j^n|; L_2 = \left(\frac{\sum_{j \in \Delta_N} [u_j^e - u_j^n]^2}{\sum_{j=1}^N (u_j^e)^2} \right) \quad (20)$$

Example 1. Consider FPE as in [4, 11]

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x} + \frac{\partial^2 u}{\partial x^2}. \quad (21)$$

The initial and boundary conditions are extracted from the exact solutions (21):

$$u(x, t) = x + t. \quad (22)$$

The computational domain is $[0, 1]$. MCB-DQM results are compared with the exact results, in Table 2. L_2 and L_∞ errors, at different times, for various mesh sizes h with $\Delta t = 0.0001$, are reported in Table 3.

Table 2, 3 confirms that the proposed results are accurate. In addition to the quantitative comparisons between the MCB-DQM solutions and exact solutions presented, qualitative comparisons between the analytical and MCB-DQM solutions have been carried out at $t = 1.0$ with $h = 0.02$, $\Delta t = 0.0001$ in Fig. 1.

Example 2: Consider the following backward Kolomorov equation [11]

$$\frac{\partial u}{\partial t} = \left[-(-x-1) \frac{\partial}{\partial x} + x^2 e^t \frac{\partial^2}{\partial x^2} \right] u. \quad (23)$$

Table 2 Comparison with exact solutions by taking $h = 0.04$, $\Delta t = 0.0001$.

t	x	MCB-DQM	Exact.
0.01	0.2	0.21012	0.21010
	0.4	0.41010	0.41010
	0.6	0.61010	0.61010
	0.8	0.81012	0.81010
0.1	0.2	0.30007	0.30000
	0.4	0.50005	0.50000
	0.6	0.70006	0.70000
	0.8	0.90008	0.90000
1.0	0.2	1.20020	1.20010
	0.4	1.40020	1.40010
	0.6	1.60020	1.60010
	0.8	1.80020	1.80010

Table 3 L_2, L_∞ errors in MCB-DQM numerical solutions of Ex. 1 at different time levels for various mesh sizes $h, \Delta t = 0.0001$

t	L_2	L_∞	L_2	L_∞	L_2	L_∞
	$h = 0.04$		$h = 0.06$		$h = 0.1$	
0.25	9.1384E-05	9.8985E-05	9.0160E-05	9.8399E-05	8.8094E-05	9.7410E-05
0.50	9.7451E-05	9.9919E-05	9.6291E-05	9.9873E-05	9.4325E-05	9.9793E-05
0.75	9.7938E-05	9.9994E-05	9.6782E-05	9.9990E-05	9.4825E-05	9.9983E-05
1.00	9.7976E-05	9.9999E-05	9.6821E-05	9.9999E-05	9.4865E-05	9.9999E-05

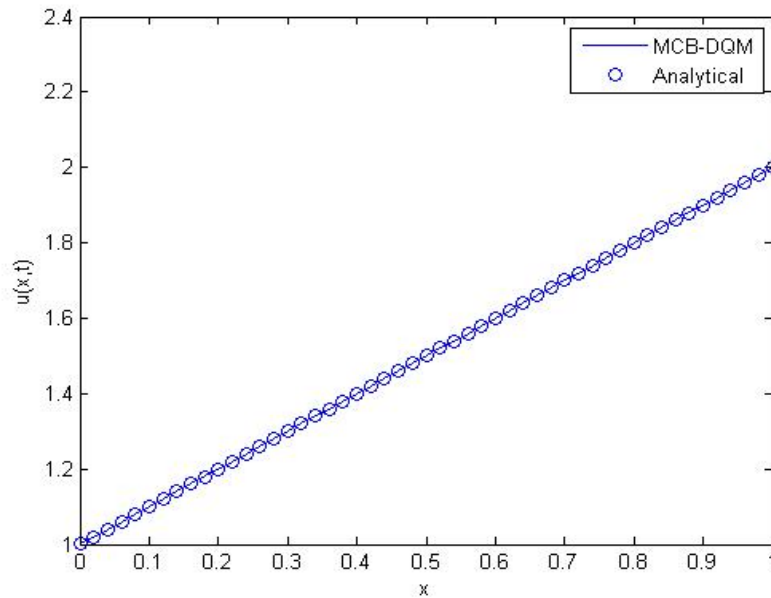


Fig 1 Analytical and MCB-DQM solutions of Ex. 1 at time $t = 1.0$ for $h = 0.02, \Delta t = 0.0001$

The initial and boundary conditions are extracted from the exact solution (24)

$$u(x, t) = (x + 1)e^t. \quad (24)$$

In this problem, we have taken the computational domain over the region $[0, 1]$. The numerical and exact solutions have been shown for various values of x and t with $h = 0.04, \Delta t = 0.0001$ in Table 4. L_2, L_∞ errors at different times for various mesh sizes h with $\Delta t = 0.0001$ are reported in Table 5. It is found that the MCB-DQM results are accurate.

In addition to the quantitative agreement between the approximate and exact solutions reported, excellent qualitative agreement between the analytical and MCB-DQM solutions have been seen at $t = 1.0$ with $h = 0.04, \Delta t = 0.0001$ in Fig. 2.

Example 3: Consider the following nonlinear

Fokker-Planck equation [4]

$$\frac{\partial u}{\partial t} = \left[-\frac{\partial}{\partial x} \left(\frac{4u(x, t)}{x} - \frac{x}{3} \right) + \frac{\partial^2}{\partial x^2} u(x, t) \right] u(x, t), \quad (25)$$

The initial and boundary conditions are extracted from the exact solutions of (25):

$$u(x, t) = x^2 e^t, \quad t \geq 0. \quad (26)$$

The numerical solutions and exact solutions are compared for domain $[0, 1]$ in terms of L_2, L_∞ errors, see Table 6. The computed and exact solutions have been shown for various values of x, t with $h = 0.04, \Delta t = 0.0001$ in Table 7. The MCB-DQM solutions are very accurate and convergent. The qualitative comparisons between analytical and MCB-DQM solutions is depicted in Fig. 3

Table 4 Comparison of solution of Ex. 2 with exact solutions for $h = 0.04$, $\Delta t = 0.0001$

t	x	MCB-DQM	Exact
0.01	0.2	1.212181	1.212181
	0.4	1.414211	1.414211
	0.6	1.616241	1.616242
	0.8	1.818298	1.818272
0.10	0.2	1.326199	1.326205
	0.4	1.547251	1.547239
	0.6	1.768342	1.768273
	0.8	1.989454	1.989308
1.00	0.2	3.262500	3.262264
	0.4	3.806286	3.805975
	0.6	4.350074	4.349686
	0.8	4.893862	4.893397

Table 5 L_2 , L_∞ errors in MCB-DQM solutions of Ex. 2 at different time levels for various mesh sizes h and $\Delta t = 0.0001$

t	L_2	L_∞	L_2	L_∞	L_2	L_∞
	$h = 0.04$		$h = 0.06$		$h = 0.1$	
0.25	1.3489E-04	2.4608E-04	1.3262E-04	2.3994E-04	1.2862E-04	2.2956E-04
0.50	1.9815E-04	3.1880E-04	1.9561E-04	3.1260E-04	1.9039E-04	3.0239E-04
0.75	2.7058E-04	4.1061E-04	2.6751E-04	4.0349E-04	2.5855E-04	3.9157E-04
1.00	3.5813E-04	5.2806E-04	3.5337E-04	5.1943E-04	3.3922E-04	5.0419E-04

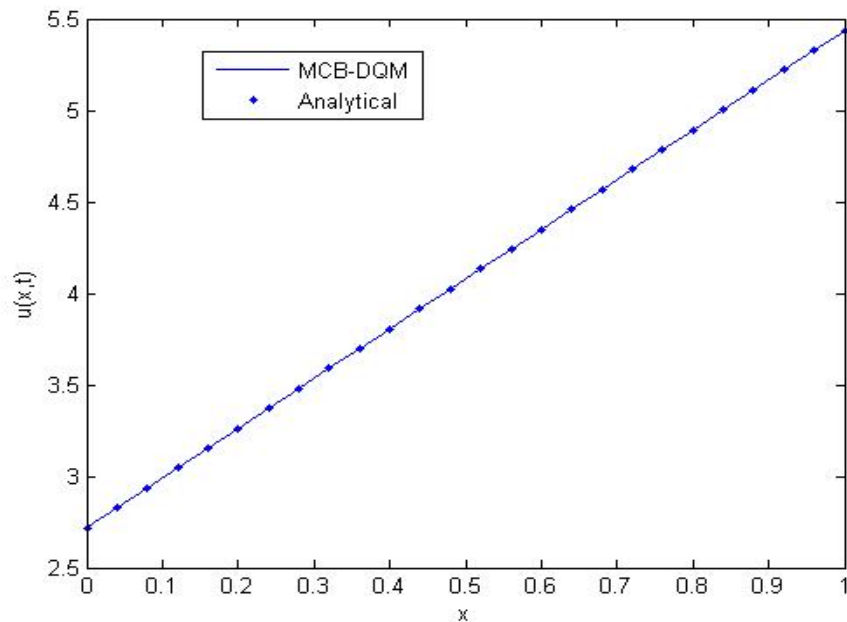
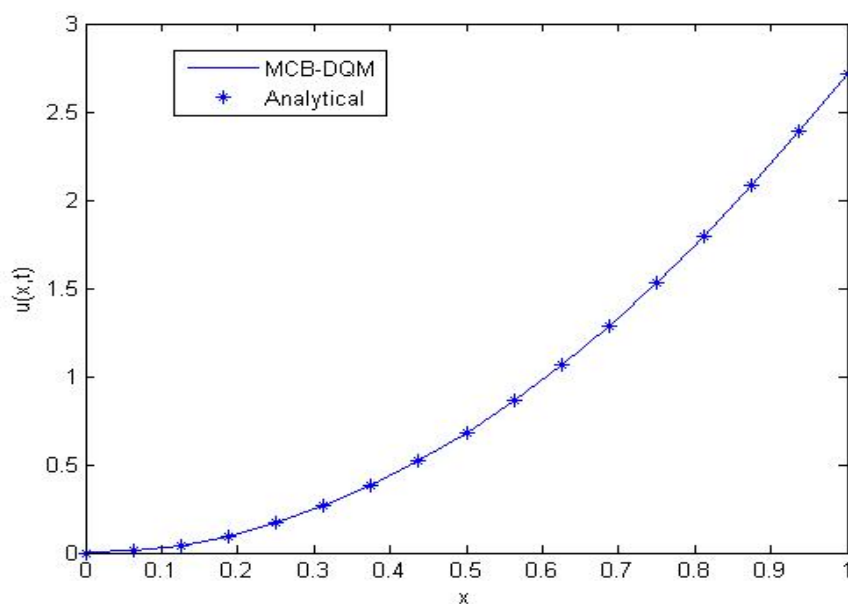
**Fig. 2** Analytical and MCB-DQM solutions of Ex. 2 at $t = 1.0$ with $h = 0.04$ and $\Delta t = 0.0001$

Table 6 L_2 , L_∞ errors in Ex. 3 at different time levels for various mesh sizes h and $\Delta t = 0.0001$

t	L_2	L_∞	L_2	L_∞	L_2	L_∞
	$h = 0.04$		$h = 0.06$		$h = 0.1$	
0.25	1.1218E-04	2.3956E-04	1.9299E-04	3.9602E-04	4.0223E-04	7.7716E-04
0.50	1.4748E-04	3.0805E-04	2.5720E-04	5.0995E-04	5.4829E-04	1.0051E-03
0.75	1.9062E-04	3.9553E-04	3.3397E-04	6.5465E-04	7.1667E-04	1.2905E-03
1.00	2.4553E-04	5.0776E-04	4.3114E-04	8.4012E-04	9.2853E-04	1.6555E-03

Table 7 Comparison for Ex. 3 with $h = 0.04$, $\Delta t = 0.0001$

t	x	MCB-DQM	Exact
0.01	0.2	0.0404060	0.0404060
	0.4	0.1616241	0.1616242
	0.6	0.3636561	0.3636544
	0.8	0.6465474	0.6464968
0.1	0.2	0.0442079	0.0442068
	0.4	0.1768446	0.1768273
	0.6	0.3979249	0.3978615
	0.8	0.7074462	0.7073094
1.0	0.2	0.1087898	0.1087421
	0.4	0.4350700	0.4349686
	0.6	0.9788895	0.9786793
	0.8	1.7402410	1.7398740

Fig 3 Analytical and MCB-DQM solutions of Ex. 3 at time $t = 1.0$ by taking $h = 0.06$ and $\Delta t = 0.0001$

5. Concluding Remarks

This paper explored the utility of a composite scheme: MCB-DQM, in space with SSP-RK43 scheme, in time for solving FPE. The efficiency and reliability of the scheme is illustrated through three numerical examples. These examples confirms: L_2 and L_∞ errors in all the cases is very small, so, a very good approximations to the exact solutions is achieved. It is demonstrates that the MCB-DQM solutions of FPE are very close to the exact solutions.

The advantage of this scheme is that it can be conveniently used to solve the linear or nonlinear physical problems and requires less storage space that causes less accumulation of numerical errors.

Acknowledgment

P. Kumar & S. N. Rai are very thankful to Babasaheb Bhimrao Ambedkar University, Lucknow INDIA, for financial assistance to carry out his research work.

References

- [1] H. Risken, The Fokker-Planck Equation Method of Solution and Applications, Springer Verlag, Berlin, Heidelberg (1989).
- [2] T.D. Frank, Stochastic feedback, nonlinear families of Markov processes, and nonlinear Fokker-Planck equations, Phys. Lett. A 331 (2004) 391–408.
- [3] G. Harrison, Numerical solution of the Fokker-Planck equation using moving finite elements'', Numer. Methods Partial Differential Equations 4 (1988) 219–232.
- [4] L. Yan, Numerical solutions of fractional Fokker-Planck equations using iterative Laplace Transform method, Abstract and applied analysis, Vol. 2013 (2013), Article Id 465160.
- [5] M. Lakestani, M. Dehghan, Numerical solution of Fokker-Planck equation using the cubic B-spline scaling functions, Numer. Methods Partial Differential Equations 25 (2009) 418–429 (2009).
- [6] V. Palleschi, F. Sarri, G. Marozzi, M.R. Torquati, Numerical solution of the Fokker-Planck equation: a fast and accurate algorithm, Phys. Lett. A 146 (1990) 378–386.
- [7] V. Palleschi, M. de Rosa, Numerical solution of the Fokker-Planck equation. II. Multidimensional case, Phys. Lett. A 163 (1992) 381–391.
- [8] S. Kazem, J.A. Rad, K. Parand, Radial basis functions methods for solving Fokker-Planck equation, Engineering Analysis with Boundary Elements, 36 (2) (2012) 181–189.
- [9] A. Sadighi, D.D. Ganji, Y. Sabzehmeidani, A study on Fokker-Planck equation by variational iteration and homotopy-perturbation methods, Int. J. Nonlinear Sci. 4 (2007) 92–102.
- [10] M. Tatari, M. Dehghan, M. Razzaghi, Application of the Adomian decomposition method for the Fokker-Planck equation, Math. Comput. Modelling 45 (2007) 639–650.
- [11] S. Hesama, A.R. Nazemia, A. Haghbinb, Analytical solution for the Fokker-Planck equation by differential transform method, Scientia Iranica B (2012) 19 (4) 1140–1145.
- [12] G. Arora, B.K. Singh, Numerical solution of Burgers' equation with modified cubic B-spline differential quadrature method, Applied Math. Comput. (2013), 224 (1), 166–177.
- [13] R. Bellman, B.G. Kashef, J. Casti, Differential quadrature: a technique for the rapid solution of nonlinear differential equations, J. Comput. Phys. 10 (1972) 40–52.
- [14] A. Korkmaz, I. Dag, Cubic B-spline differential quadrature methods and stability for Burgers' equation, Eng. Comput. Int. J. Comput. Aided Eng. Software 30 (3) (2013) 320–344.
- [15] A. Korkmaz, I. Dag, Shock wave simulations using sinc differential quadrature method, Eng. Comput. Int. J. Comput. Aided Eng. Software 28(6) (2011) 654–674.
- [16] C. Shu, B.E. Richards, Application of generalized differential quadrature to solve two dimensional incompressible navier-Stokes equations, Int. J. Numer. Meth. Fluids, 15 (1992) 791–798.
- [17] J.R. Quan, C.T. Chang, New insights in solving distributed system equations by the quadrature methods-I, Comput. Chem. Eng. 13 (1989) 779–788.
- [18] J.R. Quan, C.T. Chang, New insights in solving distributed system equations by the quadrature methods-II, Comput. Chem. Eng. 13 (1989) 1017–1024.
- [19] C. Shu, Y.T. Chew, Fourier expansion-based differential quadrature and its application to Helmholtz eigenvalue problems, Commun. Numer. Methods Eng. 13 (8) (1997) 643–653.
- [20] C. Shu, H. Xue, Explicit computation of weighting coefficients in the harmonic differential quadrature, J. Sound Vib. 204 (3) (1997) 549–555.
- [21] J.R. Spiteri, S.J. Ruuth, A new class of optimal

- high-order strong stability-preserving time-stepping schemes, SIAM J. Numer. Anal. 40 (2) (2002) 469-491.
- [22] B. K. Singh and G. Arora, A numerical scheme to solve Fisher-type reaction-diffusion equations, Nonlinear Studies/ Mesa-Mathematics in Engineering, Sciences and Aerospace 5(2) (2014) 153–164.
- [23] B. K. Singh, G. Arora, M. K. Singh, A numerical scheme for the generalized Burgers-Huxley equation, J. Egypt. Math. Society, <http://dx.doi.org/10.1016/j.joems.2015.11.003>