Solution of non-linear Fisher’s reaction-diffusion equation by using Hyperbolic B-spline based differential quadrature method

Mamta Kapoor, Varun Joshi*

Department of Mathematics, Lovely Professional University, Phagwara, Punjab, India

mamatakapoor.78@yahoo.com, varunjoshi20@yahoo.com*

Abstract. In the present paper Hyperbolic B-Spline based Differential Quadrature Method is proposed to solve the Non-Linear Fisher’s Reaction-Diffusion Equation numerically. By using the proposed method, the governing partial differential equation is converted into a system of ordinary differential equations and then SSP-RK43 scheme is implemented. Accuracy of the proposed method is checked by using discrete root mean square norm ($L_2$) and maximum error norm ($L_{\infty}$). A comparison has been made of proposed numerical scheme with the previous numerical schemes already present in the literature. The noteworthy point here is that the proposed numerical scheme is easy to use and indicates some better results.

Keywords. Differential quadrature method (DQM), Fisher’s reaction diffusion (FRD) equation, Hyperbolic B-spline, SSP-RK43 scheme

1. Introduction:

In 1937, the Fisher-Kolmogorov-Petrovsky-Piscunov equation, which is also known as Fisher KPP equation, was given by, Fisher [1] and Kolmogorov et al. [2], which afterwards known as broadly Fisher’s equation. Fisher’s equation has a vital role in the area of engineering and sciences, which are mentioned in [3, 4, 5, 6]. In the physical phenomenon Fisher’s equation is noteworthy and there is one of the special approach of reaction-diffusion equation which is as following,

\[ U_t = \lambda Uxx + \beta U (1 - U) \quad (1) \]

or

\[ \frac{\partial U}{\partial t} = \lambda \frac{\partial^2 U}{\partial x^2} + \beta U (1 - U), \quad \text{where} \quad -\infty < x < \infty, t > 0 \]

where $\beta$ is known as a real parameter, reaction term is given by $\beta U (1 - U)$ { $\beta > 0$ } and it can depend upon space variable, $AU_{xx}$ is known as diffusion term and coefficient $\lambda$ is known as a non-negative constant. Equation (1) is well-known Fisher’s equation. Kinetic advancing rate of an advantageous gene was explored by using Equation (1) [1].

In the previous decades, efforts at a large scale has been made to solve the equation (1), like Galerkin method by utilizing quartic B-spline and Crank-Nicolson method was applied to get approximate solution of equation (1) in [7], a collocation method using modified cubic B-spline (MCB-Spline) was proposed in [8], combination of quintic B-spline which was based on collocation and Crank-Nicolson technique was used in...
[9], spline approximation was used in spatial direction and finite difference (FD) approximation was used in time in [10], cubic B-spline quasi interpolation was proposed to obtain solutions of Burger’s-Fisher equation in [11], both cubic B-spline collocation method and FD techniques were used in [12], Khaled [13] applied sinc collocation method, Gazdag and Canosa [14] used an accurate space derivative, a least square method FEM (Finite Element Method) was used by Garey and Shen [15], a pseudo-spectral technique was presented by Daniel and Shizgal in [16], transient performance study of the solution and discussion of the time-asymptotic convergence of the non-linear Fisher’s equation was done by Larson [17], a new class of FD Methods for above mentioned equation was constructed by Mickens [18], a study of numerical solution of Fisher’s equation using B-spline was used by Mittal and Arora [19], Mittal and Jain [20] studied approximated solutions of non-linear FRD equation by applying collocation method which was based upon modified cubic B-spline, Mittal and Jiwari [21] used DQM for the study of FRD to get numerical approximations in, notion of moving mesh was introduced by Qiu and Sloan [22], Dag et al. [23] gave B-spline Galerkin method for proposed equation, exponential cubic B-spline method was proposed by Dag and Ersoy [24], cubic B-spline quasi-interpolation was given for proposed equation in [25]. Tamsir et al. [61] used cubic trigonometric based method to solve non-linear FRD equation, Wavelet-Galerkin approach is used to get approximated solution of above mentioned equation by Kattani and Kudrreyko [62], an exact solution to FRD equations was proposed by Wazwaz and Gorguis [63] by using Adomian Decomposition Method.

Bellman et al. [26] (in 1972) gave the basic idea of DQM which was obtained by the concept of integral quadrature. After that, the above-mentioned method of finding weighting coefficients was enhanced by Quan and Chang [27, 28] in 1989. A major headway to find the weighing coefficients was given by Shu [30] in 1990. Above mentioned methods to determine the weighting coefficients are generalized to explore the higher order approximation by Shu [30] in 1991. So far different test functions, like Lagrange polynomials, Legendre polynomial and B-spline basis functions have been used to generate different types of DQMs. A comparison was made between DQM and Harmonic DQM for buckling analysis in the case of thin isotropic plates and elastic columns by Civalik [31]. A quintic B-Spline based DQM to solve fourth order differential equation was given by Zhong [32] whereas Zhong and Lan [33] proposed spline based DQM to get the solutions of non-linear I.V.P., exponential MCB-spline functions are used as test functions in DQM by Tamsir et al. [34]. DQM based on sinc functions is used by Korkmaz and Dag [35]. Korkmaz and Dag [36] used Polynomial DQM to solve non-linear Burger’s Equation. Quartic B-spline based DQM is introduced to obtain weighting coefficients by Korkmaz et al. [37]. Arora and Singh [38] represented modified cubic B-spline based DQM (MCB-DQM) to obtain numerical solution of Burger’ equation. Arora and Joshi [39] solved one and two dimensional non-linear Burger’s equation by applying modified trigonometric cubic B-spline based DQM whereas Mittal and Dahiya [40] used MCB-spline as test function in DQM to obtain the approximate solution of 3-Dimensional hyperbolic equations. Mittal and Jiwari [41] used Polynomial DQM to get the approximate solution of non-linear Burger type equation. Jiwari et al. [42] used weighted average DQM for the solution of time dependent Burger’s equation with given initial and boundary conditions whereas Shukla et al. [43] proposed Expo-MCB-DQM for the solution of 3-Dimensional non-linear wave equation. Jiwari et al. [44] proposed polynomial DQM to get the approximate solutions of Sine -Gordon equation. A numerical study using DQM of 2-Dimensional reaction diffusion brusselator system is also given by Mittal and Jiwari [45]. DQM has been implemented to solve a different variety of 1-Dimensional and 2-Dimensional partial differential equations in the problem areas of physics, chemistry and engineering, DQM is used to find solution of kdv equation by Korkmaz [46] as kdv equation.
is used for different type of physical phenomenon like acoustic plasma waves, shallow water waves, wave phenomenon in enharmonic crystal and so on, non-linear B.V.P. has been reduced to a system of coupled non-linear O.D.E. by using GDQ (generalized differential quadrature) [47], stability and accuracy of the solution of 1-Dimensional wave problems by using iterative differential quadrature (IDQ) was used in [48], longitudinal vibration of non-uniform rods by using DQM was given by Kaisy et al. in [49], using the concept of DQM the solution of hyperbolic type heat conduction problem is given by Hsu in [50], numerical solution of thin plate problem using G-Spline based DQM was given by Mohammed and Saed [51], DQM based on modified extended cubic B-spline is used to solve 2-Dimensional hyperbolic telegraph equation by Singh and Kumar [52], hyperbolic equation is also used for vibration analysis and telegraph equation has its own importance in the branch of sciences [53], Arora et al. [66] implemented the notion of differential quadrature method to get numerical solution of 1D and 2D non-linear Schrödinger equations.

Basis Spline is a spline function which has a minimal support with respect to the given degree. We can express any spline function as a linear combination of B-splines of the given degree. The term ‘B-spline’ was basically used by ‘Isaac Jacob Schoenberg’, which is also known as basis spline. A piecewise polynomial function having degree (n-1) in a variable x is known as a spline of order n and the values of x where those pieces of polynomial are attached are known as the knots. B-spline has very important properties which helps in the creation and management of the complex shapes and the management of surfaces by using different number of knots. In literature a variety of useful PDE’s has been approximated by using a variety of B-splines functions as Mittal and Arora used cubic B-spline function in [19], Tamsir et al. [34] used expo MCB-spline function as test function, quartic B-spline basis function was used by Korkmaz et al. [37], MCB-spline based DQM was presented by Arora and Singh [38], modified trigonometric cubic B-spline based DQM was used by Arora and Joshi [39], Mittal and Dahiya used MCB-spline [40], Exponential MCB-spline was used by Shukla et al. [43].

With the time a lot of research has been taken place with an upgrading forms of B splines with different variety of methods to get the numerical results for partial differential equations, especially those partial differential equations which have a very important role in the branch of engineering and sciences. On observing the literature review and after studying a number of research papers authors got the point that still Hyperbolic B-spline has never been used with the DQM to obtain the approximated solution of the non-linear FRD equation. In this paper authors are going to represent the research work which has relevance with the solution of non-linear FRD equation by the means of cubic Hyperbolic B-spline basis function by using DQM.

In present paper, solution of non-linear Fisher’s reaction diffusion equation is given by using the modified Hyperbolic B-spline of fourth order. Present paper is categorized into different sections and subsections to understand the given concept in appropriate way. In Section 2, numerical scheme is proposed, in this section brief knowledge of differential quadrature method is presented along with Hyperbolic B-spline. In Section 2.1, hyperbolic B-spline of order 4 is defined. In Section 2.2, determination of weighting coefficients is given. In Section 2.3, present method is implemented upon the non-linear Fisher’s reaction diffusion equation. In Section 3, three numerical examples are discussed by means of tables and figures. In Section 3, accuracy and effectiveness of proposed scheme is checked by using the concept of $L_2$ and $L_{\infty}$ error norms. In Section 4, methodology discussed in paper is capsulized as conclusion.

2. Numerical Scheme (Cubic Hyperbolic B-Spline Based DQM):
DQM came into light as a very useful technique for the numerical discretization. Compare to low order FDM and FEM, DQM can obtain an acute approximated solution by implementing a smaller number of node points and that is why relatively less computational effort is required. By applying DQM, function’s derivative can be approximated at any given discrete point as a linear summation of the given functional values at knot points of complete domain or it can also be said that by employing DQM, the derivative of function at any knot can be written as the weighted linear summation of functional values at each and every given knot in the considered domain.

Let us consider a uniform grid distribution \( a = x_1 < x_2 < x_3 < \ldots < x_n = b \) of the given finite domain \([a, b]\). Where consideration is that the function \( f(x) \) is smooth enough for the complete solution domain of given problem. derivative of \( f(x) \) at any nodal point \( x_i \) can be obtained by using a linear combination of all functional values over the given solution domain, i.e.

\[
f^{(r)}_x(x_i) = \left( \frac{d^{(r)}(f)}{dx^{(r)}} \right)_{x=x_i} = \sum_{j=1}^{n} a^{(r)}_{ij} f(x_j)
\]

where \( i = 1, 2, 3, \ldots, N \) and \( r = 1, 2, 3, \ldots, N \), \( r \) is representing the order of derivative, \( a^{(r)}_{ij} \) is denoting the weighting coefficient of \( r \)th order approximating derivative and \( N \) is number of nodal points presented in the solution of domain and index \( j \) is showing the fact that \( a^{(r)}_{ij} \) is the respective weighting coefficient of functional value of \( f(x_j) \).

By using \( r = 1 \), approximation of first order derivative can be written as follows,

\[
f^{(1)}_x(x_i) = \left( \frac{df}{dx} \right)_{x=x_i} = \sum_{j=1}^{n} a^{(1)}_{ij} f(x_j)
\]

given \( i = 1, 2, 3, \ldots, N \), where \( a^{(1)}_{ij} \) is the weighting coefficient of the first order derivative.

By using \( r = 2 \), approximation of second order derivative is as follows,

\[
f^{(2)}_x(x_i) = \left( \frac{d^{(2)}(f)}{dx^{(2)}} \right)_{x=x_i} = \sum_{j=1}^{n} a^{(2)}_{ij} f(x_j)
\]

given \( i = 1, 2, 3, \ldots, N \) where \( a^{(2)}_{ij} \) is the weighting coefficient of second order derivative.

2.1. Hyperbolic B-spline:

In recent years, different new splines have been defined and used in the concept of geometrical modeling in computer aided geometry and design (CAGD). For instance, non-uniform algebraic trigonometric (NUAT) B-spline is used in [54], an orthogonal basis similar to Legendre basis are used in concept of algebraic trigonometric polynomial space by Huang and Wang [55], an orthogonal basis for NUAT spline space in [56], periodic trigonometric was used by Nouisser et al. [57], normalized spherical B-spline was used by Maes and Bultheel [58]. In the present section we will introduce the formula of hyperbolic B-spline \( H^k \) of order \( k \) which is associated with the partition \( X \) defined by,
\[
H_i^1(x) = \begin{cases} 
1 & \text{when } x_i \leq x < x_{i+1} \\
0 & \text{otherwise} 
\end{cases}
\] (5)

and for \( k > 1 \),
\[
H_i^k(x) = \frac{s(x-x_i)}{s(x_{k+1} - x_i)} H_{i+1}^{k-1}(x) + \frac{s(x_{i+k} - x)}{s(x_{i+k} - x_{i+1})} H_i^{k-1}(x)
\] (6)

where \( s(x) = \sinh(x) \), \{sine hyperbolic function of x\}

Above both equations satisfy the given properties,

(P1): For \( k \geq 2 \), \( H_i^k \in C^{k-2}(x) \)

(P2): \( H_i^k(x) \) is a piecewise hyperbolic function.

(P3): \( H_i^k(x) \geq 0 \)

(P4): Support of \( H_i^k(x) = [x_i, x_{i+k}] \)

(P5): \( H_i^k \in \Gamma_k \)

where
\[
\Gamma_k = \begin{cases} 
\text{span} \left\{ (2lx), \cosh(2lx) \right\}_{l=1}^{\frac{k-1}{2}} \cup \{1\}, & \text{where } k \text{ is odd} \\
\text{span} \left\{ \sinh((2l-1)x), \cosh((2l-1)x) \right\}_{l=1}^{\frac{k}{2}}, & \text{where } k \text{ is even} 
\end{cases}
\]

which is known as the space of the hyperbolic polynomial of order \( k \).

Let \( H_i^k \) is the Hyperbolic B-spline having order \( k \) with the given knots \( x_i \) which are uniformly distributed at \( a = x_1 < x_2 < x_3 < \ldots < x_n = b \). Thus the cubic Hyperbolic B-spline will construct a basis for all functions of \([a, b]\). The cubic hyperbolic B-spline (Hyperbolic B-spline of order 4) can be defined as the following expression,
2.2. Determination of weighting coefficients:

where \( \{H_0(x), H_1(x), \ldots, H_N(x), H_{N+1}(x)\} \) forms a basis over given domain. Values of the hyperbolic cubic B-splines and it’s derivatives at knot points are given in following table.

Table 1. (Value of \( H_i^k(x) \))

| \( x_i \) | \( x_{i+1} \) | \( x_{i+2} \) | \( x_{i+3} \) | Otherwise |
|-----------|-----------|-----------|-----------|-----------|
| \( H_i^1(x) \) | 0 | \( A \) | \( B \) | \( C \) | 0 |
| \( H_i^4(x) \) | 0 | \( D \) | \( E \) | \( F \) | 0 |

This basis hyperbolic cubic B-spline is modified in a way so that the obtained system of equations will become diagonally dominant. So the Hyperbolic cubic B-spline (basis spline) functions are modified by using the given formulae [34],

\[
\begin{align*}
\phi_1(x) &= H_1(x) + 2H_0(x) \\
\phi_2(x) &= H_2(x) - H_0(x) \\
\phi_j(x) &= H_j(x), \quad (j = 3, 4, 5, \ldots, N - 2) \\
\phi_{N-1}(x) &= H_{N-1}(x) - H_{N+1}(x) \\
\phi_N(x) &= H_N(x) + 2H_{N+1}(x)
\end{align*}
\]

(8)
Using \( r = 1 \) in equation (2) and using \( \phi_k(x) \) \( \{ k = 1, 2, 3, \ldots, N \} \) as test functions we will get,

\[
\phi'_k(x_i) = \sum_{k=1}^{N} a_{ij}^{(1)} \phi_k(x_j)
\]

where \( i, k = 1, 2, 3, \ldots, N \)

By using Table 1 and equations (8) in equation (9), a tridiagonal system of equations will be obtained, given as following,

\[
A \bar{a}^{(1)}[i] = \bar{V}^{(1)}[i], \text{ where } i = 1, 2, 3, \ldots, N
\]

and \( A \) is known as the coefficient matrix with order \( N \),

\[
A = \begin{pmatrix}
B + 2C & C & & & \\
A - C & B & C & & \\
& A - C & B & C & & \\
& & \ddots & \ddots & \ddots & \\
& & & A - C & B & C - A \\
& & & & A - C & B + 2A
\end{pmatrix}
\]

\[
\bar{a}^{(1)}[i] = \begin{pmatrix}
a_{i1}^{(1)} \\
a_{i2}^{(1)} \\
a_{i3}^{(1)} \\
\vdots \\
a_{iN-1}^{(1)} \\
a_{iN}^{(1)}
\end{pmatrix}
\]

is the given weighting coefficient vector for the node point \( x_i \) and

\[
\bar{V}[i] = \begin{pmatrix}
\phi'_{1,i} \\
\phi'_{2,i} \\
\phi'_{3,i} \\
\vdots \\
\phi'_{N-1,i} \\
\phi'_{N,i}
\end{pmatrix}
\]

is vector corresponding to the node points \( x_i \). Where
We have considered here the equation (1) with the exact solution.

Example 1. In this problem numerical solutions at different time levels have been depicted and compared with the exact solution.

We have considered here the equation (1) with $\lambda = 1$, given in [8]

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + \beta (1 - U)$$ (13)
where boundary conditions are given by:
\[
\lim_{x \to -\infty} U(x, t) = 1,
\]
\[
\lim_{x \to \infty} U(x, t) = 0
\]

Exact solution of (13) is given by
\[
U(x, t) = \frac{1}{1 + \exp\left(\frac{\beta}{\sqrt{2}}(x - \frac{1}{2} \beta t)\right)^2}
\]

The approximate solution of (13) is obtained by using the values \(a = -0.2\) and \(b = 0.8\) i.e. \([a, b] = [-0.2, 0.8]\) and at the different values of \(\beta\), like \(\beta = 2000, 5000, 10000\) with \(\Delta t = .00001\).

The obtained results are shown with the help of Figures 1, 2, and 3 for the equation (13). In Table 2 values of \(L_2\) error norm and \(L_\infty\) error norms have been shown for different values of grid points at the time levels \(t = 0.001, t = 0.002\) and \(t = 0.0025\) with \(\beta = 10000\) with \(\Delta t = 0.00001\). In Table 3, values of \(L_2\) and \(L_\infty\) error norms have been shown for for a range of grid points at the time levels \(t = 0.0025, t = 0.003\) and \(t = 0.0035\) with \(\beta = 10000\) with \(\Delta t = .00001\). In Tables 4, 5 and 6, a comparison has been made between numerically approximated value and exact value and with [19] and [20] at the different time levels for \(\beta = 10000\) and \(N = 120\) with \(\Delta t = 0.00001\) (at the place of \(\Delta t = 0.0001\)) and a good compatibility among the results has been noticed. In Table 7, we have made a comparison of results for maximum absolute error norm with [61, 64, 65]. In [61] results for \(L_\infty\) error have been obtained by using cubic trigonometric B-Spline DQM for proposed equation. In [64] results of \(L_\infty\) error have been calculated from extended MCB-Spline algorithm for non-linear Fisher’s reaction diffusion equation. In [65] results for \(L_\infty\) error have been obtained for proposed equation by using cubic B-Spline collocation method of fourth order. In Table 7, results have been compared for \(N = 40, \Delta t = .00001\) and \(\beta = 2000\).
Figure 1. \( U(x, t) \) for \( x \) where \( \beta = 2000 \) and \( t = 0.002, 0.003, 0.004, 0.005, 0.006, 0.007 \) and \( N = 200 \)
Figure 2. (U(x, t) for x at t = 0.001, 0.002, 0.003, 0.004, 0.005 for \( \beta = 5000 \) with \( N = 200 \))
Figure 3. \( U(x, t) \) for \( x \) at \( t = 0.001, 0.0015, 0.002, 0.0025, 0.003, 0.0035 \) for \( \beta = 10000 \) and\( N = 200 \)

Table 2. (Comparison between \( L_2 \) error and \( L_{\infty} \) error at time levels \( t = 0.001, 0.0015 \) and \( 0.002 \))

| \( N \) | \( L_2 \) error | \( L_{\infty} \) error | \( L_2 \) error | \( L_{\infty} \) error | \( L_2 \) error | \( L_{\infty} \) error |
|---|---|---|---|---|---|---|
| 100 | 3.40E-5 | 0.000148984 | 9.87E-5 | 0.00042713 | 0.000184642 | 0.000791347 |
| 105 | 3.13E-5 | 0.000136932 | 8.79E-5 | 0.000376261 | 0.000162582 | 0.000696421 |
| 110 | 2.91E-5 | 0.000127409 | 7.94E-5 | 0.000343105 | 0.000145173 | 0.000621478 |
| 115 | 2.73E-5 | 0.000119805 | 7.26E-5 | 0.00030995 | 0.00013129 | 0.000561684 |
| 120 | 2.60E-5 | 0.000113674 | 6.72E-5 | 0.000289729 | 0.00012011 | 0.000513513 |
| 125 | 2.48E-5 | 0.000108685 | 6.28E-5 | 0.000268607 | 0.000111026 | 0.000474358 |
| 130 | 2.39E-5 | 0.000104593 | 5.91E-5 | 0.000254515 | 0.000103586 | 0.000442271 |
| 135 | 2.32E-5 | 0.000101209 | 5.62E-5 | 0.00024072 | 9.74E-5 | 0.000415774 |
Table 3. (Comparison of $L_2$ error and $L_{\infty}$ error at time levels $t = 0.0025$, $0.003$ and $0.0035$)

| N      | $L_2$ error | $L_{\infty}$ error | $L_2$ error | $L_{\infty}$ error | $L_2$ error | $L_{\infty}$ error |
|--------|-------------|---------------------|-------------|---------------------|-------------|---------------------|
| 200    | 9.55E-5     | 0.000405218         | 0.000124687 | 0.000529286         | 0.00017481  | 0.000870629        |
| 205    | 9.43E-5     | 0.000401484         | 0.000123017 | 0.00052215          | 0.000172724 | 0.000870629        |
| 210    | 9.33E-5     | 0.000396056         | 0.000121539 | 0.000515832         | 0.000170871 | 0.000870629        |
| 215    | 9.23E-5     | 0.000393009         | 0.000120227 | 0.00051022          | 0.000169219 | 0.000870629        |
| 220    | 9.15E-5     | 0.000388823         | 0.000119058 | 0.000505223         | 0.000167742 | 0.000870629        |
| 225    | 9.08E-5     | 0.000386277         | 0.000118015 | 0.000500762         | 0.000166418 | 0.000870629        |
| 230    | 9.01E-5     | 0.000384848         | 0.000117082 | 0.000496762         | 0.000165227 | 0.000870629        |
| 235    | 8.95E-5     | 0.000383048         | 0.000116245 | 0.000493184         | 0.000164153 | 0.000870629        |

Table 4. (Comparison of Numerical and Exact solution at time level $t = 0.001$ and $t = 0.0015$)

|       | Mittal and Arora [19] | Mittal and Jain [20] | Present Method | Exact Solution | Mittal and Arora [19] | Mittal and Jain [20] | Present Method | Exact Solution |
|-------|-----------------------|----------------------|----------------|----------------|-----------------------|----------------------|----------------|----------------|
| $x$   |                       |                      |                |                |                       |                      |                |                |
| $-0.20$ | 1.00000               | 1.00000              | 1              | 1              | 1.00000              | 1.00000              | 1              | 1              |
| $-0.15$ | 1.00000               | 1.00000              | 0.99999        | 1              | 1.00000              | 1.00000              | 1              | 1              |
| $-0.10$ | 0.99999               | 0.99999              | 0.99999        | 0.99999        | 1.00000              | 1.00000              | 1              | 1              |
| $-0.05$ | 0.99994               | 0.99994              | 0.999953       | 0.99995        | 1.00000              | 1.00000              | 0.99999        | 0.99999        |
| 0.05   | 0.99633               | 0.99631              | 0.995979       | 0.99598        | 0.99994              | 0.99994              | 0.999937       | 0.999937       |
| 0.10   | 0.97203               | 0.97199              | 0.969136       | 0.96914        | 0.99956              | 0.99956              | 0.99951        | 0.99951        |
| 0.15   | 0.89085               | 0.81066              | 0.791851       | 0.79189        | 0.9655               | 0.9655               | 0.966173       | 0.966174       |
| 0.20   | 0.28644               | 0.29002              | 0.257734       | 0.25785        | 0.97347              | 0.97386              | 0.970592       | 0.970607       |
| 0.25   | 0.0158                | 0.01688              | 0.01352        | 0.01354        | 0.81756              | 0.82149              | 0.800318       | 0.800444       |
| 0.30   | 0.00032               | 0.00035              | 0.000272       | 0.00027        | 0.29819              | 0.30817              | 0.270377       | 0.270653       |
| 0.40   | 0.00000               | 0.00000              | 7.47E-8        | 7.50E-8        | 0.00034              | 0.00040              | 0.0003         | 0.000302       |
| 0.50   | 0.00000               | 0.00000              | 1.99E-11       | 1.99E-11       | 0.00000              | 0.00001              | 8.24E-8        | 8.29E-8        |
| 0.55   | 0.00000               | 0.00000              | 3.24E-13       | 3.25E-13       | 0.00000              | 0.00000              | 1.34E-9        | 1.35E-9        |
| 0.60   | 0.00000               | 0.00000              | 5.27E-15       | 5.29E-15       | 2.76E-11             | 0.00000              | 2.19E-11       | 2.20E-11       |
| 0.65   | 0.00000               | 0.00000              | 8.59E-17       | 8.63E-17       | 4.65E-13             | 0.00000              | 3.57E-13       | 3.59E-13       |
| 0.70   | 0.00000               | 0.00000              | 1.40E-18       | 1.41E-18       | 7.85E-15             | 0.00000              | 5.81E-15       | 5.85E-15       |
| \( t = 0.002 \) | \( t = 0.0025 \) |
|---|---|
| \( x \) | Mittal and Arora [19] | Mittal and Jain [20] | Present Method | Exact Solution | Mittal and Arora [19] | Mittal and Jain [20] | Present Method | Exact Solution |
| \(-0.20\) | 1.00000 | 1.00000 | 1 | 1 | 1.00000 | 1.00000 | 1 | 1 |
| \(-0.15\) | 1.00000 | 1.00000 | 1 | 1 | 1.00000 | 1.00000 | 1 | 1 |
| \(-0.10\) | 1.00000 | 1.00000 | 1 | 1 | 1.00000 | 1.00000 | 1 | 1 |
| \(-0.05\) | 1.00000 | 1.00000 | 1 | 1 | 1.00000 | 1.00000 | 1 | 1 |
| \(0.05\) | 1.00000 | 0.99999 | 1 | 1 | 1.00000 | 1.00000 | 1 | 1 |
| \(0.10\) | 0.99999 | 0.99999 | 0.99999 | 1 | 1 | 1.00000 | 1.00000 | 1 | 1 |
| \(0.15\) | 0.99995 | 0.99995 | 0.99994 | 1 | 1 | 1.00000 | 1.00000 | 0.99999 | 0.999999 |
| \(0.20\) | 0.99958 | 0.99959 | 0.999534 | 0.99999 | 1 | 1 | 1.00000 | 1.00000 | 0.99999 | 0.999993 |
| \(0.25\) | 0.99670 | 0.99679 | 0.996356 | 0.99995 | 0.99995 | 1 | 1 | 1.00000 | 1.00000 | 0.999943 |
| \(0.30\) | 0.97454 | 0.97551 | 0.971975 | 0.972007 | 0.9996 | 0.99961 | 0.99956 | 0.99957 |
| \(0.40\) | 0.30845 | 0.32607 | 0.283259 | 0.283745 | 0.97545 | 0.97701 | 0.97329 | 0.973342 |
| \(0.50\) | 0.00036 | 0.000445 | 0.00033 | 0.00033 | 0.00033 | 0.00033 | 0.31797 | 0.34396 |
| \(0.55\) | 0.000001 | 0.00001 | 5.55E-06 | 5.60E-06 | 0.01903 | 0.02347 | 0.017496 | 0.017599 |
| \(0.60\) | 1.03E-07 | 0.00000 | 9.08E-08 | 9.16E-08 | 0.00038 | 0.00051 | 0.000364 | 0.000367 |
| \(0.65\) | 1.74E-09 | 0.00000 | 1.48E-09 | 1.49E-09 | 6.48E-06 | 0.00000 | 6.12E-06 | 6.18E-06 |
| \(0.70\) | 2.93E-11 | 0.00000 | 2.41E-11 | 2.43E-11 | 1.09E-07 | 0.00000 | 1.00E-07 | 1.01E-07 |

| \( t = 0.003 \) | \( t = 0.0035 \) |
|---|---|
| \( x \) | Mittal and Arora [19] | Mittal and Jain [20] | Present Method | Exact Solution | Mittal and Arora [19] | Mittal and Jain [20] | Present Method | Exact Solution |
| \(-0.20\) | 1.00000 | 1.00000 | 00 | 1 | 1 | 1.00000 | 1.00000 | 1 | 1 |
| \(-0.15\) | 1.00000 | 1.00000 | 02 | 1 | 1 | 1.00000 | 1.00000 | 1 | 1 |
| \(-0.10\) | 1.00000 | 1.00000 | 00 | 1 | 1 | 1.00000 | 1.00000 | 1 | 1 |
| \(-0.05\) | 1.00000 | 1.00000 | 00 | 1 | 1 | 1.00000 | 1.00000 | 1 | 1 |
| \(0.05\) | 1.00000 | 1.00000 | 00 | 1 | 1 | 1.00000 | 1.00000 | 1 | 1 |
| \(0.10\) | 1.00000 | 1.00000 | 00 | 1 | 1 | 1.00000 | 1.00000 | 1 | 1 |
| t    | Tamsir et al. [61] | Shukla and Tamsir [64] | Rohilla and Mittal [65] | Present Method |
|------|-------------------|------------------------|------------------------|----------------|
| 0.0010 | 5.18E-3         | 5.18E-3                | 5.78E-4                | 9.7320E-3      |
| 0.0015 | 2.45E-3         | 2.45E-3                | 2.87E-4                | 4.2471E-3      |
| 0.0020 | 1.11E-3         | 1.11E-3                | 1.33E-4                | 1.8491E-3      |
| 0.0025 | 4.92E-4         | 4.92E-4                | 6.02E-5                | 8.0424E-4      |
| 0.0030 | 2.17E-4         | 2.17E-4                | 2.67E-5                | 3.4964E-4      |
| 0.0035 | 9.50E-5         | 9.50E-5                | 1.80E-5                | 1.5198E-4      |
| 0.004  | 4.15E-5         | 7.23E-5                | 1.14E-5                | 6.6053E-5      |

**Example 2:**

We have considered here the Fisher’s Reaction Diffusion Equation given in [19] as,

\[ U_t = \alpha U_{xx} - b U^2 + a U \]  \tag{14}
Let range of $t$ is from $0$ to $\infty$ and $x$ is chosen from $-\infty$ to $\infty$. where initial condition is as following

$$U(x, 0) = \left(\frac{-1}{4}\right) \frac{a}{b} \left[ \text{sech}^2\left(\frac{a}{\sqrt{24c}} x\right) - 2 \tanh\left(\frac{a}{\sqrt{24c}} x\right) - 2 \right]$$

and boundary conditions are given by,

$$\lim_{x \to -\infty} U(x, t) = 0.5 \quad \text{and} \quad \lim_{x \to \infty} U(x, t) = 0$$

with exact solution

$$U(x, t) = \left(\frac{-1}{4}\right) \frac{a}{b} \left[ \text{sech}^2\left(\frac{a}{\sqrt{24c}} x + \frac{5a}{12} t\right) - 2 \tanh\left(\frac{a}{\sqrt{24c}} x + \frac{5a}{12} t\right) - 2 \right]$$

In Figure 4, we have shown numerical and exact solutions with the help of graphical representation at different time levels $t = 1, 2, 3, 4$ and $5$ for $\Delta t = 0.01$ and $a = 0.5, b = 1, c = 1$. A good compatibility of both numerical and exact solutions has been obtained. In Table 8, we have obtained the $L_2$ error and $L_\infty$ error have been obtained at the time levels $t = 1, 2, 3, 4$ and $5$ for a variety of grid points. It has been observed that for every time level $L_2$ error has been decreased by increasing the number of grid points but $L_\infty$ error is same throughout the same time level but by changing the time level the $L_\infty$ error also got decreased. In Table 9, a comparison has been made with Cattani and Kudreyko [62], Mittal and Arora [19] and Mittal and Jain [20] for approximate solutions and analytical solutions at time level $t = 2$ by taking $h = 0.25, \Delta t = 0.01, a = 0.5, b = 1, c = 1$ and minimum and maximum values of $x$ are $-30$ and $30$ respectively. In [62] for Fisher’s equation a multiscale analysis was given, in [19] solution by B-spline for considered equation was given whereas in [20] solution of non-linear FRD Equation was obtained by the modified cubic B-spline collocation method. We have obtained approximate and value of exact solutions at the time level $t = 2$ and have shown the above-mentioned comparison in following Table 9. It has been observed that compatibility of the present numerical solution has a good compatibility with the previous schemes mentioned in Table 9 as well as with the exact solution. In Table 10, again a comparison is made with [62], [19] and [20] at the time level $t = 4$ by taking $h = 0.25, \Delta t = 0.01, a = 0.5, b = 1, c = 1$ where minimum and maximum values of $x$ are $-30$ and $30$ respectively. It has been observed that by this comparison that obtained approximated solutions are having good compatibility with previous results in [62, 19, 20] as well as with the exact solutions. In Table 11, a comparison is discussed for the $L_\infty$ error at time level $t = 2$. In Table 12, a comparison is discussed for the $L_\infty$ error at time level $t = 4$. 
Figure 4. Graphical representation of approximate and exact solutions at time levels $t = 1, 2, 3, 4$ and 5

Table 8. ($L_2$ error and $L_{\infty}$ error at time levels $t = 1, 2, 3, 4$ and 5)

| $n$  | $L_2$ error | $L_{\infty}$ error | $L_2$ error | $L_{\infty}$ error | $L_2$ error | $L_{\infty}$ error | $L_2$ error | $L_{\infty}$ error | $L_2$ error | $L_{\infty}$ error |
|------|-------------|--------------------|-------------|--------------------|-------------|--------------------|-------------|--------------------|-------------|--------------------|
| 200  | 1.32E-4     | 1.14E-4            | 6.26E-5     | 7.13E-5            | 7.13E-5     | 5.16E-5            | 3.38E-5     | 4.89E-5            | 3.27E-5     | 5.44E-5            |
| 210  | 1.31E-4     | 1.14E-4            | 6.26E-5     | 7.13E-5            | 7.13E-5     | 5.16E-5            | 3.38E-5     | 4.89E-5            | 3.27E-5     | 5.44E-5            |
| 220  | 1.31E-4     | 1.14E-4            | 6.26E-5     | 7.13E-5            | 7.13E-5     | 5.16E-5            | 3.38E-5     | 4.89E-5            | 3.27E-5     | 5.44E-5            |
| 230  | 1.31E-4     | 1.14E-4            | 6.26E-5     | 7.13E-5            | 7.13E-5     | 5.16E-5            | 3.38E-5     | 4.89E-5            | 3.27E-5     | 5.44E-5            |
| 240  | 1.31E-4     | 1.14E-4            | 6.26E-5     | 7.13E-5            | 7.13E-5     | 5.16E-5            | 3.38E-5     | 4.89E-5            | 3.27E-5     | 5.44E-5            |
| 250  | 1.31E-4     | 1.14E-4            | 6.26E-5     | 7.13E-5            | 7.13E-5     | 5.16E-5            | 3.38E-5     | 4.89E-5            | 3.27E-5     | 5.44E-5            |
| 260  | 1.31E-4     | 1.14E-4            | 6.26E-5     | 7.13E-5            | 7.13E-5     | 5.16E-5            | 3.38E-5     | 4.89E-5            | 3.27E-5     | 5.44E-5            |
| 270  | 1.31E-4     | 1.14E-4            | 6.26E-5     | 7.13E-5            | 7.13E-5     | 5.16E-5            | 3.38E-5     | 4.89E-5            | 3.27E-5     | 5.44E-5            |
Table 9. (Comparison of approximate and exact solutions at time level \( t = 2 \))

| \( x \) | Cattani and Kudreyko [62] | Mittal and Arora [19] | Mittal and Jain [20] | Present Method | Exact Solution |
|-------|-----------------|-----------------|-----------------|---------------|---------------|
| \(-20\) | 0.498681 | 0.498653 | 0.498652 | 0.498655 | 0.498652 |
| \(-16\) | 0.49513 | 0.495745 | 0.495741 | 0.495740 | 0.495740 |
| \(-8\) | 0.459576 | 0.459478 | 0.459477 | 0.459479 | 0.459478 |
| \(-4\) | 0.386681 | 0.386742 | 0.386787 | 0.386792 | 0.386791 |
| \(6\) | 0.041822 | 0.041877 | 0.041852 | 0.041849 | 0.041851 |
| \(10\) | 0.006455 | 0.006426 | 0.006462 | 0.006464 | 0.006465 |
| \(18\) | 7.617E–5 | 7.79E–5 | 0.000079 | 7.36447337233399E–5 | 7.91544788515397E–5 |

Table 10. (Comparison between approximated and exact solution at time level \( t = 4 \))

| \( x \) | Cattani and Kudreyko [62] | Mittal and Arora [19] | Mittal and Jain [20] | Present Method | Exact Solution |
|-------|-----------------|-----------------|-----------------|---------------|---------------|
| \(-20\) | 0.498678 | 0.499412 | 0.499413 | 0.499413 | 0.499413 |
| \(-16\) | 0.498525 | 0.498146 | 0.498142 | 0.498142 | 0.498142 |
| \(-8\) | 0.481776 | 0.481763 | 0.481756 | 0.481757 | 0.481756 |
| \(-4\) | 0.445508 | 0.445372 | 0.445395 | 0.445399 | 0.445398 |
Table 11. (Comparison for $L_{\infty}$ error at the time level $t = 2$)

| x  | Rohilla and Mittal [65] | Mittal and Arora [19] | Present |
|----|------------------------|------------------------|---------|
| -20| 1.76E-9                | 1.52E-6                | 3.9454E-8 |
| -16| 5.33E-9                | 4.56E-6                | 1.2603E-7 |
| -4 | 1.65E-7                | 4.91E-5                | 1.3202E-6 |
| 2  | 2.28E-7                | 1.61E-4                | 2.6349E-6 |
| 14 | 6.18E-9                | 9.46E-6                | 1.1151E-7 |
| 18 | 7.39E-10               | 1.23E-6                | 5.5097E-6 |

Table 12. (Comparison for $L_{\infty}$ error at time level $t = 4$)

| x  | Rohilla and Mittal [65] | Mittal and Arora [19] | Present |
|----|------------------------|------------------------|---------|
| -20| 1.78E-7                | 1.53E-6                | 3.2163E-8 |
| -16| 4.67E-7                | 4.01E-6                | 1.1001E-7 |
| -4 | 1.70E-6                | 2.53E-5                | 1.6155E-6 |
| 2  | 2.74E-6                | 1.41E-4                | 2.7432E-6 |
| 14 | 4.86E-6                | 6.29E-5                | 2.2682E-6 |
| 18 | 6.88E-7                | 1.12E-5                | 3.7171E-5 |

Example 3:

By using $\lambda = 0.1$ and $\beta = 1.0$ in equation (1) from [65], we will get

$$U_t = 0.1 \, U_{xx} + U \,(1 - U)$$  \hspace{1cm} (15)

where boundary conditions are given by, $U(a, t) = 0$ and $U(b, t) = 0$

and initial condition is taken as

$$U_0(x) = sech^2(x)$$
In following figures physical nature of this problem has been shown with the help of different graphical representation. In Figure 5, numerical solution has been shown for $x_{\text{min}} = -10$ and $x_{\text{max}} = 10$, $\Delta t = 0.05$ at different time levels from $t = 1$ to $t = 5$. It has been observed that height of the curve got increased as we changed the time level from $t = 1$ up to $t = 5$. In Figure 6, we have considered different data to show the numerical solution. Here we considered $x_{\text{min}} = -30$ and $x_{\text{max}} = 30$, $\Delta t = 0.02$ at the time levels $t = 10, 20, 30, 40, 50$. It has been noticed that curve got flatter as we changed the time levels. In Figure 7, we have tried to show the numerical solution of the given test problem for $x_{\text{min}} = -50$ and $x_{\text{max}} = 50$, $\Delta t = 0.005$ at the time levels $t = 2, 6, 10, 14, 18, 22$ and 26.

Figure 5. Graphical representation of the approximated solution at $t = 1, 2, 3, 4$ and 5

Figure 6. Pictorial representation of approximated solution at time levels $t = 10, 20, 30, 40$ and 50
Section 4. Conclusion:

In this paper a new regime known as Hyperbolic B-spline based DQM has been proposed. This method is used to get the numerical solution of the Non-linear FRD equation. FRD equation has a vast area of applications in science and engineering. Different test problems has been discussed and for the mentioned test problems $L_2$ and $L_\infty$ error norms have been evaluated. Then a comparison between the numerical approximations and the exact solutions is discussed. In the Example 1, it is noticed that $L_2$ and $L_\infty$ errors got reduced on increasing the number of grid points at different time levels. Here it is also observed that the present method has given good results when compared with methods already present in literature. In Example 2, we evaluated the $L_2$ and $L_\infty$ error norms for a range of grid points. The point noticed here is that when number of grid points were increased $L_2$ and $L_\infty$ errors got decreased. A comparison was also made for different values of $x$ and it came in notion that the results were in good agreement. For Example 3, graphical representations are given for the numerical results at different time levels with help of Figures 5, 6 and 7. So it can be said that the proposed method can be used to solve a class of partial differential equations and can be treated as a promising method to get better numerical approximations as compared to the previous methods present in literature. By using this proposed method a new dimension of exploration can be reached to obtain the approximate solutions of a variety of partial differential equations.

References

1. Fisher R A 1937 The wave of advance of advantageous genes. *Ann. Eugenics* 7(4) 355-69.
2. Kolmogorov A N, Petrovsky I G and Piskunov N S 1937 Investigation of the equation of diffusion combined with increasing of the substance and its application to a biology problem. *Bull. Moscow State Univ. Ser. A: Math. Mech* 1(6) 1-25.
3. Canosa J 2003 Diffusion in nonlinear multiplicative media *J. Math. Phys.* 10(10) 1862-68.
4. Bramson M D 1978 Maximal displacement of branching Brownian motion *Commun. Pure Appl. Math.* 31 (5) 531-81.
5. Ammerman A J and Cavalli-Sforza L L 1971 Measuring the rate of spread of early farming in Europe, *Man* 67:4-88.
6. Frank Kameneetiskii D A 1969 Diffusion and Heat transfer in chemical kinetics *1st ed. plenum press, New-York.*
7. Sahin A, Dag I and Saka B 2008 A B-spline algorithm for the numerical solution of Fisher’s equation *Kybernetes* 37 326–42.
8. Mittal R C and Jain R K 2013 Numerical solutions of nonlinear Fisher’s reaction–diffusion equation with modified cubic B-spline collocation method *Mathematical Sciences* 7(12).
9. Sahin A and Ozlem Ozmen 2014 Usage of Higher Order B-splines in Numerical Solution of Fisher’s Equation *International Journal of Nonlinear Science* 17(3) 241-53.
10. Aghamohamadi M, Rashidinia J and Ezzati R 2014 Tension spline method for solution of non-linear Fisher equation *Applied Mathematics and Computation* 249 399–407.
11. Zhu C G and Kang W S 2010 Numerical solution of Burgers–Fisher equation by cubic B-spline quasi-interpolation *Applied Mathematics and Computation* 216 (9) 2679–2686.
12. Mittal R C and Arora G 2010 Efficient numerical solution of Fisher’s equation by using B-spline method *Int. J. Comput. Math.* (87) (13) 3039–51.
13. Khaled K A 2001 Numerical study of Fisher’s reaction-diffusion equation by the Sinc collocation method *J. Comput. Appl. Math.* 137 245–55.
14. Gazdag J and Canosa J 1974 Numerical solution of Fisher’s equation *J. Appl. Probab.* 11 445–57.
15. Garey G F and Shen Y 1995 Least-squares finite element approximation of Fisher’s reaction-diffusion equation *Numer. Methods Partial Differ. Equ.* 11 175–86.
16. Daniel O and Shizgal B D 2006 A pseudospectral method of solution of Fisher’s equation *J. Comput. Appl. Math.* 193 219–42.
17. Larson D A 1978 Transient bounds and time-asymptotic behavior of solutions to nonlinear equations of Fisher’s type *SIAM J. Appl. Math.* 34 93–103.
18. Mickens R E 1994 A best finite-difference scheme for Fisher’s equation *Numer. Methods Partial Differ. Equ.* 10 581–85.
19. Mittal R C and Arora G 2010 Efficient numerical solution of Fisher’s equation by using B-spline method *Int. J. Comput. Math.* 87 (13) 3039–51.
20. Mittal R C and Jain R K 2013 Numerical solutions of nonlinear Fisher’s reaction-diffusion equation with modified cubic B-spline collocation method *Math. Sci.* 7 (12) 1–10.
21. Mittal R C and Jiwari R 2009 Numerical study of Fisher’s equation by using differential quadrature method *Int. J. Inform. Syst. Sci.* 5 (1) 143–60.
22. Qiu Y and Sloan D M 1998 Numerical solution of Fisher’s equation using a moving mesh method *J. Comput. Phys.* 146 726–46.
23. Dag I, Sahin A and Korkmaz A 2010 Numerical investigation of the solution of Fisher’s equation via the b-spline Galerkin method *Numer. Methods Partial Differ. Equ.* 26 (6) 1483–1503.
24. Dag I and Ersoy O 2016 The exponential cubic B-spline algorithm for Fisher equation *Chaos Solit. Fract.* 86 101–06.
25. Zhu C G, Kang W S 2010 Numerical solution of Burgers-Fisher equation by cubic b-spline quasi-interpolation. *Appl Math Comput* 216 2679–86.
26. Bellman R, Kashef B G and Casti J 1972 Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations *J. Comput. Phys.* 1 40–52.
27. Quan J R and Chang C T 1989 New insights in solving distributed system equations by the quadrature methods-I *Comput. Chem. Eng.* 13 779–88.
28. Quan J R and Chang C T 1989 New insights in solving distributed system equations by the quadrature methods-II *Comput. Chem. Eng.* 13 1017–24.
29. Shu C and Richards B E 1990 High resolution of natural convection in a square cavity by generalized differential quadrature in: Proceedings of third Conference on Adv. Numer. Methods Eng. Theory Appl. Swansea UK 2 978–85.
30. Shu C. 1991 Generalized differential-integral quadrature and application to the simulation of incompressible viscous flows including parallel computation Ph.D. thesis Univ. of Glasgow UK.
31. Civalek Ö 2004 Application of differential quadrature (DQ) and harmonic differential quadrature (HDQ) for buckling analysis of thin isotropic plates and elastic columns. *Engineering Structures* 26(2) 171-86.
32. Zhong H 2004 Spline-based differential quadrature for fourth order differential equations and its application to Kirchhoff plates. *Applied Mathematical Modelling* 28(4) 353-66.
33. Zhong H and Lan M (2006) Solution of nonlinear initial-value problems by the spline-based differential quadrature method. *Journal of Sound and Vibration* 296(4-5) 908-18.
34. Tamsir M, Srivastava V K and Jiwari R 2016 An algorithm based on exponential modified cubic B-spline differential quadrature method for nonlinear Burgers’ equation. *Applied Mathematics and Computation* 290 111-24.
35. Korkmaz A and Dag I 2011 Shock wave simulations using sinc differential quadrature method. *Engineering Computations* 28(6) 654-74.
36. Korkmaz A and Dag I 2011 Polynomial based differential quadrature method for numerical solution of nonlinear Burgers’ equation. *Journal of the Franklin Institute* 348(10) 2863-75.
37. Korkmaz A, Aksoy A M and Dag I 2011 Quartic B-spline differential quadrature method. *Int. J. Nonlinear Sci* 11(4) 403-11.
38. Arora G and Singh B K 2013 Numerical solution of Burgers’ equation with modified cubic B-spline differential quadrature method. *Applied Mathematics and Computation* 224 166-77.
39. Arora G and Joshi V 2018 A computational approach using modified trigonometric cubic B-spline for numerical solution of Burgers’ equation in one and two dimensions. *Alexandria Engineering Journal* 57(2) 1087-98.
40. Mittal R C and Dahiya S 2017 Numerical simulation of three-dimensional telegraphic equation using cubic B-spline differential quadrature method. *Applied Mathematics and Computation* 313 442-52.
41. Mittal R C and Jiwari R 2012 A differential quadrature method for numerical solutions of Burgers’-type equations. *International Journal of Numerical Methods for Heat & Fluid Flow* 22(7) 880-95.
42. Jiwari R, Mittal R C and Sharma K K 2013 A numerical scheme based on weighted average differential quadrature method for the numerical solution of Burgers’ equation. *Applied Mathematics and Computation* 219(12) 6680-91.
43. Shukla H S, Tamsir M, Jiwari R and Srivastava V K 2018 A numerical algorithm for computation modelling of 3D nonlinear wave equations based on exponential modified cubic B-spline differential quadrature method. *International Journal of Computer Mathematics* 95(4) 752-66.
44. Jiwari R, Pandit S and Mittal R C 2012 Numerical simulation of two-dimensional sine-Gordon solitons by differential quadrature method. *Computer Physics Communications* 183(3) 600-16.
45. Mittal R C and Jiwari R 2011 Numerical solution of two-dimensional reaction–diffusion Brusselator system. *Applied Mathematics and Computation* 217(12) 5404-15.
46. Korkmaz A 2010 Numerical algorithms for solutions of Korteweg–de Vries equation. *Numerical methods for partial differential equations* **26**(6) 1504-21.
47. Tomasiello S 1998 Differential quadrature method: application to initial-boundary-value problems. *Journal of Sound and Vibration* **218**(4) 573-85.
48. Tomasiello S 2011 Numerical stability of DQ solutions of wave problems *Numerical Algorithms* **57**(3) 289-312.
49. Al Kaisy A M, Esmaeel R A and Nassar M M 2007 Application of the differential quadrature method to the longitudinal vibration of non-uniform rods *Engineering Mechanics* **14**(5) 303-10.
50. Hsu M H 2009 Differential quadrature method for solving hyperbolic heat conduction problems *Tamkang Journal of Science and Engineering* **12**(3) 331-8.
51. Mohammed O H and Saeed M A 2019 Numerical solution of thin plates problem via differential quadrature method using G-spline. *Journal of King Saud University-Science* **31**(2) 209-14.
52. Singh B K and Kumar P 2018 An algorithm based on a new DQM with modified extended cubic B-splines for numerical study of two dimensional hyperbolic telegraph equation. *Alexandria engineering journal* **57**(1) 175-91.
53. Dehghan M and Ghesmati A 2010 Solution of the second-order one-dimensional hyperbolic telegraph equation by using the dual reciprocity boundary integral equation (DRBIE) method. *Engineering Analysis with Boundary Elements* **34**(1) 51-59.
54. Mainar E, Pena J M and Sanchez-Reyes J 2001 Shape preserving alternatives to the rational Bézier model. *Computer aided geometric design* **18**(1) 37-60.
55. Yu H and Guozhao W 2005 Constructing a quasi-Legendre basis based on the C-Bézier basis. *Progress in Natural Science* **15**(6) 559-63.
56. Wei Y W and Wang G Z 2014 An orthogonal basis for non-uniform algebraic-trigonometric spline space. *Applied Mathematics-A Journal of Chinese Universities* **29**(3) 273-82.
57. Nouissir O, Sbibih D & Sablonniere P 2003 A family of spline quasi-interpolants on the sphere. *Numerical Algorithms* **33**(1-4) 399-413.
58. Maes J and Bultheel A 2007 Modeling Genus Zero Closed Manifolds with Spherical Powell-Sabin B-splines *Computer Aided Geometric Design* **24** 79-89.
59. Shu C 2012 *Differential quadrature and its application in engineering*. Springer Science & Business Media.
60. Wang, X. 2015 *Differential quadrature and differential quadrature based element methods: theory and applications*. Butterworth-Heinemann.
61. Tamsir M, Dhimna N and Srivastava V K 2018 Cubic trigonometric B-spline differential quadrature method for numerical treatment of Fisher’s reaction-diffusion equations. *Alexandria engineering journal* **57**(3) 2019-26.
62. Cattani C and Kudreyko A 2008 Multiscale analysis of the Fisher equation in: (ed.) ICCSA 2008, Part I. Lecture Notes in Computer Science *Springer Berlin* **5072** 1171–1180.
63. Wazwaz A M and Gorguis A 2004 An analytic study of Fisher's equation by using Adomian decomposition method. *Applied Mathematics and Computation* **154**(3) 609-20.
64. Shukla H S and Tamsir M 2016 Extended modified cubic B-spline algorithm for nonlinear Fisher’s reaction-diffusion equation. *Alexandria Engineering Journal* **55**(3) 2871-79.
65. Rohila R and Mittal R C 2018 Numerical study of reaction diffusion Fisher’s equation by fourth order cubic B-spline collocation method *Mathematical Sciences* **12**(2) 79-89.
66. Arora G, Joshi V and Mittal R C (2019) Numerical Simulation of Nonlinear Schrödinger Equation in One and Two Dimensions. *Mathematical Models and Computer Simulations* **11**(4) 634-648.