Numerical regime "Uniform Algebraic Hyperbolic tension B-spline DQM" for the solution of Fisher's Reaction-Diffusion equation

Mamta Kapoor[1], Varun Joshi[2]

1. Department of Mathematics, Lovely Professional University, Punjab, India

2. Department of Mathematics, Lovely Professional University, Punjab, India

Abstract:

In present paper, "Modified Uniform Algebraic Hyperbolic (UAH) tension B-spline based Differential quadrature method" is implemented to obtain the numerical solution of Fisher's reaction-diffusion equation. In this work, modified UAH tension B-spline of order 4 is considered as basis function, in order to evaluate the required weighting coefficients in differential quadrature method. Reduced system of ordinary differential equations is solved by implementing the SSP-RK 43 scheme. Accuracy of the present scheme is checked by means of L_2 and L_∞ error norms. Present scheme is easy to implement and produced acceptable results.

Keywords: *Fisher's Reaction-Diffusion equation, UAH tension B-spline, Differential quadrature method, SSP-RK43 scheme.*

1. Introduction

1.1. Fishers' Reaction Diffusion Equation

Fisher's equation came into existence by [1], is a well-known non-linear reaction-diffusion equation, according to literature.

Fisher's reaction-diffusion equation is provided as follows:

$$\frac{\partial u}{\partial t} = \mu \frac{\partial^2 u}{\partial x^2} + \rho\, u\,(1-u) \qquad \textbf{(1)}$$

Where $x \in (-\infty, \infty)$, $t \geq 0$

Where u is a real valued function. μ is considered as non-negative constant, ρ is the real-valued parameter and $\mu \frac{\partial^2 u}{\partial x^2}$ is taken as the diffusion term and $\rho\ u(1-u)$ is given as the reaction of the mention system. Fisher [1] gave the mentioned equation for the propagation of gene in habitat. It got noticed that diffusion and non-linear local multiplication caused the growth of mutual gene population.

Initial condition for the mentioned equation is given as follows:

$$u(x, 0) = u_0(x) \in [0, 1] \quad \textbf{(2)}$$

Where x ∈ [-∞ , ∞]

Boundary conditions are presented as follows:

$$\lim_{x \to -\infty} u(x,t) = 1 \quad \textbf{(3)}$$

and

$$\lim_{x \to \infty} u(x,t) = 0 \quad \textbf{(4)}$$

Where t ≥ 0

$$\lim_{x \to \pm\infty} u(x,t) = 0 \quad \textbf{(5)}$$

t ≥ 0

Where x is the spatial variable and t is the time given. After reviewing the literature, the point came into notion that equations (1), (2) and (3) taken together are discussed as the non-linear conditions and the equations (1) and (4) taken together are known as the local conditions [2]. It is noticed that, in Fisher's equation, difficulty regarding numerical aspect can be faced as it is sensitive in sense of the correct solution basically at the boundary of right hand. Most of the numerical schemes may lead to the non-acceptable results, if the sensitive solution is not considered which is related to the initial distribution at infinite [3], [4], [5], [6].

A variety of theoretical and numerical studies exist in literature regarding solution of Fisher's reaction-diffusion equation. Canosa [7] discussed a very important characteristics of Fisher's equation in the aspect of numerical solution, that there exists stable property of the solution for the perturbation of compact support and instability of the solution occurs when perturbation tends to zero at infinity. Gazdag and Canosa [3] gave the solution of Fisher's reaction-diffusion equation in numerical aspect and showed that the process of transition from the superspeed wave into the wave of minimum speed. One more interesting characteristic is regarding the relation of steady state wave speed and the nature of the approximation at infinite, given by Larson [8] and Hagan [9]. Different researchers have provided a range of numerical methods for the approximation of Fisher's Reaction-Diffusion equation. Sahin et al. [10] gave the concept of B-spline basis function based Galerkin method for solving the Fisher's reaction-diffusion equation numerically. In this paper, quartic B-spline function was taken into account. Al-Khaled [11] presented the notion of sinc collocation method for the numerical approximation of Fisher's reaction-diffusion equation. Hari haran et al. [12] presented the notion of Harr Wavelet regime for solving Fisher's equation numerically. Sahin and Ozen [13] gave the concept of B-spline of higher order for fetching the numerical solution of Fisher's equation. Where quintic B-spline was used for the spatial discretization and for time discretization, Crank-Nicolson scheme was implemented. Mittal and Kumar [14], used the Wavelet Galerkin scheme for the numerical approximation of fisher's reaction-diffusion equation. Mittal and Arora [15]presented the notion of B-spline basis functions for the numerical of Fisher's reaction-diffusion equation. Mittal and Jain [16], gave the notion of "modified cubic B-spline based collocation scheme" for getting the numerical solution of non-linear Fisher's reaction-diffusion equation, where modified cubic B-spline basis function was used for the spatial discretization and SSP-RK54 scheme was implemented to solve the reduced system of ordinary differential equations.

1.2 Differential quadrature method:

DQM is a numerical technique to discretize the partial derivatives and lot of implementation of DQM has been tried in different fields like chemical engineering, biosciences and many more [17]. Initially DQM was introduced by Bellman and his associates [18] in 1972, for approximating the solution of partial differential equations. man concept of DQM exists in finding the weighting coefficients, partial derivatives of any smooth function can be approximated with the aid of weighting coefficients and the functional values at the given node points. Weighting coefficients of DQM can be obtained by using different test functions. Bellman et al. used Legendre polynomials and spline functions for finding the weighting coefficients [19], [18]. Quan and Chang [20] gave an explicit formulation to find the weighing coefficients in DQM. A major breakthrough in this field was given by Shu [17], Shu gave a general recurrence relation to obtain the weighting coefficients of first order as well as for the higher order derivatives. Lot of DQMs based upon a series of basis functions have been developed in order to find the numerical solution of partial differential equations. Mittal and Bhatia [21] gave the numerical approximation of Hyperbolic telegraph equation of two dimensions by using modified B-spline based DQM and used SSP-RK43 scheme for solving resulting system of ODEs. Mittal and Dahiya [22] gave the notion of modified cubic B-spline DQM for solving hyperbolic diffusion equation by using the modified cubic B-spline based DQM and used SSP-RK43 regime to solve the reduced system of ODE. Arora and Joshi [23] presented the concept of modified trigonometric cubic B-spline based DQM for solving 1D and 2D Burgers' equation and implemented SSP-RK43 method for solving ODE system. Korkmaz and Dag [24] used the notion of cubic B-spline DQM for the numerical approximation of Advection-Diffusion equation. Arora and Singh [25] gave the concept of MCB-DQM for the numerical approximation of 1D Burgers' equation and implemented SSP-RK 43 method for the solution of reduced system of ODE. Bashan et al. [26] presented the concept of modified cubic B-spline based DQM for the solution of Schrodinger equation and used SSP-RK43 method for solving reduced system of ODE. Korkmaz and Dag [27] gave the concept of Sinc DQM for obtaining the numerical approximation of non-linear Burgers' equation in this paper, Sinc test function was used in DQM to obtain weighting coefficients. Jiwari et al. [28] presented the notion of Polynomial DQM for getting the numerical approximation of 2D Sine-Gordon solitons. Present paper is divided into different sections and subsections, in order to maintain the better understanding of the proposed work.

- In Section 2, full insight upon the newly developed scheme is given.

- In Subsection 2.1, Formulae of approximating the partial derivatives by differential quadrature method are provided.
- In Subsection 2.2, a complete elaboration of UAH tension B-spline is presented.
- In Subsection 2.3, a full process of evaluating the weighting coefficients is given.
- In Subsection 2.4, implementation of the proposed scheme is given along with the formulae of SSP-RK43 scheme and corresponding error norms formulae.
- In Section 3, numerical examples are elaborated, tables and graphs are discussed for better understanding of the work.
- In Section 4, brief discussion is provided as conclusion.

2. Numerical Scheme: Modified cubic UAH tension B-spline DQM

2.1. Description of the proposed scheme

In present paper, considered number of knots are n and domain of computation is [a, b]. Node points are considered in the way like $a = x_1 < x_2 < x_3 < \ldots\ldots\ldots. < x_n = b$. The distribution of these grid points is uniform in nature and considered step size is $h = x_{i+1} - x_i$ in x-axis direction.

Partial derivative of $u(x, t)$ having m^{th} order is as follows,

$$u_x^{(m)}(x_l) = \sum_{k=1}^{n} q_{lj}^{(m)} u(x_j) \quad \textbf{(6)}$$

$$(Where\ l = 1, 2, 3, \ldots\ldots\ldots\ldots\ldots\ldots\ldots, n)$$

By putting m = 1 in Equation (6), approximation of first order partial derivative will be obtained,

$$u_x^{(1)}(x_l) = \sum_{k=1}^{n} q_{lj}^{(1)} u(x_j) \quad \textbf{(7)}$$

(Where l = 1, 2, 3, ..., n)

By putting m = 2 in Equation (6), approximation of partial derivative of second order will be obtained,

$$u_x^{(2)}(x_l) = \sum_{k=1}^{n} q_{lj}^{(2)} u(x_j) \quad \textbf{(8)}$$

$$(Where\ l = 1, 2, 3, \ldots\ldots\ldots\ldots\ldots\ldots\ldots, n)$$

Where $u_x^{(1)}(x_l)$ is the partial derivative of u of first order at x_l and $u_x^{(2)}(x_l)$ is the partial derivative of u with second order at grid point x_l. $u(x_j)$ is functional values of u at the mentioned grid points.

2.2. UAH tension B-spline

In present paper Uniform Algebraic Hyperbolic (UAH) tension B-spline [29], [30] is considered as the basis function, in order to evaluate the required weighting coefficients in Differential Quadrature Method. Domain of computation is from a to b, divided into a series with uniform length $\frac{(b-a)}{n}$, knots are given as follows, $x_l = a + lh$, $l = 0, 1, 2, 3, 4, \ldots\ldots, n$ and $a = x_0 < x_1 < x_2 < \ldots\ldots\ldots\ldots < x_n = b$. Uniform Algebraic Hyperbolic (UAH) tension B-spline of second order is defined as follows,

$$B_{l,2}(x) = \begin{cases} \frac{\sinh[\tau(x-x_{l-2})]}{\sinh(\tau h)}, x \in [x_{l-2}, x_{l-1}] \\ \frac{\sinh[\tau(x_l-x)]}{\sinh(\tau h)}, \ x \in [x_{l-1}, x_l] \\ 0, elsewhere \end{cases} \tag{9}$$

Where τ = √ η (η is a real number)

For k ≥ 3, $B_{l,k}$ is recursively defined as follows:

$$B_{l,k}(\mathrm{x}) = \int_{\infty}^{x} [\delta_{l,k-1}\ \psi_{1,k-1}(x) - \delta_{l+1,k-1}\,\psi_{l+1,k-1}]\, dx \tag{10}$$

and

$$\delta_l, j = \left(\int_{-\infty}^{\infty} \psi_{1,j}(x)dx\right)^{-1} \tag{11}$$

Uniform Algebraic Hyperbolic (UAH) tension B-spline of third order is presented as follows,

$B_{l,3}(\mathrm{x}) =$

$$\begin{cases} \left[\frac{2\delta_{l,2}}{\tau\sin h(\tau h)}\right]\left[sinh^2\left\{\frac{\tau(x - x_{l-2})}{2}\right\}\right], & [\boldsymbol{x_{i-2}}, \boldsymbol{x_{i-1}}) \\ 1 - \left[\frac{2\delta_{l,2}}{\tau\sinh(\tau h)}\right]\left[sinh^2\left\{\frac{\tau(x - x_l)}{2}\right\}\right] - \left[\frac{2\delta_{l+1,2}}{\tau\sinh(\tau h)}\right]\left[sinh^2\left\{\frac{\tau(x - x_{l-1})}{2}\right\}\right], & [\boldsymbol{x_{i-1}}, \boldsymbol{x_i}) \\ \left[\frac{2\delta_{l+1,2}}{\tau\sinh(\tau h)}\right]\left[sinh^2\left\{\frac{\tau(x - x_{l+1})}{2}\right\}\right], & [\boldsymbol{x_i}, \boldsymbol{x_{i+1}}) \\ 0, & \boldsymbol{else} \end{cases}$$

(12)

Similarly Uniform Algebraic Hyperbolic (UAH) tension B-spline of fourth order is defined as follows:

$B_{l,4}(\mathrm{x}) =$

$$
\begin{cases}
\dfrac{\delta_{l,3}\delta_{l,2}}{\tau\sinh(\tau h)}\left[(x_{l-2}-x)+\dfrac{\sinh[\tau(x-x_{l-2})]}{\tau}\right], & [\boldsymbol{x_{l-2}},\ \boldsymbol{x_{l-1}}) \\
\delta_{l,3}[\dfrac{\delta_{l,2}}{\tau\sinh(\tau h)}\left\{(x_{l-2}-x_{l-1})+\dfrac{\sinh[\tau(x_{l-1}-x_{l-2})]}{\tau}\right\}+(x-x_{l-1}) & \\
-\dfrac{\delta_{l,2}}{\tau\sinh(\tau h)}\{(x_{l-1}-x)+\dfrac{1}{\tau}(\sinh(\tau(x-x_l))+\sinh(\tau(x_l-x_{l-1}))))\} & \\
-\dfrac{\delta_{l+1,2}}{\tau\sin h(\tau h)}\left\{(x_{l-1}-x)+\dfrac{\sinh(\tau(x-x_{l-1}))}{\tau}\right\}] & \\
- & \\
\dfrac{\delta_{l+1,3}\,\delta_{l+1,2}}{\tau\sinh(\tau h)}\left\{(x_{l-1}-x)+\dfrac{\sin(\tau(x-x_{l-1}))}{\tau}\right\}, & [\boldsymbol{x_{l-1}},\boldsymbol{x_l}) \\
1-\dfrac{\delta_{l,3}\,\delta_{l+1,2}}{\tau\sinh(\tau h)}\{(x-x_{l+1})-\dfrac{\sinh(\tau(x-x_{l+1}))}{\tau}\} & \\
-\delta_{l+1,3}[\dfrac{\delta_{l+1,2}}{\tau\sin h(\tau h)}\{(x_{l-1}-x_l)+\dfrac{\sinh(\tau(x_l-x_{l-1}))}{\tau}\} & \\
+(x-x_l)-\dfrac{\delta_{l+1,2}}{\tau\sinh(\tau h)}\{(x_l-x)+\dfrac{(\sinh(\tau(x-x_{l+1}))+\sinh(\tau(x_l-x_{l+1})))}{\tau}\} & \\
-\dfrac{\delta_{l+2,2}}{\tau\sin h(\tau h)}\left\{(x_l-x)+\dfrac{\sinh(\tau(x-x_l))}{\tau}\right\}], & [\boldsymbol{x_l},\quad \boldsymbol{x_{l+1}}) \\
\dfrac{\delta_{l+1,3}\,\delta_{l+2,2}}{\tau\sin h(\tau h)}\left[(x-x_{l+2})-\dfrac{\sinh(\tau(x-x_{l+2}))}{\tau}\right], & [\boldsymbol{x_{l+1}},\quad \boldsymbol{x_{l+2}}) \\
0, & \boldsymbol{otherwise}
\end{cases}
$$

(13)

Table 1: Table for the values of UAH tension B splines of order 4 i.e. $B_{l,4}(x)$ and $B_{l,4}'(x)$ at different node points is given below:

	$\boldsymbol{x_{l-2}}$	$\boldsymbol{x_{l-1}}$	$\boldsymbol{x_l}$	$\boldsymbol{x_{l+1}}$	$\boldsymbol{x_{l+2}}$
$\boldsymbol{B_{l,4}(x)}$	0	n_1	n_2	n_3	0
$\boldsymbol{B'_{l,4}(x)}$	0	n_4	0	n_5	0

In order to improve the results, modified Uniform Algebraic Hyperbolic (UAH) tension B-spline can be implemented in the way such that the resultant matrix system will be diagonally dominant [25]. Where by implementing following set of equations improvised values can be obtained.

$$\begin{cases} MB_1(x) = B_1(x) - 2\,B_0(x) \\ MB_2(x) = B_2(x) - B_0(x) \\ MB_l(x) = B_l(x), [l = 3, 4, 5, \dots\dots\dots\dots\dots\dots\dots, n-2] \\ MB_{n-1}(x) = B_{n-1}(x) - B_{n+1}(x) \\ MB_n(x) = B_n(x) - 2\,B_{n+1}(x) \end{cases}$$

(14)

2.3. Process of finding weighting coefficients

We have the formula to approximate first order derivative i.e. given as follows,

$$MB_k^{(1)}(x_l) = \sum_{j=1}^{n} q_{lj}^{(1)} MB_k\,(x_j) \quad \textbf{(15)}$$

(where l = 1, 2 , 3, ……….., n) and (k = 1, 2 , 3, ……….., n)

At grid point x_1 : By applying the formulas from equations (13) and (14) and the values from Table 1 in the equation (15), we will get the following set of equations, for different values of k,

For k = 1:

$$MB_1^{(1)}(x_1) = \sum_{j=1}^{n} q_{1j}^{(1)} MB_1\left(x_j\right) = q_{11}^{(1)}\,[n_2 + 2\,n_3] + q_{12}^{(1)}[n_3]$$

For k = 2:

$$MB_2^{(1)}(x_1) = \sum_{j=1}^{n} q_{1j}^{(1)} MB_2\left(x_j\right) = q_{11}^{(1)}\,[n_1 - n_3] + q_{12}^{(1)}[n_2] + q_{13}^{(1)}\,[n_3]$$

For k = 3:

$$MB_3^{(1)}(x_1) = \sum_{j=1}^{n} q_{1j}^{(1)} MB_3\left(x_j\right) = q_{12}^{(1)}[n_1] + q_{13}^{(1)}\,[n_2] + q_{14}^{(1)}[n_3]$$

……………………………………………….

……………………………………………….

……………………………………………….

For k = n:

$$MB_n^{(1)}(x_1) = \sum_{j=1}^{n} q_{1j}^{(1)} MB_n\left(x_j\right) = q_{1,n-1}^{(1)}[n_1] + q_{1,n}^{(1)}\,[n_2 + 2\,n_1]$$

From above set of equation at grid point x_1 and for the values of k = 1, 2, 3,……., n, we will obtain the following tridiagonal system of algebraic equations:

$$N_1\,\vec{q}^{(1)}[l] = \vec{S}[l], \text{ Where l = 1, 2, 3, ..., n}$$

$$N_1 = \begin{pmatrix} n_2 + 2\,n_3 & n_3 & & & & & & \\ n_1 - n_3 & n_2 & n_3 & & \cdots & & & \\ & n_1 & n_2 & n_3 & & & & \\ & \vdots & & & \ddots & & \vdots & \\ & & & & n_1 & n_2 & n_3 & 0 \\ & & & & \cdots & n_1 & n_2 & n_3 - n_1 \\ & & & & & & n_1 & n_2 + 2\,n_1 \end{pmatrix}$$

$$\vec{q}^{(1)}[1] = \begin{pmatrix} q_{1,1}^{(1)} \\ q_{1,2}^{(1)} \\ q_{1,3}^{(1)} \\ \vdots \\ \vdots \\ q_{1,n-1}^{(1)} \\ q_{1,n}^{(1)} \end{pmatrix}, \vec{S}[1] = \begin{pmatrix} MB_1'(x_1) \\ MB_2'(x_1) \\ MB_3'(x_1) \\ \vdots \\ \vdots \\ MB_{n-1}'(x_1) \\ MB_n'(x_1) \end{pmatrix} = \begin{pmatrix} 2\,n_5 \\ n_4 - n_5 \\ \vdots \\ 0 \end{pmatrix}$$

At grid point x_2: By applying the formulas from equations (13) and (14) and the values from Table 1 in the equation (15), we will get the following set of equations, for different values of k,

$$N_1 = \begin{pmatrix} n_2 + 2\,n_3 & n_3 & & & & & & \\ n_1 - n_3 & n_2 & n_3 & & \cdots & & & \\ & n_1 & n_2 & n_3 & & & & \\ & \vdots & & & \ddots & & \vdots & \\ & & & & n_1 & n_2 & n_3 & 0 \\ & & & & \cdots & n_1 & n_2 & n_3 - n_1 \\ & & & & & & n_1 & n_2 + 2\,n_1 \end{pmatrix}$$

$$\vec{q}^{(1)}[2] = \begin{pmatrix} q_{2,1}^{(1)} \\ q_{2,2}^{(1)} \\ q_{2,3}^{(1)} \\ \vdots \\ \vdots \\ q_{2,n-1}^{(1)} \\ q_{2,n}^{(1)} \end{pmatrix}, \vec{S}[2] = \begin{pmatrix} MB_1'(x_2) \\ MB_2'(x_2) \\ MB_3'(x_2) \\ \vdots \\ \vdots \\ MB_{n-1}'(x_2) \\ MB_n'(x_2) \end{pmatrix} = \begin{pmatrix} n_5 \\ 0 \\ n_4 \\ \vdots \\ \\ 0 \end{pmatrix}$$

....

....

....

At grid point x_n: By applying the formulas from equations (13) and (14) and the values from Table 1 in the equation (15), we will get the following set of equations, for different values of k,

$$N_1 = \begin{pmatrix} n_2 + 2\,n_3 & n_3 & & & & & & & \\ n_1 - n_3 & n_2 & n_3 & & \cdots & & & & \\ & n_1 & n_2 & n_3 & & & & & \\ & \vdots & & & \ddots & & & \vdots & \\ & & & & & n_1 & n_2 & n_3 & 0 \\ & & & & \cdots & & n_1 & n_2 & n_3 - n_1 \\ & & & & & & & n_1 & n_2 + 2\,n_1 \end{pmatrix}$$

$$\vec{q}^{(1)}[n] = \begin{pmatrix} q_{n,1}^{(1)} \\ q_{n,2}^{(1)} \\ q_{n,3}^{(1)} \\ \vdots \\ \vdots \\ q_{n,n-1}^{(1)} \\ q_{n,n}^{(1)} \end{pmatrix}, \vec{S}[2] = \begin{pmatrix} MB_1'(x_n) \\ MB_2'(x_n) \\ MB_3'(x_n) \\ \vdots \\ \vdots \\ MB_{n-1}'(x_n) \\ MB_n'(x_n) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \\ n_5 - n_4 \\ n_4 \end{pmatrix}$$

Similarly in order to find the weighting coefficients of order r where r ≥ 2, is given as following from , given as follows [17],

$$q_{ij}^{(m)} = \mathrm{m}\left[q_{lj}^{(1)}\, q_{ll}^{(m-1)} - \frac{q_{lj}^{(m-1)}}{x_l - x_j}\right] \text{ for } l \neq j \qquad \textbf{(16)}$$

$$\{l = 1,2,3,\ldots.,n\}\ and\ \{m = 2,3,\ldots\ldots,n-1\}$$

$$q_{ll}^{(m)} = -\sum_{j=1,\ j\neq l}^{N} q_{lj}^{(l)} \qquad \text{for } l = j \tag{17}$$

So with the help of above-mentioned equations 2^{nd} and higher order weighting coefficients can be easily obtained.

2.4. Implementation of the proposed scheme

By implementing the formulae given in equations (7) and (8) into equation (1), following equation will be obtained,

$$\frac{du_l}{dt} = \mu \sum_{k=1}^{n} q_{lj}^{(2)}\, u(x_j)\, \rho\, u_l(1-u_l) \tag{18}$$

Where $l, j = 1,2,3,4,5,\ldots\ldots\ldots,n$

The resultant system of ordinary differential equations will be solved by implementing the SSP-RK 43 regime, formulae given as follows [25] and [31]:

$$u^{(1)} = u^m + \frac{\Delta t}{2} L(u^m)$$

$$u^{(2)} = u^{(1)} + \frac{\Delta t}{2} L(u^{(1)})$$

$$u^{(3)} = \frac{2}{3} u^m + \frac{u^{(2)}}{3} + \frac{\Delta t}{6} L(u^{(2)})$$

$$u^{(m+1)} = u^{(3)} + \frac{\Delta t}{2} L(u^{(3)})$$

Error norms are considered as follows:

$$L_2 = \left[h \sum_{j=1}^{n} \{u_{exact} - u_{numerical}\}^2 \right]^{\frac{1}{2}}$$

$$L_\infty = \max |u_{exact} - u_{numerical}|$$

3. Numerical Examples and discussion

In present example Fisher's equation is considered with λ = 1 [15], [16].

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \beta\, u(1-u)$$

Computational Domain: [-0.2, 0.8], Where boundary conditions are:

$$\lim_{x\to-\infty} u(x,t) = 1$$

and

$$\lim_{x\to\infty} u(x,t) = 0$$

Exact solutions is provided as follows:

$$u(x,t) = \frac{1}{\left\{1+\exp\left(\sqrt{\left(\frac{\beta}{6}\right)}\; x - \frac{5\beta}{6}\; t\right)\right\}^2}$$

and, initial condition can be easily fetched with the aid of given Exact solution as follows:

$$u(x,0) = \frac{1}{\left\{1+\exp\left(\sqrt{\left(\frac{\beta}{6}\right)}\; x\right)\right\}^2}$$

In Figure 1, exact and numerical solutions are compared graphically for the number of grid points N = 101, Δ t = 0.00001, τ = 0.8, β = 2500 at the time levels t = 0.001, 0.002, 0.003, 0.004 and 0.005 respectively. It is noticed that exact and numerical solutions are in good agreement for the mentioned data. In Figure 2, exact and numerical results are matched with each other for N = 51, Δ t = 0.00001, τ = 0.5 at the time level t = 0.008, where β = 500, 1000, 1500 and 2000 respectively. In Figure 3, exactness of the solution is compared with numerical solution graphically for N = 51, Δ t = 0.00001, τ = 0.5 at the time levels t = 0.0015, 0.0025, 0.0035 and 0.0045 respectively along with β = 3000. In Table 2, L_2 and L_∞ error norms are evaluated for different number of grid points at the time levels t = 0.0055 and 0.0065 respectively for β = 2000. In Table 3, L_2 and L_∞ error norms are evaluated at the mentioned time levels for the number of grid points N = 101 and 201 respectively for β = 10000. In Table 4, numerical and exact solutions are matched for β = 10000 at the time levels t = 0.001, 0.002 and 0.003 respectively.

Example 2:

In this example, following form of Fisher's equation is considered [15], [16],

$$\frac{\partial u}{\partial t} = \alpha u_{xx} + A\,u - B\,u^2$$

Computational Domain: [-30, 30]

Where Exact solution is also mentioned as follows:

$$u(x,t) = -\frac{1}{4}\frac{A}{B}[\operatorname{sech}^2\left(-\frac{\sqrt{A}}{24C}x + \frac{5A}{12}t\right) - 2\tanh\left(-\frac{\sqrt{A}}{24C}x + \frac{5A}{12}t\right) - 2]$$

Initial condition can be easily obtained from the given Exact solution.

Initial condition is:

$$u(x,t) = -\frac{1}{4}\frac{A}{B}[\operatorname{sech}^2\left(-\frac{\sqrt{A}}{24C}x\right) - 2\tanh\left(-\frac{\sqrt{A}}{24C}x\right) - 2]$$

Boundary conditions are mentioned below:

$$\lim_{x\to-\infty} u(x,t) = 0.5$$

$$\lim_{x\to\infty} u(x,t) = 0$$

In Table 5, present solution is matched with [32] and [15] along with the exact solution and absolute error is also provided at time level t = 2, present numerical solutions is almost same with the compared ones and with the exact solution. In Table 6, present numerical solutions are compared with [32] and [15] as well as with exact solution at time level t = 4. In Figures 4, 5 and 6, graphical representation of the exact and numerical solutions is provided at the mentioned parameters, in all these figures, a good compatibility of exact and numerical solution is obtained.

4. Conclusion:

Present work is based upon finding the approximated solution of Fisher's reaction-diffusion equation, for this purpose "Modified UAH tension B-spline based DQM" is implemented. After the spatial discretization of the partial derivatives by using modified UAH tension B-spline as basis function, a system of ordinary differential equations obtained. Reduced system of ODE is dealt by using SSP-RK 43 scheme. Present numerical solution are compared with the existing results as well as with the exact solution. By means of graphs a good compatibility of exact and numerical solutions is obtained. Present scheme is accurate, easy to implement and produced better results.

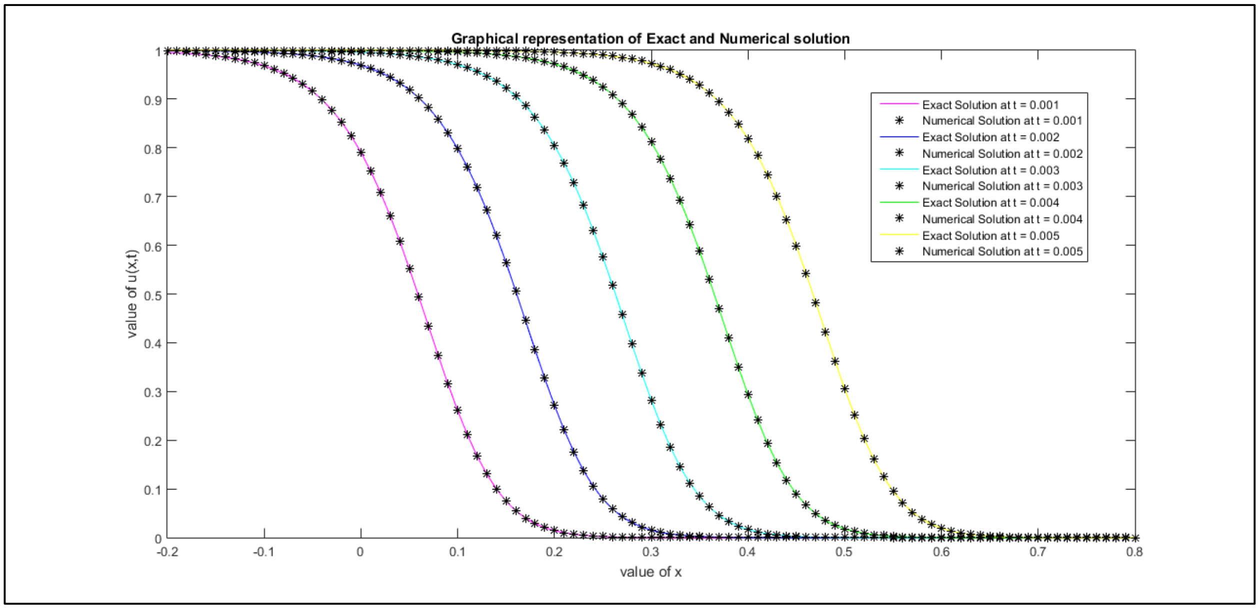

Figure 1: Graphical representation of Exact and Numerical solution for N = 101, Δ t = 0.00001, τ = 0.8, β = 2500 at the time levels t = 0.001, 0.002, 0.003, 0.004 and 0.005 respectively for Example 1

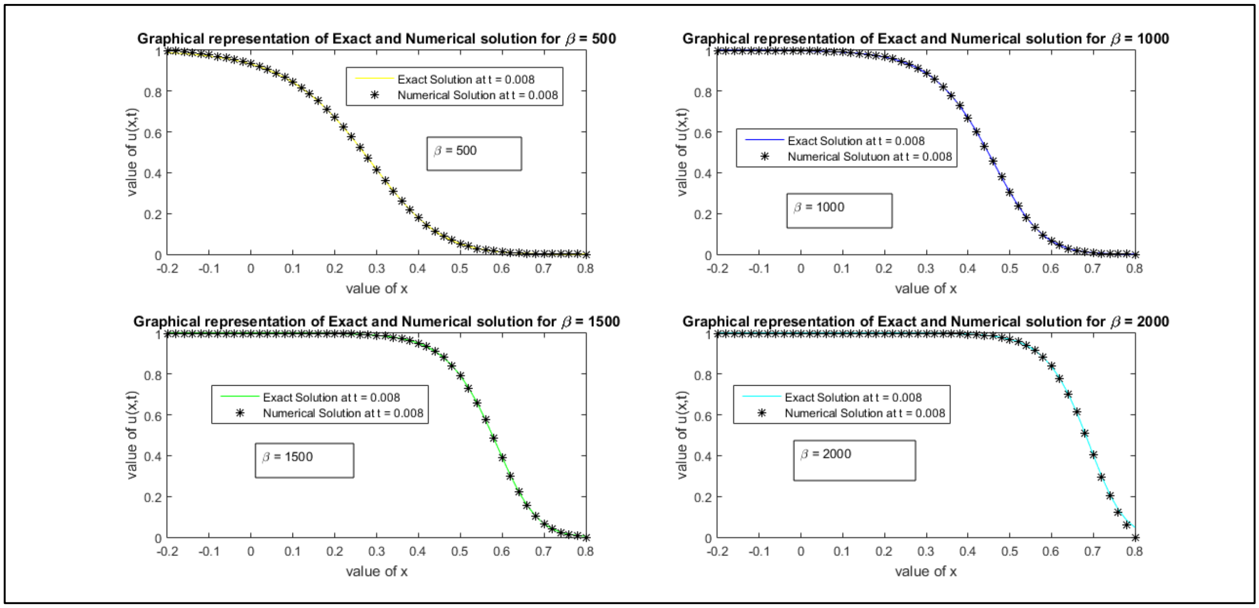

Figure 2: Graphical representation of Exact and Numerical solution for N = 51, Δ t = 0.00001, τ = 0.5 at the time level t = 0.008 for β = 500, β = 1000, β = 1500 and β = 2000 respectively for Example 1

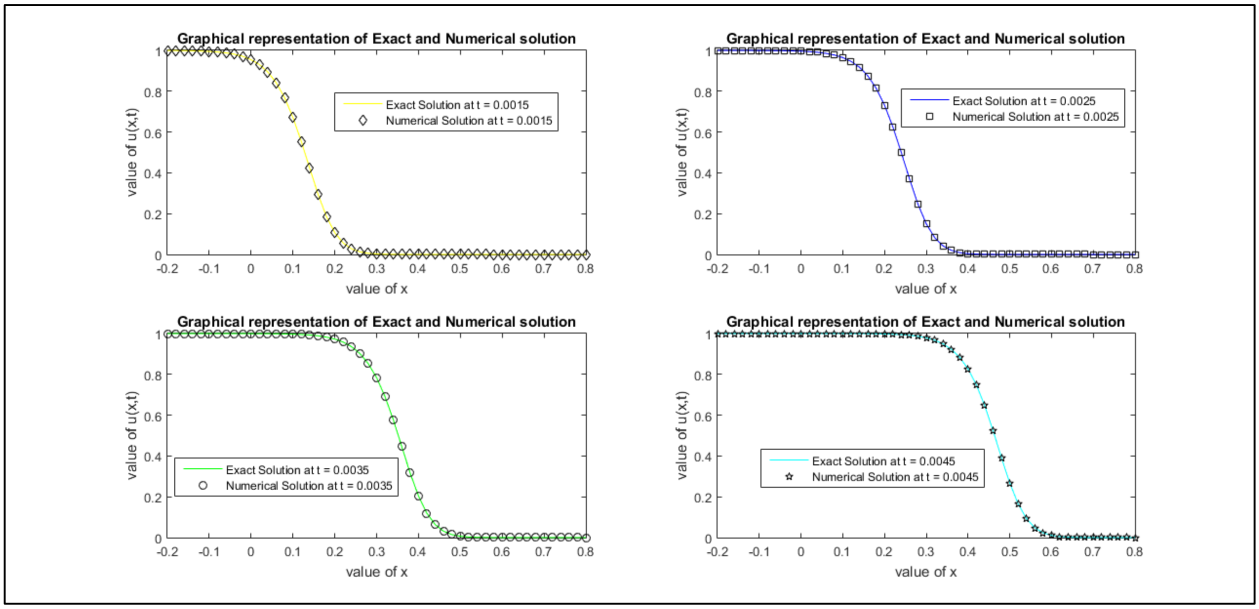

Figure 3: Graphical representation of Exact and Numerical solution for N = 51, Δ t = 0.00001, τ = 0.5 at the time levels t = 0.0015, 0.0025, 0.0035 and 0.0045 respectively for β = 3000 for Example 1

Table 2: L_2 and L_∞ error norms at time levels t = 0.005 and 0.0065 respectively at the mentioned time levels for β = 2000, Δ t = 0.00001 and tension parameter τ = 0.5 for Example 1

Number of grid points	t = 0.0055 L_2 error	t = 0.0055 L_∞ error	t = 0.0065 L_2 error	t = 0.0065 L_∞ error
50	2.33E-05	6.98E-05	8.93E-05	5.05E-04
70	6.73E-06	1.87E-05	7.33E-05	5.05E-04
90	3.62E-06	1.87E-05	6.90E-05	5.05E-04
110	2.91E-06	1.87E-05	6.68E-05	5.05E-04
130	2.70E-06	1.87E-05	6.55E-05	5.05E-04
150	2.61E-06	1.87E-05	6.45E-05	5.05E-04

Table 3: Description of L_2 and L_∞ Error norms at different time levels for N = 101 and N = 201 respectively, where Δ t = 0.00001, τ = 0.007 and β = 10000 for Example 1

Time level	N = 101 L_2 error	N = 101 L_∞ error	N = 201 L_2 error	N = 201 L_∞ error
0.001	3.35E-05	1.49E-04	1.95E-05	8.46E-05
0.002	1.80E-04	7.64E-04	6.74E-05	2.88E-04
0.003	3.78E-04	1.61E-03	1.24E-04	5.27E-04

0.0015	9.66E-05	4.14E-04	4.16E-05	1.78E-04
0.0025	2.76E-04	1.17E-03	9.53E-05	4.05E-04
0.0035	4.95E-04	2.06E-03	1.74E-04	8.71E-04

Table 4: Comparison of Numerical and Exact Solution at different time levels for $\tau = 0.007$ and $\beta = 10000$ for Example 1

x	t = 0.001	t = 0.001	t = 0.002	t = 0.002	t = 0.003	t = 0.003
	Numerical	**Exact**	**Numerical**	**Exact**	**Numerical**	**Exact**
2.00E-02	9.99E-01	9.99E-01	1	1.00E	1.00E	1.00E
3.00E-02	9.98E-01	9.98E-01	1	1.00E	1.00E	1.00E
4.00E-02	9.98E-01	9.98E-01	1	1.00E	1.00E	1.00E
5.00E-02	9.96E-01	9.96E-01	1	1.00E	1.00E	1.00E
6.00E-02	9.94E-01	9.94E-01	1	1.00E	1.00E	1.00E
7.00E-02	9.92E-01	9.92E-01	1	1.00E	1.00E	1.00E
8.00E-02	9.88E-01	9.88E-01	1	1.00E	1.00E	1.00E
9.00E-02	9.81E-01	9.81E-01	1.00E	1.00E	1.00E	1.00E
6.00E-01	9.11E-15	9.17E-15	1.56E-07	1.59E-07	3.87E-01	3.89E-01
6.10E-01	4.03E-15	4.05E-15	6.92E-08	7.01E-08	2.73E-01	2.75E-01

Table 5: Comparison of Numerical solution and Exact solution along with description of Absolute Error norms at $t = 2$, $\tau = 0.005$, $\Delta t = 0.0001$, $N = 101$, $A = 0.5$, $B = 1$ and $C = 1$ for Example 2

x	Cattani et al. [11]	Mittal and Arora [22]	Present Scheme	Exact Solution	Absolute Error
-20	4.99E-01	0.498653	4.99E-01	4.99E-01	3.38E-10
-16	0.49513	0.495745	4.96E-01	4.96E-01	3.64E-10
-12	0.486758	0.486679	4.87E-01	4.87E-01	2.20E-09
-8	0.459576	0.459478	4.64E-01	4.64E-01	9.83E-09
2	0.158878	0.159011	1.43E-01	1.43E-01	1.09E-07
6	0.041822	0.041877	4.19E-02	4.19E-02	5.26E-08
10	0.006455	0.006426	5.83E-03	5.83E-03	6.25E-09
14	0.00075	0.000746	6.05E-04	6.05E-04	4.75E-09
18	7.62E-05	7.79E-05	7.92E-05	7.92E-05	1.07E-09

Table 6: Comparison of Numerical solution and Exact solution along with description of Absolute Error norms at $t = 4$, $\tau = 0.005$, $\Delta t = 0.0001$, $N = 101$, $A = 0.5$, $B = 1$ and $C = 1$ for Example2

x	Cattani et al. [11]	Mittal and Arora [22]	Present Scheme	Exact Solution	Absolute Error
-2.00E+01	4.99E-01	4.99E-01	4.99E-01	4.99E-01	1.64E-08
-16	4.99E-01	4.98E-01	4.98E-01	4.98E-01	3.86E-10
-12	4.95E-01	4.94E-01	4.94E-01	4.94E-01	1.03E-09
-8	4.82E-01	4.82E-01	4.84E-01	4.84E-01	5.52E-09
2	2.79E-01	2.80E-01	2.63E-01	2.63E-01	4.73E-08
6	1.17E-01	1.17E-01	1.17E-01	1.17E-01	5.60E-08
10	2.59E-02	2.59E-02	2.37E-02	2.37E-02	9.33E-08
14	3.70E-03	3.56E-03	2.93E-03	2.93E-03	1.39E-08
1.80E+01	4.09E-04	3.95E-04	4.06E-04	4.06E-04	7.77E-09

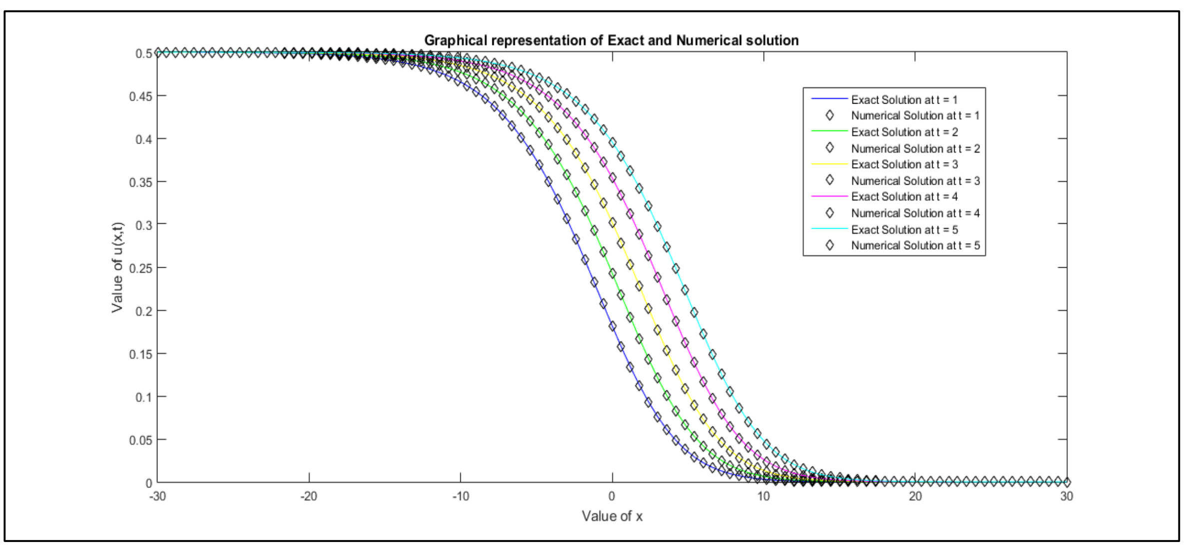

Figure 4: Graphical representation of Exact and Numerical solution for N = 101, Δt = 0.0001, τ = 0.007 at the time levels t = 1, 2, 3, 4 and 5 respectively, Where A = 0.5, B = 1 and C = 1 for Example 2

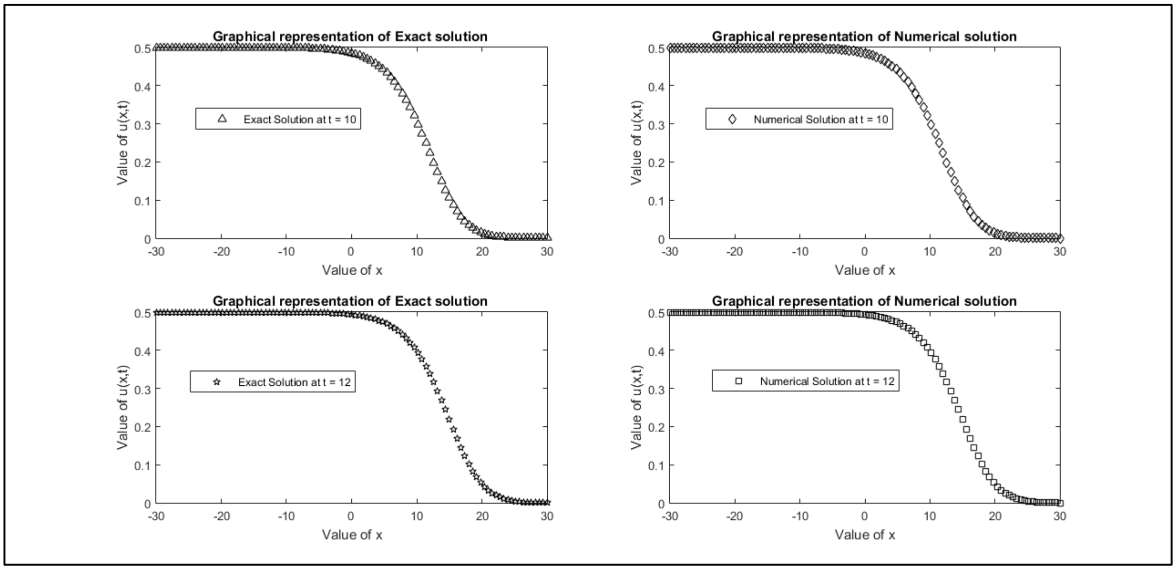

Figure 5: Graphical representation of Exact and Numerical solution for N = 101, Δ t = 0.0001, τ = 0.5 at the time levels t = 10 and 12 respectively, Where A = 0.5, B = 1 and C = 1 for Example 2

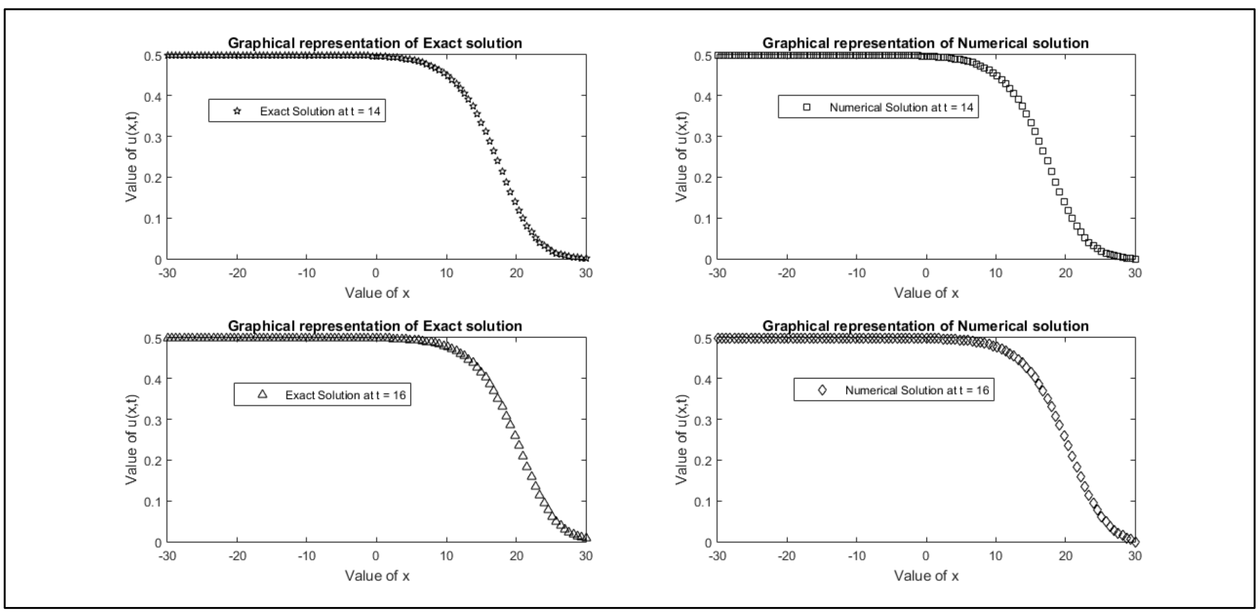

Figure 6: Graphical representation of Exact and Numerical solution for N = 101, Δ t = 0.0001, τ = 0.5 at the time levels t = 14 and 16 respectively, Where A = 0.5, B = 1 and C = 1 for Example 2

References:

1. Al-Khaled, K. (2001). Numerical study of Fisher's reaction–diffusion equation by the Sinc collocation method. *Journal of Computational and Applied Mathematics*, *137*(2), 245-255.

2. Alinia, N., & Zarebnia, M. (2018). A new tension B-spline method for third-order self-adjoint singularly perturbed boundary value problems. *Journal of Computational and Applied Mathematics*, *342*, 521-533.

3. Alinia, N., & Zarebnia, M. (2019). A numerical algorithm based on a new kind of tension B-spline function for solving Burgers-Huxley equation. *Numerical Algorithms*, *82*(4), 1121-1142.

4. Arora, G., & 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), 1087-1098.

5. Singh, B. K., & Arora, G. (2014). A numerical scheme to solve Fisher-type reaction-diffusion equations. *Nonlinear Studies/Mesa-Mathematics in Engineering, Science and Aerospace*, *5*(2), 153-164.

6. Bashan, A., Yagmurlu, N. M., Ucar, Y., & Esen, A. (2017). An effective approach to numerical soliton solutions for the Schrödinger equation via modified cubic B-spline differential quadrature method. *Chaos, Solitons & Fractals*, *100*, 45-56.

7. Bellman, R., Kashef, B., Lee, E. S., & Vasudevan, R. (1975). Solving hard problems by easy methods: differential and integral quadrature. *Computers & Mathematics with Applications*, *1*(1), 133-143.

8. Bellman, R., Kashef, B. G., & Casti, J. (1972). Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations. *Journal of computational physics*, *10*(1), 40-52.

9. Canosa, J. (1973). On a nonlinear diffusion equation describing population growth. *IBM Journal of Research and Development*, *17*(4), 307-313.

10. Carey, G. F., & Shen, Y. (1995). Least- squares finite element approximation of Fisher's reaction–diffusion equation. *Numerical Methods for Partial Differential Equations*, *11*(2), 175-186.

11. Cattani, C., & Kudreyko, A. (2008). On the discrete harmonic wavelet transform. *Mathematical Problems in Engineering*, *2008*.

12. Fisher, R. A. (1937). The wave of advance of advantageous genes. *Annals of eugenics*, *7*(4), 355-369.

13. Gazdag, J., & Canosa, J. (1974). Numerical solution of Fisher's equation. *Journal of Applied Probability*, *11*(3), 445-457.

14. Hagan, P. S. (1982). Traveling wave and multiple traveling wave solutions of parabolic equations. *SIAM Journal on Mathematical Analysis*, *13*(5), 717-738.

15. Hagstrom, T., & Keller, H. B. (1986). The numerical calculation of traveling wave solutions of nonlinear parabolic equations. *SIAM journal on scientific and statistical computing*, *7*(3), 978-988.

16. Hariharan, G., Kannan, K., & Sharma, K. R. (2009). Haar wavelet method for solving Fisher's equation. *Applied mathematics and computation*, *211*(2), 284-292.

17. Jiwari, R., Pandit, S., & Mittal, R. C. (2012). Numerical simulation of two-dimensional sine-Gordon solitons by differential quadrature method. *Computer Physics Communications*, *183*(3), 600-616.

18. Korkmaz, A., & Dağ, İ. (2011). Shock wave simulations using sinc differential quadrature method. *Engineering Computations*.

19. Korkmaz, A., & Dağ, İ. (2012). Cubic B- spline differential quadrature methods for the advection- diffusion equation. *International Journal of Numerical Methods for Heat & Fluid Flow*.

20. Larson, D. A. (1978). Transient bounds and time-asymptotic behavior of solutions to nonlinear equations of Fisher type. *Siam Journal on applied mathematics*, *34*(1), 93-104.

21. Mittal, R. C., & Jain, R. K. (2013). Numerical solutions of nonlinear Fisher's reaction–diffusion equation with modified cubic B-spline collocation method. *Mathematical Sciences*, *7*(1), 12.

22. Mittal, R. C., & Arora, G. (2010). Efficient numerical solution of Fisher's equation by using B-spline method. *International Journal of Computer Mathematics*, *87*(13), 3039-3051.

23. Bhatia, R., & Mittal, R. C. (2018). Numerical study of Schrödinger equation using differential quadrature method. *International Journal of Applied and Computational Mathematics*, *4*(1), 36.

24. Mittal, R. C., & Dahiya, S. (2015). Numerical simulation on hyperbolic diffusion equations using modified cubic B-spline differential quadrature methods. *Computers & Mathematics with Applications*, *70*(5), 737-749.

25. Mittal, R. C., & Kumar, S. (2006). Numerical study of Fisher's equation by wavelet Galerkin method. *International Journal of Computer Mathematics*, *83*(3), 287-298.

26. Qiu, Y., & Sloan, D. (1998). Numerical solution of Fisher's equation using a moving mesh method. *Journal of Computational Physics*, *146*(2), 726-746.

27. Quan, J. R., & Chang, C. T. (1989). New insights in solving distributed system equations by the quadrature method—I. Analysis. *Computers & Chemical Engineering*, *13*(7), 779-788.\

28. Şahin, A., Dağ, İ., & Saka, B. (2008). AB- spline algorithm for the numerical solution of Fisher's equation. *Kybernetes*.

29. Sahin, A., & Ozmen, O. (2014). Usage of higher order B-splines in numerical solution of Fisher's equation. *International Journal of Nonlinear Science*, *17*(3), 241-253.

30. Shu, C. (2012). *Differential quadrature and its application in engineering*. Springer Science & Business Media.

31. Spiteri, R. J., & Ruuth, S. J. (2002). A new class of optimal high-order strong-stability-preserving time discretization methods. *SIAM Journal on Numerical Analysis*, *40*(2), 469-491.

32. Zhao, S., & Wei, G. W. (2003). Comparison of the discrete singular convolution and three other numerical schemes for solving Fisher's equation. *SIAM Journal on Scientific Computing*, *25*(1), 127-147.