○ **Mathematical Sciences**
a SpringerOpen Journal

ORIGINAL RESEARCH   **Open Access**

Numerical solutions of nonlinear Fisher's reaction–diffusion equation with modified cubic B-spline collocation method

Ramesh Chand Mittal and Rakesh Kumar Jain[*]

Abstract

In this paper, a numerical method is proposed to approximate the numeric solutions of nonlinear Fisher's reaction–diffusion equation with modified cubic B-spline collocation method. The method is based on collocation of modified cubic B-splines over finite elements, so we have continuity of the dependent variable and its first two derivatives throughout the solution range. We apply modified cubic B-splines for spatial variable and derivatives, which produce a system of first-order ordinary differential equations. We solve this system by using SSP-RK54 scheme. The proposed method needs less storage space that causes less accumulation of numerical errors. The numerical approximate solution to the nonlinear Fisher's reaction–diffusion equation has been computed without using any transformation and linearization process. Illustrative three test examples are included to establish the effectiveness and pertinence of the technique. Easy and economical implementation is the strength of this method.

Keywords: Nonlinear Fisher's reaction–diffusion equation, Modified cubic B-spline basis functions, SSP-RK54 scheme, Thomas algorithm

Introduction

We consider the nonlinear Fisher's reaction–diffusion equation

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2} + \beta\, u(1 - u), \qquad a \le x \le b, \ t > 0, \quad (1.1)$$

with the initial and boundary conditions

$$u(x, 0) = u_0(x), \qquad a \le x \le b \qquad (1.2)$$

$$u(a, t) = g_0(t), \qquad u(b, t) = g_1(t), \qquad t \in [0, t]. \quad (1.3)$$

The properties of Fisher's equation have been contrived theoretically by many authors. The analysis of travelling wave solution of Fisher's equation has been studied by many computational approaches. Travelling wave fronts have important applications in various fields of science and engineering, for example, chemistry, biology, and medicine [1]. One of the first numerical solutions was described by Gazdag and Canosa [2] with a pseudo-spectral approach. Ablowitz and Zepetella [3]

have established an explicit solution of Fisher's equation for a special wave speed. Twizell et al. [4] and Parekh and Puri [5] have demonstrated implicit and explicit finite-differences algorithms to discuss the numerical study of Fisher's equation. Tang and Weber [6] have proposed a Galerkin finite element method for solving Fisher's equation. Mickens [7] has introduced a best finite-difference scheme for Fisher's equation. Mavoungou and Cherruault [8] have depicted a numerical study of Fisher's equation by Adomian's method. Qiu and Sloan [9] have used a moving mesh method for numerical solution of Fisher's equation. Al-Khaled [10] has proposed the sinc collocation method for Fisher's equation. Zhao and Wei [11] have presented a comparison of the discrete singular convolution and three other numerical schemes for solving Fisher's equation. Wazwaz and Gorguis [12] have given the exact solutions to Fisher's equation and to a nonlinear diffusion equation of the Fisher type by employing the Adomian decomposition method. Olmos and Shizgal [13] have constructed the numerical solutions to Fisher's equation using a pseudo-spectral approach. Mittal and Kumar [14] and El-Azab [15] have contrived Fisher's equation by applying the wavelet Galerkin

* Correspondence: rkjain_2000@rediffmail.com
Department of Mathematics, I. I. T. Roorkee, Roorkee, Uttarakhand 247667, India

Springer

method. Recently, Sahin et al. [16] have applied the B-spline Galerkin approach to find numerical solution of Fisher's equation. Cattani and Kudreyko [17] have developed multiscale analysis of the Fisher equation. Mittal and Jiwari [18] have developed a numerical study of Fisher's equation by using differential quadrature method. Mittal and Arora [19] have presented efficient numerical solution of Fisher's equation by using B-spline collocation method.

In this paper, we have presented a simple numerical method that uses modified cubic B-spline for the spatial derivatives which produce a system of first-order ordinary differential equations. We solve this system by using SSP-RK54 scheme. This method needs less storage space that causes less accumulation of numerical errors. The approximate numerical solution to the nonlinear Fisher's reaction–diffusion equation has been computed without using any transformation and linearization process.

This paper is organized as follows: In the 'Description of the methods' section, description of cubic B-spline collocation method is explained. In the 'Modified cubic B-spline collocation method' section, procedure for implementation of present method is described for Equations 1.1 to 1.3. In the 'The initial vector α^0' section, procedure to obtain an initial vector which is required to start our method is explained. We have discussed three test examples to authenticate the adaptability and accuracy of the presented method in the 'Numerical experiments and discussion' section, followed by the 'Conclusions' section that briefly summarizes the numeric approximate outcomes.

Table 1 Coefficient of cubic B-splines and its derivatives at knots x_j

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

Description of the methods

In the cubic B-spline collocation method, the approximate solution can be written as a linear combination of cubic B-spline basis functions for the approximation space under consideration.

We consider a mesh $a = x_0 < x_1, \ldots, x_{N-1} < x_N = b$ as a uniform partition of the solution domain $a \le x \le b$ by the knots x_j with $h = x_j - x_{j-1} = \frac{b-a}{N}, j = 1, \ldots, N$.

Our numerical treatment for solving Equation 1.1 using the collocation method with cubic B-splines is to find an approximate solution $U^N(x,t)$ to the exact solution $u(x,t)$ in the form

$$U^N(x,t) = \sum_{j=-1}^{N+1} \alpha_j(t) B_j(x), \qquad (2.1)$$

where $\alpha_j(t)$ are unknown time-dependent quantities to be determined from the boundary conditions and collocation from the differential equation.
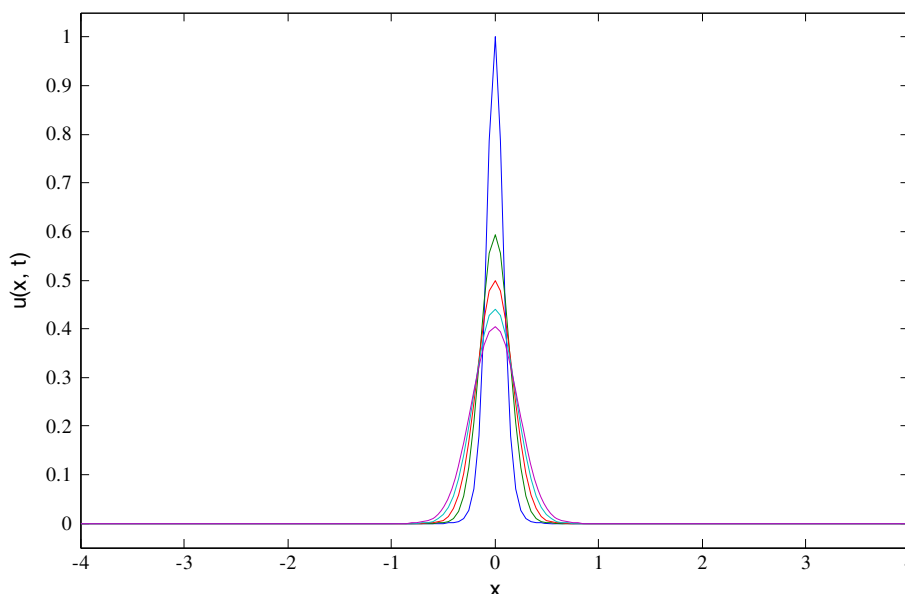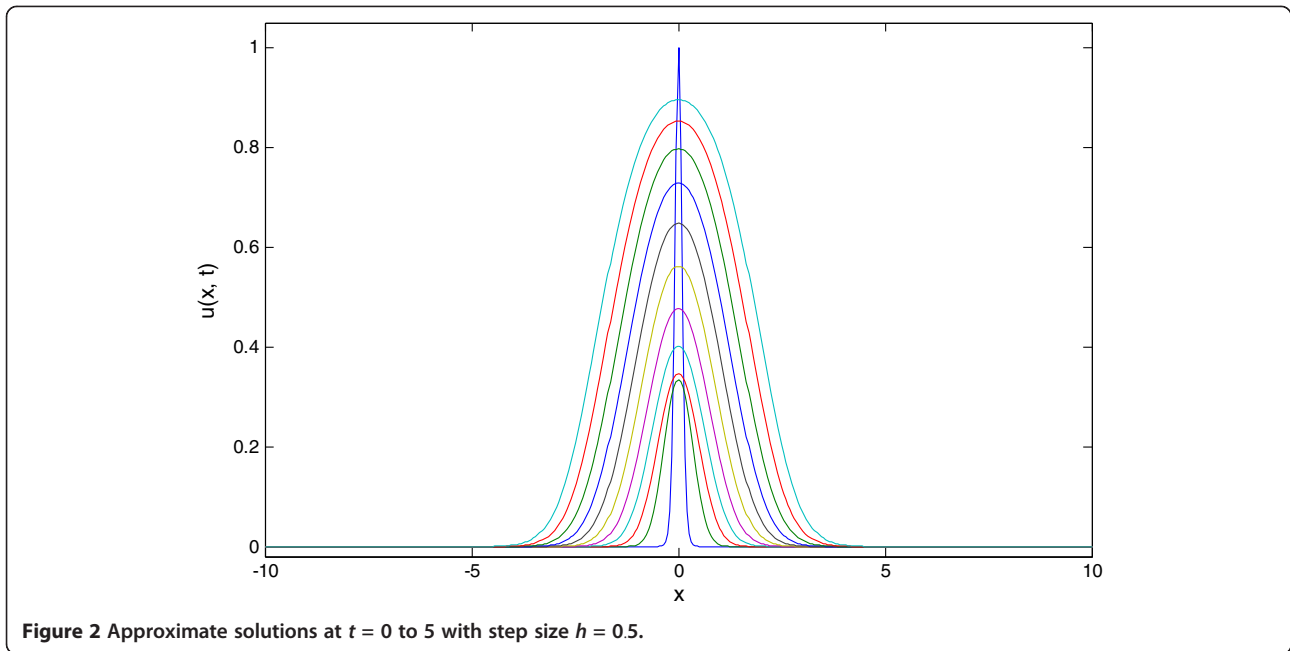

Figure 1 Approximate solutions at $t = 0$ to 0.2 with step size $h = 0.05$.
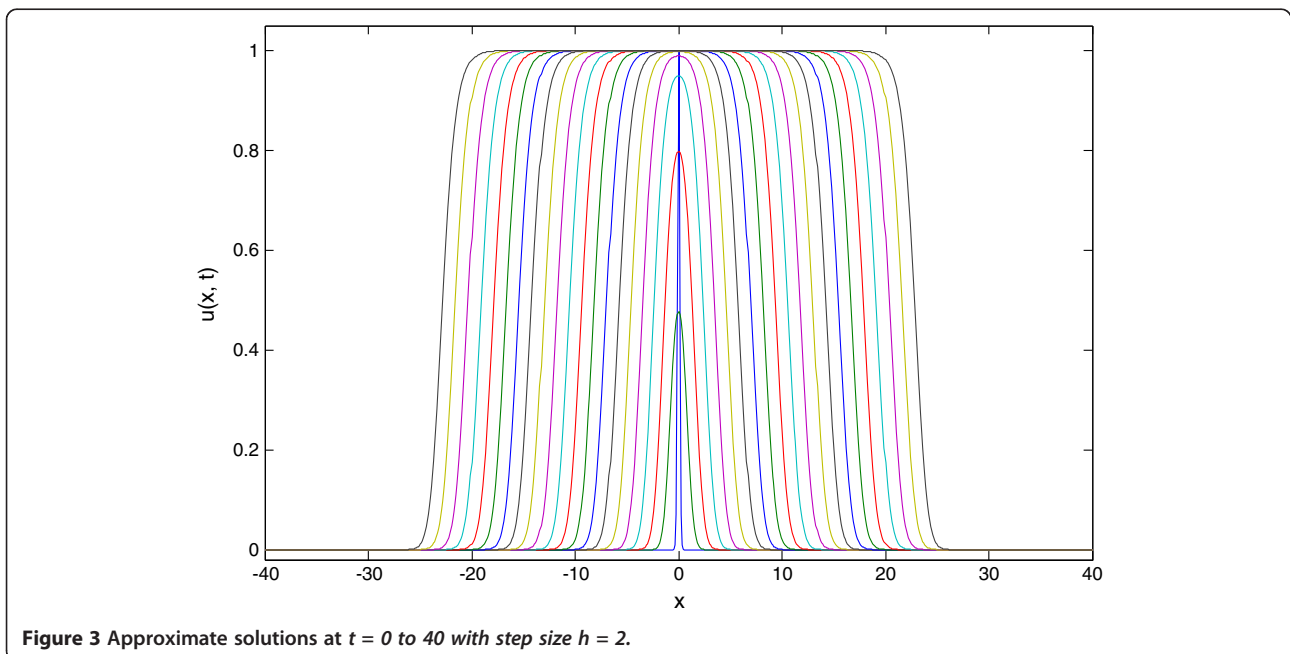
Figure 2 Approximate solutions at *t* = 0 to 5 with step size *h* = 0.5.

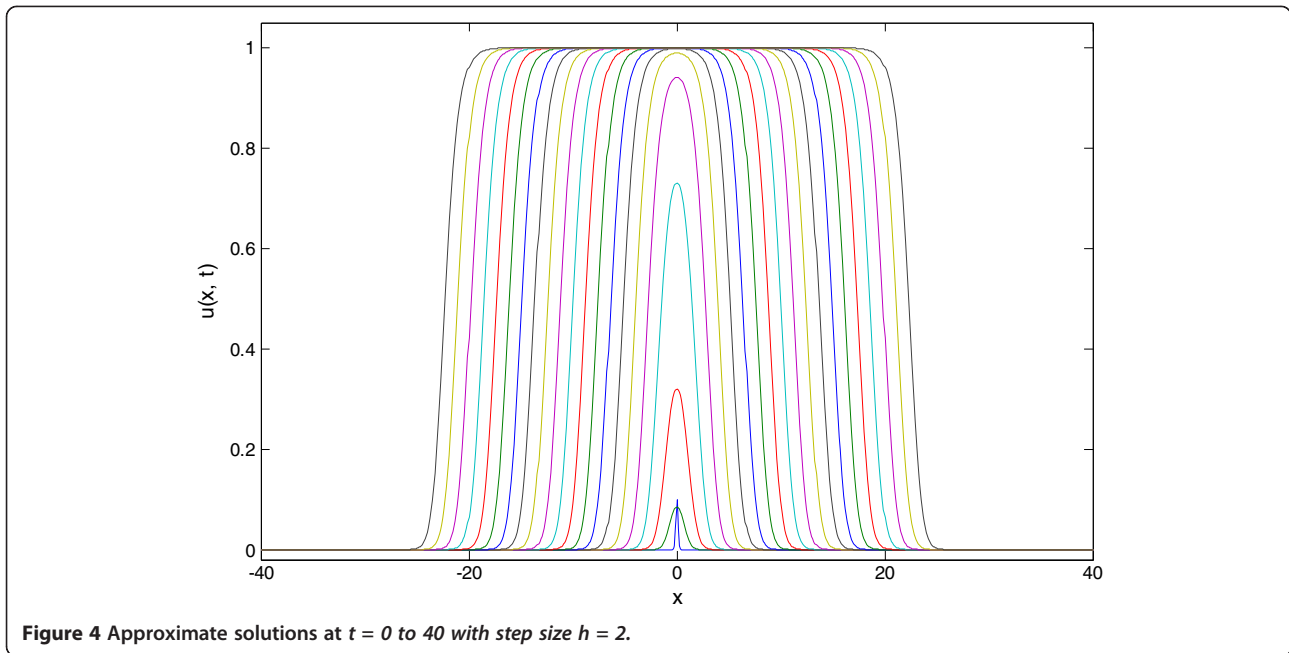The cubic B-spline $B_j(x)$ at the knots is given by

$$B_j(x) = \frac{1}{h^3}\begin{cases} (x - x_{j-2})^3 & x\varepsilon\,[x_{j-2}, x_{j-1}) \\ (x - x_{j-2})^3 - 4(x - x_{j-1})^3 & x\varepsilon\,[x_{j-1}, x_j) \\ (x_{j+2} - x)^3 - 4(x_{j+1} - x)^3 & x\varepsilon\,[x_j, x_{j+1}) \\ (x_{j+2} - x)^3 & x\varepsilon\,[x_{j+1}, x_{j+2}) \\ 0 & otherwise \end{cases}$$

(2.2)

where $B_{-1}, B_0, B_1, \ldots, B_{N-1}\, B_N, B_{N+1}$ forms a basis over the region $a \le x \le b$.

Each cubic B-spline covers four elements, so each element is covered by four cubic B-splines. The values of $B_j(x)$ and its derivative may be tabulated as in Table 1.

Using approximate function (2.1) and cubic B-spline function (2.2), the approximate values of UN(x) and its two derivatives at the knots/nodes are determined in terms of the time parameters αj as follows:
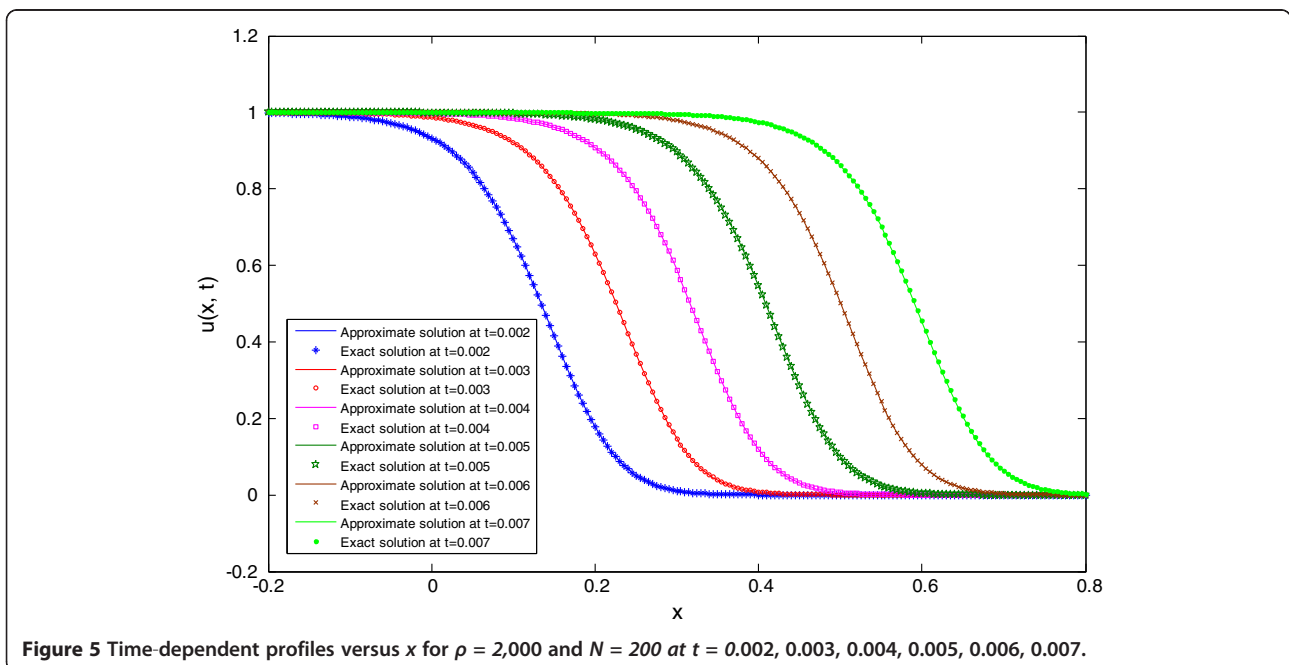

Figure 3 Approximate solutions at *t* = 0 to 40 with step size *h* = 2.

Figure 4 Approximate solutions at $t = 0$ to 40 with step size $h = 2$.

$$U_j = \alpha_{j-1} + 4\alpha_j + \alpha_{j+1}$$
$$hU_j' = 3(\alpha_{j+1} - \alpha_{j-1})$$
$$h^2 U_j'' = 6(\alpha_{j-1} - 2\alpha_j + \alpha_{j+1})$$
(2.3)

Modified cubic B-spline collocation method

In this paper, we have used the following modification in cubic B-spline basis functions to solve Fisher's equation. In the proposed method, there is a necessity to modify the cubic B-spline basis functions into a new set of basis functions in such manner, so we obtained a diagonally dominant system of differential equations for handling with Dirichlet boundary conditions. The procedure for modifying the basis functions is given as follows:

$$\tilde{B}_0(x) = B_0(x) + 2B_{-1}(x)$$

$$\tilde{B}_1(x) = B_1(x) - B_{-1}(x)$$

$$\tilde{B}_j(x) = B_j(x), j = 2, \ldots, N - 2$$
(3.1)

Figure 5 Time-dependent profiles versus x for $\rho = 2{,}000$ and $N = 200$ at $t = 0.002, 0.003, 0.004, 0.005, 0.006, 0.007$.
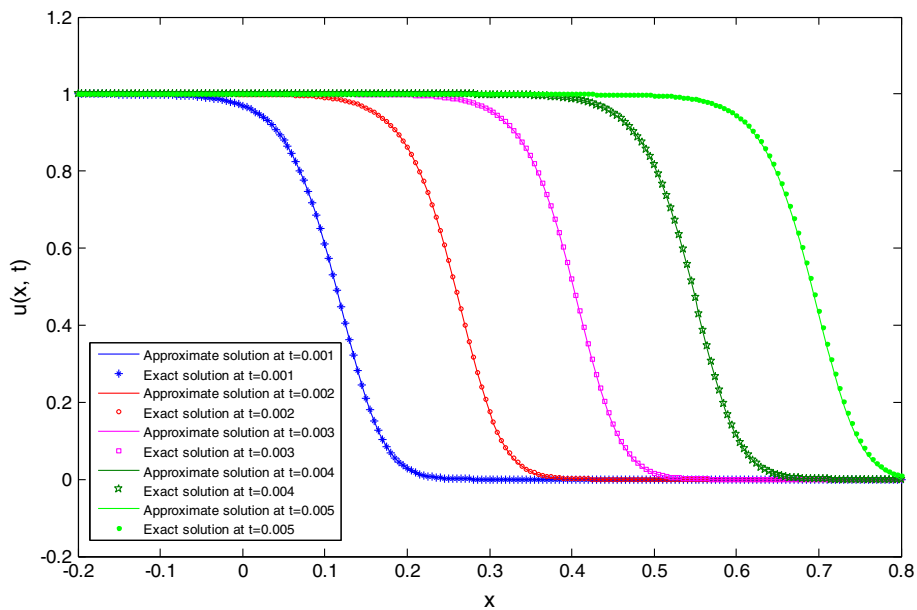
Figure 6 Time-dependent profiles versus *x* for *ρ* = 5,000 and *N* = 200 at *t* = 0.001, 0.002, 0.003, 0.004, 0.005.

$$\tilde{B}_{N-1}(x) = B_{N-1}(x) - B_{N+1}(x)$$

$$\tilde{B}_{N}(x) = B_{N}(x) + 2B_{N+1}(x)$$

Now, we consider the approximate solution using the modified cubic B-spline basis functions in the form

$$U^N(x_0, t) = g_0(t), \text{for } j = 0$$

$$U^N(x_j, t) = \sum_{j=0}^{N} \alpha_j \tilde{B}_j(x), \text{for } j = 1, \dots, N-1 \quad (3.2)$$

$$U^N(x_N, t) = g_1(t), \text{for } j = N$$

To apply the proposed method with the modified set of cubic B-spline basis functions $\tilde{B}_j(x), j = 0, \dots, N$ to Equations 1.1 to 1.3, we proceed as follows:
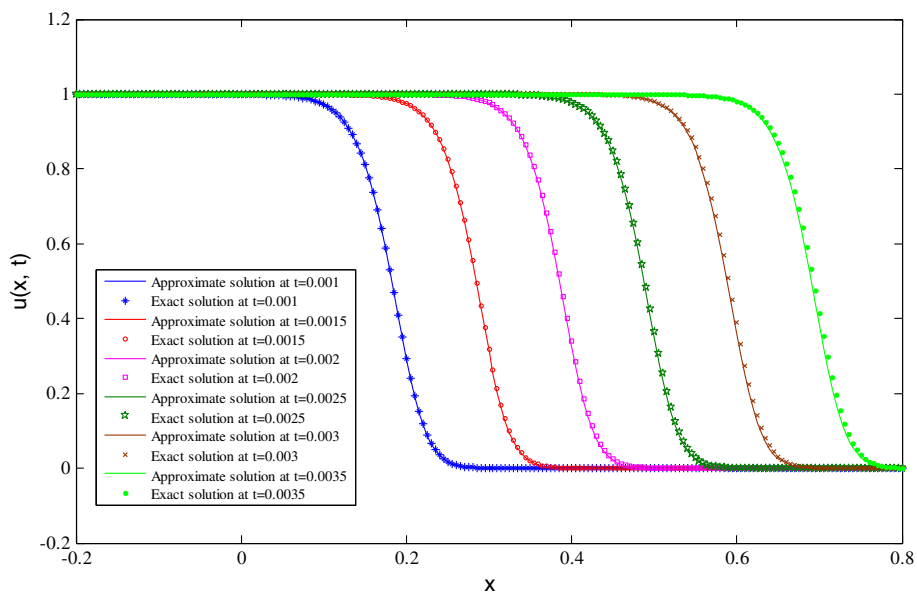

Figure 7 Time-dependent profiles versus *x* for *ρ* = 10,000 and *N* = 200 at *t* = 0.001, 0.0015, 0.002, 0.0025, 0.003, 0.0035.

Table 2 Comparison of numerical and exact solutions

x	t = 0.001			t = 0.0015			t = 0.002		
	Mittal and Arora [19]	Present	Exact solution	Mittal and Arora [19]	Present	Exact solution	Mittal and Arora [19]	Present	Exact solution
−0.20	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
−0.15	1.00000	1.00000	1.00000	1.00000	1.00001	1.00000	1.00000	1.00001	1.00000
−0.10	0.99999	0.99999	0.99999	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
−0.05	0.99994	0.99994	0.99994	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
0.05	0.99633	0.99631	0.99631	0.99994	0.99994	0.99994	1.00000	1.00000	1.00000
0.10	0.97203	0.97199	0.97209	0.99956	0.99956	0.99956	0.99999	0.99999	0.99999
0.15	0.80985	0.81066	0.81200	0.99655	0.99657	0.99661	0.99995	0.99995	0.99995
0.20	0.28644	0.29002	0.29376	0.97347	0.97386	0.97430	0.99958	0.99959	0.99959
0.25	0.01580	0.01688	0.01774	0.81756	0.82149	0.82512	0.99670	0.99679	0.99688
0.30	0.00032	0.00035	0.00038	0.29819	0.30817	0.31675	0.97454	0.97551	0.97634
0.40	0.00000	0.00000	0.00000	0.00034	0.00040	0.00045	0.30845	0.32607	0.34035
0.50	0.00000	0.00000	0.00000	0.00000	0.00001	0.00001	0.00036	0.00045	0.00053
0.55	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00001	0.00001	0.00001
0.60	0.00000	0.00000	0.00000	2.76E−11	0.00000	0.00000	1.03E−07	0.00000	0.00000
0.65	0.00000	0.00000	0.00000	4.65E−13	0.00000	0.00000	1.74E−09	0.00000	0.00000
0.70	0.00000	0.00000	0.00000	7.85E−15	0.00000	0.00000	2.93E−11	0.00000	0.00000

Our numerical treatment for solving Equation 1.1 using the collocation method with modified cubic B-splines is to find an approximate solution $U^N(x,t)$ to the exact solution $u(x,t)$ which is given in (3.2), where $\alpha_j(t)$ are time-dependent quantities to be determined from the boundary conditions and collocation from the differential equation.

Using approximate solution (3.2) and modified cubic B-spline function (3.1), the approximate values of $U_t^N(x)$ at the knots/nodes are determined in terms of the time parameters α_j as follows:

$$
\begin{aligned}
(U_t)_0 &= \dot{g}_0(t), \text{ for } j = 0 \\
(U_t)_j &= \sum_{j=0}^{N} \dot{\alpha}_j \tilde{B}_j(x), \text{ for } j = 1, \ldots, N-1 \\
(U_t)_N &= \dot{g}_1(t), \text{ for } j = N
\end{aligned}
\qquad (3.3)
$$

Table 3 Comparison of numerical and exact solutions

x	t = 0.0025			t = 0.003			t = 0.0035		
	Mittal and Arora [19]	Present	Exact solution	Mittal and Arora [19]	Present	Exact solution	Mittal and Arora [19]	Present	Exact solution
−0.20	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
−0.15	1.00000	1.00001	1.00000	1.00000	1.00002	1.00000	1.00000	1.00002	1.00000
−0.10	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
−0.05	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
0.05	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
0.10	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
0.15	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
0.20	0.99999	0.99999	0.99999	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
0.25	0.99995	0.99995	0.99995	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
0.30	0.99960	0.99961	0.99963	0.99999	0.99999	0.99999	1.00000	1.00000	1.00000
0.40	0.97545	0.97701	0.97822	0.99961	0.99964	0.99966	0.99999	0.99999	0.99999
0.50	0.31797	0.34396	0.36446	0.97625	0.97839	0.97995	0.99962	0.99966	0.99968
0.55	0.01903	0.02347	0.02728	0.83391	0.84903	0.85994	0.99703	0.99735	0.99757
0.60	0.00038	0.00051	0.00063	0.32714	0.36191	0.38895	0.97700	0.97968	0.98155
0.65	6.48E−06	0.00001	0.00001	0.02012	0.02608	0.03136	0.83851	0.85715	0.87012
0.70	1.09E−07	0.00000	0.00000	0.00040	0.00057	0.00074	0.33612	0.37997	0.41371

Table 4 Comparison of numerical and exact solutions at *t* = 2

x	Cattani and Kudreyko [17]	Mittal and Arora [19]	Present	Exact
−20	0.498681	0.498653	0.498652	0.498652
−16	0.495130	0.495745	0.495741	0.495740
−12	0.486758	0.486679	0.486670	0.486669
−8	0.459576	0.459478	0.459477	0.459478
−4	0.386681	0.386742	0.386787	0.386791
2	0.158878	0.159011	0.158859	0.158850
6	0.041822	0.041877	0.041852	0.041851
10	0.006455	0.006426	0.006462	0.006465
14	0.000750	0.000746	0.000754	0.000755
18	7.617E−05	7.79E−05	0.000079	0.000079

Using (3.1) and Table 1 in (3.3), we obtained

$$(U_t)_0 = 6\dot{\alpha}_0, \text{for } j = 0$$
$$(U_t)_j = \dot{\alpha}_{j-1} + 4\dot{\alpha}_j + \dot{\alpha}_{j+1}, \text{for } j = 1, \ldots, N-1 \quad (3.4)$$
$$(U_t)_N = 6\dot{\alpha}_N, \text{for } j = N$$

Using (3.1) to (3.4) in (1.1) to (1.3), we have

$$6\,\dot{\alpha}_0 = \dot{g}_0(t), \text{for } j = 0$$

$$\sum_0^{N+1} \dot{\alpha}_j \tilde{B}_j(x) = \alpha \left\{ \sum_0^{N+1} \alpha_j \tilde{B}_j''(x) \right\}$$
$$+ \beta \left\{ \sum_0^{N+1} \alpha_j \tilde{B}_j(x) \right\} \left\{ 1 - \sum_0^{N+1} \alpha_j \tilde{B}_j(x) \right\},$$

for $j = 1, \ldots, N-1$

$$6\,\dot{\alpha}_N = \dot{g}_1(t), \text{for } j = N \quad (3.5)$$

Using (3.2) to (3.4) in (3.5), we get a system of ordinary differential equations of the form

$$A\dot{\alpha} = \varphi \quad (3.6)$$

where $A = \begin{bmatrix} 6 & 0 & & & & \\ 1 & 4 & 1 & & & \\ & \cdots & \cdots & \cdots & & \\ & & \cdots & \cdots & \cdots & \\ & & & 1 & 4 & 1 \\ & & & & 0 & 6 \end{bmatrix}$,

$$\dot{\alpha} = \begin{bmatrix} \dot{\alpha}_0 \\ \dot{\alpha}_1 \\ \cdots \\ \cdots \\ \dot{\alpha}_{N-1} \\ \dot{\alpha}_N \end{bmatrix}, \quad \varphi = \begin{bmatrix} \varphi_0 \\ \varphi_1 \\ \cdots \\ \cdots \\ \varphi_{N-1} \\ \varphi_N \end{bmatrix}$$

$$\varphi_0 = \dot{g}_0(t), \text{for } j = 0$$

Table 5 Comparisons of numerical and exact solutions at *t* = 4

x	Cattani and Kudreyko [17]	Mittal and Arora [19]	Present	Exact
−20	0.498678	0.499412	0.499413	0.499413
−16	0.498525	0.498146	0.498142	0.498142
−12	0.494757	0.494149	0.494140	0.494140
−8	0.481776	0.481763	0.481756	0.481756
−4	0.445508	0.445372	0.445395	0.445398
2	0.279025	0.280082	0.279947	0.279941
6	0.116980	0.117196	0.116975	0.116963
10	0.025927	0.025881	0.025967	0.025974
14	0.003695	0.003559	0.003618	0.003622
18	0.000409	0.000395	0.000405	0.000406

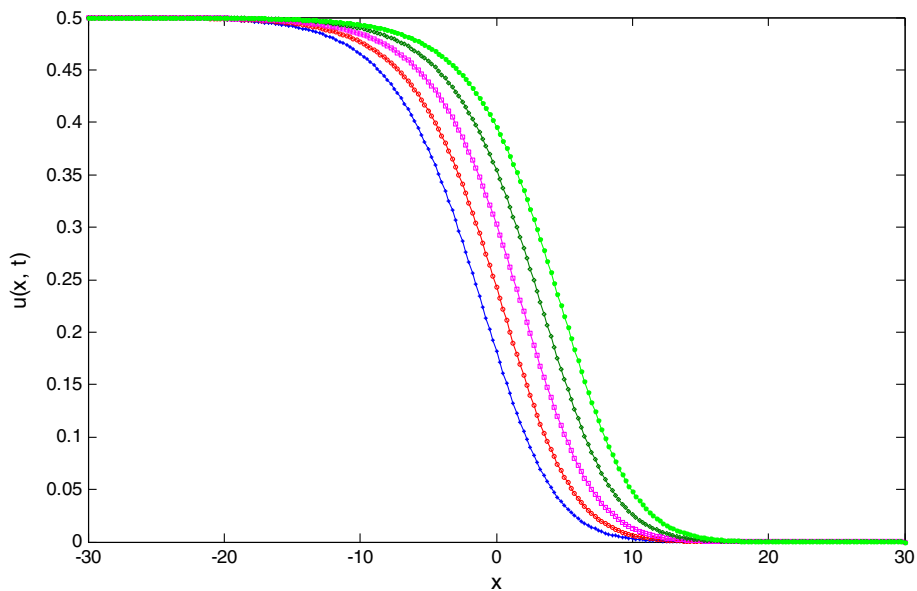**Figure 8 Approximate solutions at *t* = 1 to 5 with step 1.**

$$\varphi_j = \alpha \left(\alpha_{j-1} - 2\,\alpha_j + \alpha_{j+1} \right)$$
$$+ \beta \left(\alpha_{j-1} + 4\,\alpha_j + \alpha_{j+1} \right) \left\{ 1 - \left(\alpha_{j-1} + 4\,\alpha_j + \alpha_{j+1} \right) \right\}$$

for $j = 1, \ldots, N-1$

$$\varphi_N = \dot{g}_1(t), \text{ for } j = N$$

Here A is the $(N + 1) \times (N + 1)$ tridiagonal matrix and φ is the $(N + 1)$-order vector which depends on the boundary conditions.

Now, we solve the first-order ordinary differential equation system (3.6) by using SSP-RK54 scheme [20]. Once the parameter α^0 has been determined at a specified time level, we can compute the solution at the required knots. In (3.6), first we solve this system for vector $\dot{\alpha}$ by using a variant of the Thomas algorithm only once at each time level $t > 0$, then we get a first-order system of ordinary differential equations which can be solved for vector α by using SSP-RK54 scheme, and consequently, the solution $U^N(x,t)$ is completely known.

The initial vector α^0

The initial vector α^0 can be obtained from the initial condition and boundary values of the derivatives of the initial condition as the following expressions:

$$U(x_j, 0) = g_0(0), j = 0,$$

$$U(x_j, 0) = \varphi(x_j), j = 1, \ldots, N-1$$

$$U(x_j, 0) = g_1(0), j = N.$$

This yields a $(N + 1) \times (N + 1)$ system of equations of the form
$$A\alpha^0 = b, \quad (4.1)$$
where

$$A = \begin{bmatrix} 6 & 0 & & & & \\ 1 & 4 & 1 & & & \\ & \cdots & \cdots & \cdots & & \\ & & \cdots & \cdots & \cdots & \\ & & & 1 & 4 & 1 \\ & & & & 0 & 6 \end{bmatrix}, \quad \alpha^0 = \begin{bmatrix} \alpha_0^0 \\ \alpha_1^0 \\ \cdots \\ \cdots \\ \alpha_{N-1}^0 \\ \alpha_N^0 \end{bmatrix}, \text{ and }$$

$$b = \begin{bmatrix} g_0(0) \\ \varphi(x_1) \\ \cdots \\ \cdots \\ \varphi(x_{N-1}) \\ g_1(0) \end{bmatrix}$$

The solution of (4.1) can be found using the Thomas algorithm.

Numerical experiments and discussion

In order to show the utility and adaptability of the method, it is tested on the following three test examples.

Example 1. We consider Equation 1.1 as given in [6] with initial condition

$$u_0(x) = \text{sech}^2(10x), \quad (5.1)$$

and boundary conditions are

$$\lim_{x\to-\infty} u(x,t) = 0 \text{ and } \lim_{x\to\infty} u(x,t) = 0. \quad (5.2)$$

For all *cases*, we set $\alpha = 0.1$, $\beta = 1.0$, $h = 0.05$, and $\Delta t = 0.005$.

The space scale L is adjusted to ensure that there is sufficient space for waves to propagate.

The contour plots of u at different t are shown in Figures 1, 2, 3, and 4.

Figure 1 is for a short period of time, showing the results or $t = 0$ to 0.2 with step size $h = 0.05$. At the very beginning, near $x = 0$, $u_{xx} < 0$ with a large absolute value, but the reaction term $u(1 - u)$ is quite small, that is, the effect of diffusion dominates over the effect of reaction, so the peak goes down rapidly and gets flatter.

Figure 2 is for the period of time for $t = 0$ to 5 with step size $h = 0.5$. It shows that after the peak of the contour arrives at the lowest level, the reaction term dominates the diffusion (gradually), so it begins to go up and flatten itself until at the top $u = 1$.

Figure 3 is for the period of time $t = 0$ to 40 with step size $h = 2$. It shows that after the peak has returned to its original position, the contour on the top becomes flatter and flatter. So, after a long time, the contour looks like a bell with a flat top and very steep lateral sides, which propagate to the left and right symmetrically. The wave fronts (i.e., the lateral sides) approach a fixed shape, and their propagating speed approaches a constant value c.

Finally, in Figure 4, we show the results for the initial condition

$$u_0(x) = 0.1\text{sech}^2(10x)$$

Example 2. We consider Equation 1.1 as given in [19] with $\alpha = 1$ as

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \beta\, u(1 - u), \quad (5.3)$$

and boundary conditions are

$$\lim_{x\to-\infty} u(x,t) = 1.0 \text{ and } \lim_{x\to\infty} u(x,t) = 0. \quad (5.4)$$

The exact solution of Equation 5.3 is taken as

$$u(x,t) = \frac{1}{\left[1 + \exp\left(\left(\sqrt{\frac{\beta}{6}}\right)x - \left(\frac{5\beta}{6}\right)t\right)\right]^2} \quad (5.5)$$

The numerical solution of Equation 5.3 with given boundary conditions by taking $[a, b] = [-0.2, 0.8]$ has been computed at $\beta = 2,000$, $5,000$, and $10,000$ with $\Delta t = 0.0001$. The number of partitions is taken 40 for $\beta = 2,000$, $5,000$, and 120 for $\beta = 10,000$.

The computed results are presented graphically to compare results with other researchers. Results obtained are found in good agreement with the results obtained by Olmos and Shizgal [13] and Mittal and Arora [19].

Figures 5, 6, and 7 have depicted the exact and numerical solutions at different times.

For $\beta = 10,000$, results are also presented in a tabular form in Tables 2 and 3 to compare with the exact solutions.

Example 3. We consider Fisher's equation as given in [19]

$$\frac{\partial u}{\partial t} = \alpha\frac{\partial^2 u}{\partial x^2} - bu^2 + au, \quad (5.6)$$

where $t \in [0,t]$, $0 < t < \infty$, $-\infty < x < \infty$ with initial condition

$$u(x,0) = -\frac{1}{4}\frac{a}{b}\left[\text{sech}^2\left(-\sqrt{\frac{a}{24c}}x\right) - 2\tanh\left(-\sqrt{\frac{a}{24c}}x\right) - 2\right] \quad (5.7)$$

and boundary conditions are

$$\lim_{x\to-\infty} u(x,t) = 0.5 \text{ and } \lim_{x\to\infty} u(x,t) = 0. \quad (5.8)$$

The exact solution of the problem is given as

$$u(x,t) = -\frac{1}{4}\frac{a}{b}\left[\text{sech}^2\left(\pm\sqrt{\frac{a}{24c}}x + \frac{5a}{12}t\right)\right.$$
$$\left. -2\tanh\left(\pm\sqrt{\frac{a}{24c}}x + \frac{5a}{12}t\right) - 2\right]. \quad (5.9)$$

The solution of the equation predicts a wave front of increasing allele frequency that propagates through the population. Only original alleles are present in front of the wave, and behind the wave is an area taken by over the mutant allele. In short, this equation states that the change of the density of labeled particles at a given time depends on the infection rate $-bu^2 + au$ and the diffusion in the neighboring area.

The term au measures the infection rate, which is proportional to the product of the density of the infected and uninfected particles. The term $-bu^2$ shows how fast the infected particles are diffusing. Some authors have already found an analytical solution [12,17] of Equation 5.6 for the initial condition (5.7). This solution presents a shock-like travelling wave. The amplitude of the wave is proportional to $\frac{a}{b}$. It means that the amplitude increases as the coefficient a increases, but decreases as b increases. The support of the wave is defined by $\sqrt{\frac{24c}{a}}$.

The rate of the wave propagation is $\frac{\left(\sqrt[5]{6}\right)}{\sqrt{ac}}$

The solution of Equation 5.6 is found out by using the B-spline collocation method for $a = 0.5$, $b = c = 1.0$.

The results obtained are shown in Tables 4 and 5 and compared with the solutions obtained in [17] and [19] with the exact solution. Figure 8 shows the time-

dependent profile versus x with $h = 0.25$ and $\Delta t = 0.01$, where $[x_L, x_R] = [-30, 30]$, at $t = 1, 2, 3, 4, 5$.

Conclusions

In this paper, we have developed a collocation method for solving nonlinear Fisher's reaction–diffusion equation with Dirichlet's boundary conditions using modified cubic B-spline basis functions. In the present method, we apply modified cubic B-splines for spatial variable and derivatives, which produce a system of first-order ordinary differential equations. The resulting systems of ordinary differential equations are solved by using SSP-RK54 scheme. The numerical approximate solutions to nonlinear Fisher's reaction–diffusion equation have been computed without using any transformation and linearization process. This method is tested on three test examples, and the approximate numerical outcomes obtained are comparable with existing solutions found in the literature. Easy and economical implementation is the strength of this method. The computed results justify the advantage of this method.

References
1. Sherratt, J: On the transition from initial data traveling waves in the Fisher–KPP equation. Dyn. Stab. Syst. **13**(2), 167–174 (1998)
2. Gazdag, J, Canosa, J: Numerical solution of Fisher's equation. J. Appl. Probab. **11**, 445–457 (1974)
3. Ablowitz, M, Zepetella, A: Explicit solution of Fisher's equation for a special wave speed. Bull. Math. Biol. **41**, 835–840 (1979)
4. Twizell, EH, Wang, Y, Price, WG: Chaos free numerical solutions of reaction–diffusion equations. Proc. R. Soc. London Sci. A. **430**, 541–576 (1990)
5. Parekh, N, Puri, S: A new numerical scheme for the Fisher's equation. J. Phys. A **23**, L1085–L1091 (1990)
6. Tang, S, Weber, RO: Numerical study of Fisher's equation by a Petrov–Galerkin finite element method. J. Aus. Math. Soc. Sci. B **33**, 27–38 (1991)
7. Mickens, RE: A best finite-difference scheme for Fisher's equation. Numer. Meth. Part. Diff. Eqns. **10**, 581–585 (1994)
8. Mavoungou, T, Cherruault, Y: Numerical study of Fisher's equation by Adomian's method. Math. Comput. Model. **19**(1), 89–95 (1994)
9. Qiu, Y, Sloan, DM: Numerical solution of Fisher's equation using a moving mesh method. J. Comput. Phys. **146**, 726–746 (1998)
10. Al-Khaled, K: Numerical study of Fisher's reaction–diffusion equation by the sinc collocation method. J. Comput. Appl. Math. **137**, 245–255 (2001)
11. Zhao, S, Wei, GW: Comparison of the discrete singular convolution and three other numerical schemes for solving Fisher's equation. SIAM Comput. **25**(1), 127–147 (2003)
12. Wazwaz, AM, Gorguis, A: An analytic study of Fisher's equation by using Adomian decomposition method. Appl. Math. Comput. **154**, 609–620 (2004)
13. Olmos, D, Shizgal, BD: A pseudospectral method of solution of Fisher's equation. J. Comput. Appl. Math. **193**, 219–242 (2006)
14. Mittal, RC, Kumar, S: Numerical study of Fisher's equation by wavelet Galerkin method. Int. J. Comput. Math. **83**, 287–298 (2006)
15. El-Azab, MS: An approximation scheme for a nonlinear diffusion Fisher's equation. Appl. Math. Comput. **186**, 579–588 (2007)
16. Sahin, A, Dag, I, Saka, B: A B-spline algorithm for the numerical solution of Fisher's equation. Kybernetes **37**, 326–342 (2008)
17. Cattani, C, Kudreyko, A: Multiscale analysis of the Fisher equation. In: (ed.) ICCSA 2008 Proceedings of the International Conference on Computational Science and its Applications, Part I. Lecture Notes in Computer Science, 5072nd edn, pp. 1171–1180. Springer, Berlin (2008)
18. Mittal, RC, Jiwari, R: Numerical study of Fisher's equation by using differential quadrature method. Int. J. Information and Systems Sciences **5**(1), 143–160 (2009)
19. Mittal, RC, Arora, G: Efficient numerical solution of Fisher's equation by using B-spline method. Int. J. Comput. Math. **87**(13), 3039–3051 (2010)
20. Spiteri, RJ, Ruuth, SJ: A new class of optimal high-order strong-stability-preserving time-stepping schemes. SIAM J. Numer. Anal. **40**, 469–491 (2002)