Chapter

An Algebraic Hyperbolic Spline Quasi-Interpolation Scheme for Solving Burgers-Fisher Equations

Mohamed Jeyar, Abdellah Lamnii, Mohamed Yassir Nour, Fatima Oumellal and Ahmed Zidna

Abstract

In this work, the results on hyperbolic spline quasi-interpolation are recalled to establish the numerical scheme to obtain approximate solutions of the generalized Burgers-Fisher equation. After introducing the generalized Burgers-Fisher equation and the algebraic hyperbolic spline quasi-interpolation, the numerical scheme is presented. The stability of our scheme is well established and discussed. To verify the accuracy and reliability of the method presented in this work, we select two examples to conduct numerical experiments and compare them with the calculated results in the literature.

Keywords: Burgers-Fisher equation, Algebraic Hyperbolic Spline, Quasi-interpolation

1. Introduction

The utilization of quasi-interpolation methods has been advanced in several fields of numerical analysis. This method can yield directly to solutions and does not require the solution of any linear system. In general, quasi-interpolation methods have attracted much attention because of their potential for solving partial differential equations [1–3], curve and surface fitting, integration, differentiation, and so on. In [2], Foucher and Sablonnière developed some collocation methods based on quadratic spline quasi-interpolants for solving the elliptic boundary value problems. In [4], Bouhiri et al. have used the cubic spline collocation method to solve a two-dimensional convection-diffusion equation. Generally, the problems involving Burger's equation arise in several important applications throughout science and engineering, including fluid motion, gas dynamics, [5] transfer and number theory [6].

In literature, recent developments in the resolution of the nonlinear Burger's-Fisher equation have been achieved. In a recent study [7], Mohammadi developed a stable and accurate numerical method, based on the exponential spline and finite difference approximations, to solve the generalized Burgers'-Fisher equation. The main advantage of the last method is its simplicity. Kaya et al. [8] presented numerical simulation and explicit solutions of the generalized Burgers-Fisher. Ismail et al. [9] used the Adomian decomposition method for the solutions of Burger-Huxley and Burgers-Fisher equations. In [10] Mickens proposed a non-standard finite difference scheme for the Burgers-Fisher equation. A compact finite difference method for the generalized Burgers-Fisher equation was proposed by Sari et al. [11]. Khattak [12] presented a computational radial basis function method for the Burgers-Fisher equation and some various powerful mathematical methods such as factorization method [13], tanh function methods [6, 14], spectral collocation method [15, 16] and variational iteration method [17]. In [18], the fractional-order Burgers-Fisher and generalized Fisher's equations have been solved by using the Haar wavelet method. Recently, in Ref. [19] discontinuous Legendre wavelet Galerkin method is presented for the numerical solution of the Burgers-Fisher and generalized Burgers-Fisher equations. It consists to combines both the discontinuous Galerkin and the Legendre wavelet Galerkin methods. In [20], Zhu and Kang presented a numerical scheme to solve the hyperbolic conservation laws equation based on cubic B-spline quasi-interpolation. Nonlinear partial differential equations are encountered in a variety of domains of science. Burgers- Fisher equation is a well nonlinear equation because it combines the reaction, convection and diffusion mechanisms. The sticky tag of this equation is called Burgers-Fisher because it gathers the properties of the convective phenomenon from the Burgers equation and the diffusion transport as well as the reaction mechanism from the Fisher equation. This equation shows an exemplary model to express the interaction between the reaction mechanisms, convection effect and diffusion transport. For current applications, Burgers-Fisher equation is much known in financial mathematics, physics, applied mathematics.

In this work, we consider the generalized Burger's-Fisher equation ([9]) of the form:

$$\frac{\partial u}{\partial t} + \alpha u^{\delta} \frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial^2 x} + \beta u\left(1 - u^{\delta}\right), \qquad x \in \Omega = [0,1], \qquad t \geq 0 \tag{1}$$

with the initial condition

$$u(x,0) = \left\{\frac{1}{2} + \frac{1}{2}\tanh\left(\frac{-\alpha\delta}{2(\delta+1)}\right)x\right\}^{\frac{1}{\delta}}, \qquad x \in \Omega = [0,1] \tag{2}$$

and the boundary conditions

$$u(0,t) = \left\{\frac{1}{2} + \frac{1}{2}\tanh\left[\frac{\alpha\delta}{2(\delta+1)}\left(\frac{\alpha}{\delta+1} + \frac{\beta(\delta+1)}{\alpha}\right)t\right]\right\}^{\frac{1}{\delta}}, \qquad t \geq 0 \tag{3}$$

and

$$u(1,t) = \left\{\frac{1}{2} + \frac{1}{2}\tanh\left[\frac{-\alpha\delta}{2(\delta+1)}\left(1 - \left(\frac{\alpha}{\delta+1} + \frac{\beta(\delta+1)}{\alpha}\right)t\right)\right]\right\}^{\frac{1}{\delta}}, \qquad t \geq 0 \tag{4}$$

The exact solution of Eq. (1) (presented in [9]) is given by:

$$u(x,t) = \left\{\frac{1}{2} + \frac{1}{2}\tanh\left[\frac{-\alpha\delta}{2(\delta+1)}\left(x - \left(\frac{\alpha}{\delta+1} + \frac{\beta(\delta+1)}{\alpha}\right)t\right)\right]\right\}^{\frac{1}{\delta}}. \tag{5}$$

Our main purpose in this chapter is to use the univariate quasi-interpolant associated with the algebraic hyperbolic B-spline of order 4 for solving the Burgers-Fisher Eqs. (10). Firstly, we approximate first and second-order partial derivatives by those of the algebraic hyperbolic spline $\mathcal{Q}_4 u(x_i, t_n)$ quasi-interpolant. Then, we use this derivatives to approximate $\left(\frac{\partial u}{\partial x}\right)_i^n$ and $\left(\frac{\partial^2 u}{\partial^2 x}\right)_i^n$. The resulting system can be solved using

MATLAB's ode solver. More precisely, we provide a powerful numerical scheme applying a hyperbolic quasi interplant used in [21] to solve Burger's Fisher equation. This method produces better results compared to the results obtained by all the schemes in the literature, for example, those studied in [22, 23].

The chapter is organized as follows. Section 2 is dedicated to the description of the quasi-interpolation of the algebraic hyperbolic splines. Afterward, Section 3 is devoted to the presentation of numerical techniques to solve the Burger's-Fisher equation. The stability of the scheme has been studied in Section 4. In Section 5, two examples of the Burger's-Fisher equation are illustrated and compared to those obtained with some previous results. Finally, our conclusion is presented in Section 6.

2. Algebraic hyperbolic spline quasi-interpolation of order 4

In this section, we recall the results on hyperbolic spline quasi-interpolation that we will use to establish the numerical method (see [21]). Let $T = \{x_i = ih\}_{i=-\infty}^{+\infty}$ $(0 < h < \pi)$ be a set of knots which partition the parameter axis x uniformly.

For $k \geq 3$, the B-spline family that generates the space $\Gamma_k = \{\sinh(x), \cosh(x), 1, x, \cdots, x^{k-3}\}$ is called algebraic hyperbolic B-spline (for more details see [24]), which can be defined as for $k = 2$:

$$N_{0,2}(x) = \begin{cases} \dfrac{h \sinh(x)}{2(\cosh(h) - 1)}, & 0 \leq x < h, \\ \dfrac{h \sinh(2h - x)}{2(\cosh(h) - 1)}, & h \leq x < 2h, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

$$N_{i,2}(x) = N_{0,2}(h, x - ih), \quad (i = 0, \pm 1, \pm 2, \cdots) \tag{7}$$

and for $k \geq 3$,

$$N_{i,k}(x) = \frac{1}{h} \int_{x-h}^{x} N_{i,k-1}(h, s) ds, \quad (i = 0, \pm 1, \pm 2, \cdots). \tag{8}$$

We apply the recursion formula (8) to get the algebraic hyperbolic B-spline of order 4, which is defined in Γ_4 as follows:

$$N_{i,4}(x) = \begin{cases} \dfrac{x - x_i + \sinh(x - x_i)}{2h(1 - \cosh(h))}, & x_i \leq x < x_{i+1}, \\ \dfrac{x - x_{i+2} - 2h\cosh(h) + 2(x - x_i)\cosh(h) + 2\sinh(x_{i+1} - x) + \sinh(x_{i+2} - x)}{2h(\cosh(h) - 1)}, & x_{i+1} \leq x < x_{i+2}, \\ \dfrac{x_{i+2} - x + 6h\cosh(h) - 2(x - x_i)\cosh(h) - \sinh(x_{i+2} - x) - 2\sinh(x_{i+3} - x)}{2h(\cosh(h) - 1)}, & x_{i+2} \leq x < x_{i+3}, \\ \dfrac{x - x_{i+4} + \sinh(x_{i+4} - x)}{2h(cosh(h) - 1)}, & x_{i+3} \leq x < x_{i+4}, \\ 0, & \text{otherwise}. \end{cases} \tag{9}$$

According to [21], the univariate Quasi-Interpolant associated to the algebraic hyperbolic B-spline of order 4, can be expressed as operators of the form

$$\mathcal{Q}_4^1 f(x) = \sum_{i=-3}^{n-1} (\overline{\nu}_h^1 f_{i+2} + \nu_h^1 (f_{i+1} + f_{i+3})) N_{i,4}(x) \tag{10}$$

where $\nu_h^1 = \frac{1}{4} \operatorname{csch}\left(\frac{h}{2}\right)^2 (h \operatorname{csch}(h) - 1), \overline{\nu}_h^1 = 1 - 2\nu_h^1$ and $f_i = f(x_i)$.

The error associated with the quadrature formula based on $Q_4^1 f$ is of order 5 as the following theorem describes.

Theorem 1 There exists a constant C_2 such that for all $f \in L_1^4([a,b])$ and for all partitions τ_h of $[a,b]$,

$$\| f - Q_4^1 f \|_\infty \le C_2 h^5 \|\mathbf{L}_4 f\|_\infty, \tag{11}$$

with L_4 is an operator defined by: $\mathbf{L}_4 := \mathbf{D}^2(\mathbf{D}^2 - 1)$ and $\mathbf{L}_4 f = 0$ for all $f \in \Gamma_4$.

Proof The proof is almost the same as that of Theorem 15 in [21].

3. Numerical scheme using hyperbolic spline quasi-interpolation

For approximate derivatives of f by derivatives of $Q_4^1 f$ up to the order h^4, we can evaluate the value of f at x_i by

$$(Q_4^1 f)' = \sum_{j=-3}^{n-1} \left(\overline{\upsilon}_h^1 f_{j+2} + \upsilon_h^1 \left(f_{j+1} + f_{j+3} \right) \right) N'_{j,4} \tag{12}$$

and

$$(Q_4^1 f)'' = \sum_{j=-3}^{n-1} \left(\overline{\upsilon}_h^1 f_{j+2} + \upsilon_h^1 \left(f_{j+1} + f_{j+3} \right) \right) N''_{j,4}. \tag{13}$$

The values of $N'_{j,4}$ and $N''_{j,4}$ using the formula (9) are

$$N'_{i,4}(x) = \begin{cases} \dfrac{1 - \cosh(x - x_i)}{2h(1 - \cosh(h))}, & x_i \le x < x_{i+1}, \\ \dfrac{1 + 2\cosh(h) - 2\cosh(x_{i+1} - x) - \cosh(x_{i+2} - x)}{2h(\cosh(h) - 1)}, & x_{i+1} \le x < x_{i+2}, \\ \dfrac{2\cosh(x_{i+3} - x) + \cosh(x_{i+2} - x) - 2\cosh(h) - 1}{2h(\cosh(h) - 1)}, & x_{i+2} \le x < x_{i+3}, \\ \dfrac{1 - \cosh(x_{i+4} - x)}{2h(\cosh(h) - 1)}, & x_{i+3} \le x < x_{i+4}, \\ 0, & \text{otherwise}, \end{cases} \tag{14}$$

and

$$N''_{i,4}(x) = \begin{cases} \dfrac{\sinh(x - x_i)}{2h(1 - \cosh(h))}, & x_i \le x < x_{i+1}, \\ \dfrac{2\sinh(x_{i+1} - x) + \sinh(x_{i+2} - x)}{2h(\cosh(h) - 1)}, & x_{i+1} \le x < x_{i+2}, \\ \dfrac{-\sinh(x_{i+2} - x) - 2\sinh(x_{i+3} - x)}{2h(\cosh(h) - 1)}, & x_{i+2} \le x < x_{i+3}, \\ \dfrac{\sinh(x_{i+4} - x)}{2h(\cosh(h) - 1)}, & x_{i+3} \le x < x_{i+4}, \\ 0, & \text{otherwise}, \end{cases} \tag{15}$$

By using Eq. (10), the first derivative of algebraic hyperbolic spline quasi-interpolation at x_i for all $i \in \{2, \cdots, n-2\}$ is

$$\begin{aligned} Q_4^1 f'(x_i) &= \sum_{j=-3}^{n-1} \left(\overline{\nu}_h^1 f_{j+2} + \nu_h^1 \left(f_{j+1} + f_{j+3} \right) \right) N'_{j,4}(x_i) \\ &= \left(\overline{\nu}_h^1 f_i + \nu_h^1 (f_{i-1} + f_{i+1}) \right) N_{i-2,4}{}'(x_i) + \left(\overline{\nu}_h^1 f_{i+1} + \nu_h^1 (f_i + f_{i+2}) \right) N'_{i-1,4}(x_i) \\ &= -\nu_h^1 f_{j-2} - \overline{\nu}_h^1 f_{i-1} + \overline{\nu}_h^1 f_{i+1} + \nu_h^1 f_{i+2} \end{aligned} \tag{16}$$

That is to say

$$Q_4^1 f'(x_i) = -\nu_h^1 f_{j-2} - \overline{\nu}_h^1 f_{i-1} + \overline{\nu}_h^1 f_{i+1} + \nu_h^1 f_{i+2} \tag{17}$$

and the second derivative of algebraic hyperbolic spline quasi-interpolation at x_i for all $i \in \{2, \cdots, n-2\}$ is

$$\begin{aligned} Q_4^1 f''(x_i) &= \sum_{j=-3}^{n-1} \left(\overline{\nu}_h^1 f_{j+2} + \nu_h^1 \left(f_{j+1} + f_{j+3} \right) \right) N''_{j,4}(x_i) \\ &= \left(\overline{\nu}_h^1 f_i + \nu_h^1 (f_{i-1} + f_{i+1}) \right) N''_{i-2,4}(x_i) + \left(\overline{\nu}_h^1 f_{i+1} + \nu_h^1 (f_i + f_{i+2}) \right) N''_{i-1,4}(x_i) \end{aligned} \tag{18}$$

That is to say

$$Q_4^1 f''(x_i) = a_h \left(\nu_h^1 f_{i-2} + (-2\nu_h^1 + \overline{\nu}_h^1) f_{i-1} + 2(\nu_h^1 - \overline{\nu}_h^1) f_i + (-2\nu_h^1 + \overline{\nu}_h^1) f_{i+1} + \nu_h^1 f_{i+2} \right) \tag{19}$$

with $a_h = \frac{sinh\,(h)}{2h(cosh\,(h) - 1)}$.

Discretizing (1) in time we get

$$u_i^{n+1} - u_i^n + \alpha \left(u^\delta\right)_i^n \tau \left(\frac{\partial u}{\partial x}\right)_i^n = \tau \left(\frac{\partial^2 u}{\partial^2 x}\right)_i^n + \beta \tau u_i^n \left(1 - \left(u^\delta\right)_i^n\right) \tag{20}$$

where u_i^n is the approximation of the value $u(x,t)$ at (x_i, t_n), $t_n = n\tau$ and τ is the time step with $0 \le i \le M$ and $0 \le n \le N$. Then, we use the derivatives of the algebraic hyperbolic spline $Q_4 u(x_i, t_n)$ quasi-interpolant to approximate $\left(\frac{\partial u}{\partial x}\right)_i^n$ and $\left(\frac{\partial^2 u}{\partial^2 x}\right)_i^n$.

Assume that $U^n = \left(u_0^n, u_1^n, \cdots, u_M^n\right)$ is known for the non-negative integer n. We set unknown vectors as

$$\begin{cases} \left(\frac{\partial U}{\partial x}\right)^n = \left(\left(\frac{\partial u}{\partial x}\right)_0^n, \left(\frac{\partial u}{\partial x}\right)_1^n, \cdots, \left(\frac{\partial u}{\partial x}\right)_M^n \right) \\ \left(\frac{\partial^2 U}{\partial^2 x}\right)^n = \left(\left(\frac{\partial^2 u}{\partial^2 x}\right)_0^n, \left(\frac{\partial^2 u}{\partial^2 x}\right)_1^n, \cdots, \left(\frac{\partial^2 u}{\partial^2 x}\right)_M^n \right) \end{cases} \tag{21}$$

From the initial conditions ((3), (4)) and boundary conditions (2), we can compute the numerical solution of (1) step by step using the scheme (20) and formulas ((17), (19)).

According to (17), (19) and (21). the scheme (20) can be rewritten as

$$u_i^{n+1} = u_i^n + \alpha(u^\delta)_i^n \frac{\tau}{2h}\left(-\nu_h^1 u_{i-2}^n - \overline{\nu}_h^1 u_{i-1}^n + \overline{\nu}_h^1 u_{i+1}^n + \nu_h^1 u_{i+2}^n\right) + \frac{\tau}{2h} a_h \left(\nu_h^1 u_{i-2}^n + \left(-2\nu_h^1 + \overline{\nu}_h^1\right) u_{i-1}^n \right. \\ \left. + 2\left(\nu_h^1 - \overline{\nu}_h^1\right) u_i^n + \left(-2\nu_h^1 + \overline{\nu}_h^1\right) u_{i+1}^n + \nu_h^1 u_{i+2}^n\right) + \beta\tau u_i^n \left(1 - \left(u^\delta\right)_i^n\right), \text{for all } i \in \{2, \cdots, M-2\} \tag{22}$$

with $a_h = \frac{sinh\,(h)}{2h(\,cosh\,(h) - 1)}$.

This scheme is called the algebraic hyperbolic quasi-interpolation (AHQI) scheme.

4. Stability analysis

Sharma and Singh provided a method to study the stability of the nonlinear partial equation in [25], which we used in this section to study the stability of our scheme.

If we set $r = \frac{\tau}{2h}$, $\mathcal{A}_i^n = \alpha(u^\delta)_i^n$, $\mathcal{B}_i^n = \beta(u^\delta)_i^n$, then the scheme (22) becomes

$$u_i^{n+1} = \left(-\mathcal{A}_i^n r\nu_h^1 + r a_h \nu_h^1\right) u_{i-2}^n + \left(-\mathcal{A}_i^n r\overline{\nu}_h^1 + r a_h\left(\overline{\nu}_h^1 - 2\nu_h^1\right)\right) u_{i-1}^n + \left(1 + \beta\tau - \tau\mathcal{B}_i^n \right. \\ \left. + 2 r a_h\left(\nu_h^1 - \overline{\nu}_h^1\right)\right) u_i^n + \left(\mathcal{A}_i^n r\overline{\nu}_h^1 + r a_h\left(\overline{\nu}_h^1 - 2\nu_h^1\right)\right) u_{i+1}^n + \left(\mathcal{A}_i^n r\nu_h^1 + r a_h \nu_h^1\right) u_{i+2}^n. \tag{23}$$

If we move to the L-infinity norm then we obtain

$$\left\|U^{n+1}\right\|_L^\infty \le \sup_i \left| -\mathcal{A}_i^n r\nu_h^1 + r a_h \nu_h^1\right| \left\|U^n\right\|_L^\infty + \sup_i \left| -\mathcal{A}_i^n r\overline{\nu}_h^1 + r a_h\left(\overline{\nu}_h^1 - 2\nu_h^1\right)\right| \left\|U^n\right\|_L^\infty \\ + \sup_i \left|1 + \beta\tau - \tau\mathcal{B}_i^n + 2 r a_h\left(\nu_h^1 - \overline{\nu}_h^1\right)\right| \left\|U^n\right\|_L^\infty \\ + \sup_i \left|\mathcal{A}_i^n r\overline{\nu}_h^1 + r a_h\left(\overline{\nu}_h^1 - 2\nu_h^1\right)\right| \left\|U^n\right\|_L^\infty + \sup_i \left|\mathcal{A}_i^n r\nu_h^1 + r a_h \nu_h^1\right| \left\|U^n\right\|_L^\infty. \tag{24}$$

If we set $\mathcal{M}_1^n = Sup_i\,|a_h - \mathcal{A}_i^n|$, $\mathcal{M}_2^n = Sup_i\,|a_h + \mathcal{A}_i^n|$, $\mathcal{M}_3^n = Sup_i\,|1 + \beta\tau - \tau\mathcal{B}_i^n|$, then the Eq. (24) becomes

$$\left\|U^{n+1}\right\|_L^\infty \le \left(\mathcal{M}_3^n + |r|\left(|\overline{\nu}_h^1|\left(6|a_h| + \left(\mathcal{M}_1^n + \mathcal{M}_2^n\right)\right) + |\nu_h^1|\left(2|a_h| + \left(\mathcal{M}_1^n + \mathcal{M}_2^n\right)\right)\right)\right) \left\|U^n\right\|_L^\infty. \tag{25}$$

It implies that the scheme is stable if

$$\mathcal{M}_3^n + |r|\left(|\overline{\nu}_h^1|\left(6|a_h| + \left(\mathcal{M}_1^n + \mathcal{M}_2^n\right)\right) + |\nu_h^1|\left(2|a_h| + \left(\mathcal{M}_1^n + \mathcal{M}_2^n\right)\right)\right) \le \mathcal{C}, \tag{26}$$

with $\mathcal{C}$ is a finite positive constant.

5. Numerical results

In this section, the proposed quasi-interpolation splines collocation methods are tested for their validity for solving the generalized Burgers-Fisher equation with the initial condition (2) and the boundary conditions (3). Two different examples for the Burgers-Fisher equation are solved and the obtained results are compared with those presented in [22, 25]. To verify the accuracy and reliability of the present method in this article, we select two examples to conduct numerical experiments and compare them with the calculated results in the existing literature. That's why we divided this section into two subsections, in each subsection we compared our scheme (AHQI scheme) to each example by comparing their maximum error which is defined by

$$e = \underset{1 \leq i \leq M}{Max} |u_i^{exact} - u_i^{approach}|. \tag{27}$$

5.1 First example: MCN scheme

In the first example, we compared the maximum error of AHQI scheme with MCN scheme proposed in [22]. In **Table 1** we showed the maximum error of each scheme with different values of N with $\alpha = \beta = \delta = 1$ and we remarked that our method is better than that presented in [22], also we illustrated the

x	N = 10		N = 100		N = 1000	
	AHQI	MCN	AHQI	MCN	AHQI	MCN
	1×10^{-5}	1×10^{-4}	1×10^{-6}	1×10^{-5}	1×10^{-9}	1×10^{-8}
0.1	0.0442	0.0987	0.0248	0.0865	0.0128	0.2880
0.2	0.0923	0.1269	0.0760	0.1153	0.0384	0.2834
0.3	0.1399	0.1352	0.1225	0.1232	0.0727	0.2060
0.4	0.1869	0.1376	0.1662	0.1250	0.1293	0.1419
0.5	0.2329	0.1383	0.2064	0.1253	0.2180	0.1158
0.6	0.2778	0.1379	0.2394	0.1251	0.3439	0.1315
0.7	0.3213	0.1359	0.2595	0.1235	0.4929	0.1836
0.8	0.3633	0.1287	0.2458	0.1162	0.5740	0.2489
0.9	0.4037	0.2489	0.1225	0.0882	0.3241	0.2516

Table 1.
Values of errors by AHQI and MCN.

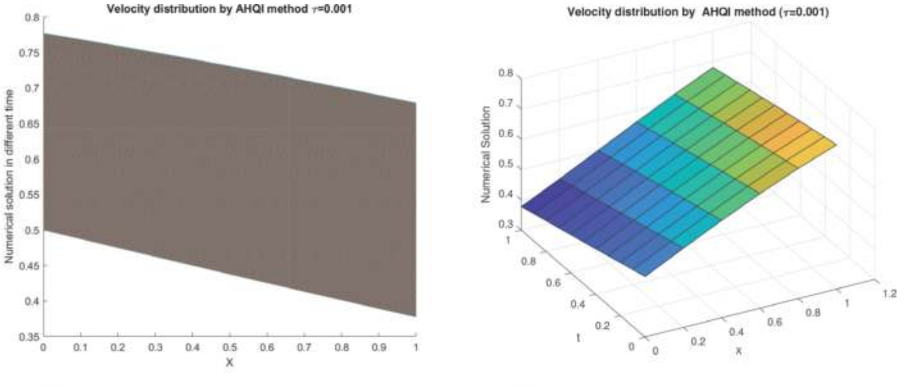

Figure 1.
The behavior of numerical results of equation (1) by AHQI for $\tau = 0.001$.

numerical results of Eq. (1) by our method in **Figure 1(a)** and **(b)** for different values in space and time.

5.2 Second example: BSQI scheme

For the second example, we compared our scheme to BSQI scheme proposed in [25] for different values of α, β and τ: in the **Table 2** $\alpha = \beta = 0.001, \tau = 0.0001$ and in the **Table 3** $\alpha = \beta = 1, \tau = 0.0001$ with $M = 16$. In each table we calculate the maximum error for different values of δ and we remarked that for $\delta = 1$ and $\delta = 2$ our method is better than the other scheme but is close to it for $\delta = 3$. We also

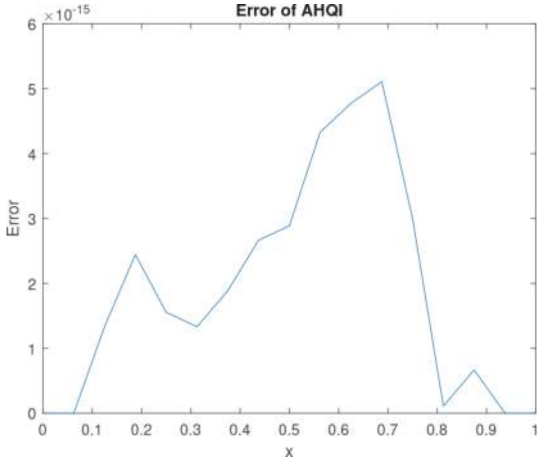

Figure 2.
The absolute errors for $\alpha = \beta = 0.001$, $\tau = 0.0001$, $\delta = 1$, $t = 1$.

t	$\delta = 1$		$\delta = 2$		$\delta = 4$	
	AHQI	BSQI	AHQI	BSQI	AHQI	BSQI
0.001	2.22044×10^{-16}	4.44089×10^{-16}	2.88657×10^{-15}	9.76996×10^{-15}	2.37587×10^{-14}	3.37508×10^{-14}
0.005	3.88578×10^{-16}	1.55431×10^{-15}	1.26565×10^{-14}	4.17888×10^{-13}	1.16573×10^{-13}	1.66422×10^{-13}
0.010	6.66133×10^{-16}	1.83187×10^{-15}	2.45359×10^{-14}	9.18154×10^{-14}	2.32702×10^{-13}	3.25628×10^{-13}
0.500	1.19904×10^{-14}	3.81917×10^{-14}	2.12041×10^{-13}	1.07303×10^{-12}	2.12041×10^{-12}	3.88234×10^{-12}
1.000	5.10702×10^{-15}	3.04201×10^{-14}	2.17048×10^{-13}	1.08047×10^{-12}	2.12352×10^{-12}	3.91087×10^{-12}

Table 2.
Error for various values of δ *and* x *with* $\alpha = \beta = 0.001$, $\tau = 0.0001$.

t	$\delta = 1$		$\delta = 2$		$\delta = 4$	
	AHQI	BSQI	AHQI	BSQI	AHQI	BSQI
0.2	3.99146×10^{-8}	5.55746×10^{-7}	1.26606×10^{-7}	2.56108×10^{-6}	1.76174×10^{-7}	1.76161×10^{-6}
0.4	7.94950×10^{-8}	9.05507×10^{-7}	1.74941×10^{-7}	4.24308×10^{-6}	1.82135×10^{-6}	4.17351×10^{-7}
0.6	1.76244×10^{-7}	2.18808×10^{-6}	2.71508×10^{-7}	3.56848×10^{-6}	1.32793×10^{-6}	2.42401×10^{-6}
0.8	2.34913×10^{-7}	2.93314×10^{-6}	3.79941×10^{-7}	1.46518×10^{-6}	7.62594×10^{-7}	2.35757×10^{-6}
1	2.47511×10^{-7}	3.01455×10^{-6}	3.98493×10^{-7}	5.54230×10^{-6}	3.79941×10^{-7}	1.44350×10^{-6}

Table 3.
Error for various values of δ *and* x *with* $\alpha = \beta = 1$, $\tau = 0.0001$.

illustrated the maximum error for $\alpha = \beta = 0.001, \delta = 1, \tau = 0.0001$ and $\alpha = \beta = 1, \delta = 1, \tau = 0.00001$ in $t = 1$ and in different values of space as the **Figures 1** and **2** respectively show. The results of our method for three different space size steps ($\delta = 1, 2, 4$) and five different time size steps t are shown in **Tables 2** and **3**. It is very clear that a good agreement between the analytical solution and the present numerical results with a minimum error is obtained, and the error becomes clear when using a large size step for time and space.

6. Conclusion

In this work, a numerical scheme to solve the nonlinear Burgers -Fisher equation has been proposed using algebraic hyperbolic spline quasi-interpolation. The numerical scheme stability was well established. The scheme efficiency, as well as its accuracy, are justified by treating well-known examples in the literature, for each case the error is reported. We conclude that the scheme with algebraic hyperbolic spline quasi-interpolation can solve Burgers-Fisher equations since it produces reasonably good results, with high convergence with very small errors.

Author details

Mohamed Jeyar[1#], Abdellah Lamnii[2*#], Mohamed Yassir Nour[2,3#], Fatima Oumellal[2#] and Ahmed Zidna[3]

1 Faculty of Sciences, First Mohammed University, Oujda, Morocco

2 Hassan First University of Settat, Faculté des Sciences et Technique, LaboratoryMISI, Morocco

3 LGIPM, Université Lorraine, Metz, France

*Address all correspondence to: a_lamnii@yahoo.fr

# These authors contributed equally.

References

[1] R. Chen, Quasi-interpolation with radial basis function and application to solve partial differential equations, Ph. D Thesis, Fudan University, (2005).

[2] F. Foucher, P. Sablonnière, Quadratic spline quasi-interpolants and collocation methods, Math. Comp. Simul, **79** (2009, 3455-3465.

[3] C.G. Zhu, R.H. Wang, Numerical solution of burgers' equation by cubic b-spline quasiinterpolation, Appl. Math. Comput, **208** (2009), 260-272.

[4] S. Bouhiri, A. Lamnii, M. Lamnii Cubic quasi-interpolation spline collocation method for solving convection-diffusion equations, Mathematics and Computers in Simulation **164** (2019), 33-45.

[5] E. J. Parkes, B.R. Duffy, An automated tanh-function method for finding solitary wave solutions to non-linear evolution equations, Comput. Phys. Comm, **98** (1996), 288-300.

[6] A. M. Wazwaz, The tanh method for generalized forms of nonlinear heat conduction and Burgers-Fisher equations, Appl. Math. Comput, **169** (2005), 321-338.

[7] R. Mohammadi, Spline solution of the generalized Burgers'-Fisher equation, Applicable Analysis, **91** (2012), 2189-2215.

[8] D. Kaya, S. M. El-Sayed, A numerical simulation and explicit solutions of the generalized Burger-Fisher equation, Appl Math Comput, **152** (2004), 403-413.

[9] H.N.A. Ismail, K. Raslan, A.A.A. Rabboh: Adomian decomposition method for Burgers' Huxley and Burgers-Fisher equations, Appl, Math. Comput. **159** (2004), 291-301.

[10] R.E. Mickens, A.B. Gumel, Construction and analysis of a non-standard finite difference scheme for the Burgers-Fisher equation, J. Sound Vib, **257** (2002),791-797.

[11] M. Sari, G. Gürarslan, I. Dağ, A compact finite difference method for the solution of the generalized Burgers-Fisher equation, Numer.Methods Partial Differential Equations, **26** (2010), 125-134.

[12] A.J. Khattak, A computational meshless method for the generalized Burger's-Huxley equation, Appl. Math. Modelling, **33**(2009), 3718-3729.

[13] H. Fahmy, Travelling wave solutions for some time-delayed equations through factorizations, Chaos Soliton Fract, **38** (2008), 1209-1216.

[14] S. A. El-Wakil, M. A. Abdou, Modified extended tanh-function method for solving nonlinear partial differential equation, Chaos Soliton Fract, **31** (2007), 1256-1264.

[15] A. Golbabai, M. Javidi, A spectral domain decomposition approach for the generalized Burger's-Fisher equation, Chaos Soliton Fract, **39** (2009), 385-392.

[16] M. Javidi, Spectral collocation method for the solution of the generalized Burger-Fisher equation, Appl Math Comput, **174**(2006), 345-352.

[17] M. Moghimi, F. S. A. Hejazi, Variational iteration method for solving generalized Burger-Fisher and Burger equations, Chaos Soliton Fract, **33** (2007), 1756-1761.

[18] A. K. Gupta and S. Saha Ray, On the Solutions of Fractional Burgers-Fisher and Generalized Fisher's Equations Using Two Reliable Methods, International Journal of Mathematics

and Mathematical Sciences
Volume 2014, Article ID 682910.

[19] S. Kumar, S. Saha Ray,Numerical treatment for Burgers-Fisher and generalized Burgers-Fisher equations, Mathematical Sciences, **15** (2021), 21-28.

[20] C.G. Zhu, W.S. Kang, Applying cubic b-spline quasi-interpolation to solve hyperbolic conservation laws, UPB Sci. Bull., Series D, **72** (2010) 49-58.

[21] S. Eddargani, A. Lamnii, M. Lamnii, D. Sbibih, A. Zidna, Algebraic hyperbolic spline quasi-interpolants and applications, JCAM, **347** (2019), 196-209.

[22] V. Chandraker, A. Awasthi, S. Jayaraj, Numerical Treatment of Burger-Fisher equation, Procedia Technology, **25** (2016), 1217-1225.

[23] C.G. Zhu, W.S. Kang, Numerical solution of Burgers-Fisher equation by cubic B-spline quasi-interpolation, Applied Mathematics and Computation, **216** (2010), 2679-2686.

[24] Y. Lü, G. Wang, X. Yang, Uniform hyperbolic polynomial B-spline curves, Comput. Aided Geom. Design. **19** (2002), 379-393.

[25] K. K. Sharma, P. Singh, Hyperbolic partial differential-difference equation in the mathematical modeling of neuronal firing and its numerical solution, Applied Mathematics and Computation, **201** (2008), 229-238.