Chapter 2

Numerical solution of coupled type fractional order Burgers' equation using finite difference and Fibonacci collocation method

2.1 Introduction

The system of Burgers' equations is the most common nonlinear time-dependent partial differential equation consisting of both nonlinear propagation and diffusive effects. Things become even more complicated when dealing with Burgers' equations in coupled form as it consists of the information regarding two different solutions depending on each other. Due to its wide range of applicability in various fileds of science, finding the best solutions of the Burgers' equations have always been the hot topic to the researchers [59]. Many researchers have solved the coupled Burgers' equations (CBEs) of integer order analytically with prescribed initial and boundary conditions. Kaya [60] has given an explicit solutions of CBEs by considering some particular initial conditions with the help of the decomposition method. Soliman [61] has solved the CBEs using $tanh$-function. Mohamed and Torky [62] have given an analytical solutions of CBEs by considering some particular initial and boundary conditions. As we know that finding an analytical solution is not always possible when dealing with PDEs. Things become more complex to deal with nonlinear coupled PDEs. That is why researchers are seeking their interest in developing an efficient method to solve this type of PDEs numerically. Many researchers have solved the CBEs of integer order numerically viz., Ahmed [63] has solved the CBEs numerically using variational iteration algorithm. Abazari and Borhanifar [64] have used the differential transformation method to solve the CBEs. Bak et al. [65] have solved the CBEs with the help of the semi-Lagrangian approach.

The study of fractional order nonlinear diffusion equations is quite limited. The growing interest in fractional order diffusion equations (FDEs) is due to their numerous useful applications in various engineering areas. It is used for an accurate description of the solute transportation in complex media like porous aquifers. Due to the nonlocal property of fractional derivative, FDEs have comprehensive applications in the fields of engineering, physics, economics, etc. Also due to nonlocal property, FDEs have a greater memory effect that for the standard order diffusion equation. Time, space, and time-space FDEs are extensively used to describe physical and engineering problems like anomalous diffusion, solute transport, signal processing, control, etc. For the description and understanding of dispersion phenomena, the FDEs have fundamental importance and have received a lot of attention during last few decades [66–79]. Thus the scientists and engineers are involved to find the solutions of FDEs for their non-local behavior, greater flexibility in models and its convergence to the integer order systems. The current study is mainly focused on developing an efficient numerical technique to deal with fractional order nonlinear coupled type diffusion equations. From the above discussions, we can say that this type of model contains almost every type of complexity. The main goal of this chapter is not only developing the numerical method but also making sure that the developed technique is performing far better than previously existing numerical techniques that deal with the said types of complex models.

In this chapter a numerical method has been developed to solve the following non-linear fractional-order coupled Burgers' equations as given by

$$\frac{\partial u}{\partial t} - \epsilon \frac{\partial^\beta u}{\partial x^\beta} + a_0 u \frac{\partial u}{\partial x} + a_1 \Big(u \frac{\partial v}{\partial x} + v \frac{\partial u}{\partial x} \Big) = 0, \tag{2.1}$$

$$\frac{\partial v}{\partial t} - \epsilon \frac{\partial^\beta v}{\partial x^\beta} + b_0 v \frac{\partial v}{\partial x} + b_1 \Big(u \frac{\partial v}{\partial x} + v \frac{\partial u}{\partial x} \Big) = 0, \tag{2.2}$$

under the initial conditions

$$u(x, 0) = f_1(x), \;\; v(x, 0) = f_2(x), \;\; a \leq x \leq b, \tag{2.3}$$

and the boundary conditions

$$\begin{aligned} u(a, t) &= g_1(t), \;\; , u(b, t) = g_2(t), \\ v(a, t) &= h_1(t), \;\; , v(b, t) = h_2(t), \;\; t \in [0, T], \end{aligned} \tag{2.4}$$

where a_0, b_0, a_1 and b_1 are constants and ϵ is positive kinematic viscous parameter depending on Reynolds number, $\epsilon = \frac{1}{\text{Reynolds Number}}$. β $(1 < \beta < 2)$ is order of fractional spatial derivative.

Physically the above problem can be explained as two types of interaction of pollutes ($a_1 \neq 0$ and $b_1 \neq 0$) with concentrations u and v have taken place in the porous medium. This model governs the effect on the solute concentrations by the pollutes due to mutual interaction, random motion and flow of the medium. If one choose $a_1 = b_1 = 0$, then there is no interaction taking place between these two pollutants. Thus, the effect on their concentrations is caused by the transportation of the particles into the medium individually.

In the present study a non-standard finite difference collocation (NSFDC) method is developed with a Fibonacci polynomial, where the solution in series form is approximated with Fibonacci polynomial and the constants are considered as function of t. After calculating the residue of the CBEs, the collocation method is used to get the required number of a linear equations to obtain the unknown constants.

2.2 Properties of Fibonacci Polynomial

It is known that Fibonacci polynomial can be constructed by the following recurrence relation

$$F_{m+2}(x) = xF_{m+1}(x) + F_m(x), \quad m \geq 0,$$

with initial conditions as

$$F_0(x) = 0, \quad F_1(x) = 1.$$

From the above relation, the explicit form of the series is obtained as

$$F_m(x) = \sum_{r=0}^{\lfloor \frac{m-1}{2} \rfloor} \binom{m-r-1}{r} x^{m-2r-1} \ , \tag{2.5}$$

where $\lfloor \cdot \rfloor$ denotes the floor function, which is defined by $\lfloor \alpha \rfloor = \max\{n \in Z | n \leq \alpha\}$. Above equation can be rewritten as

$$F_i(x) = \sum_{j=0}^{i} \frac{(\frac{i+j-1}{2})!}{j!(\frac{i-j-1}{2})!} x^j, \ \ (j+i) = odd, \ \ i \geq 0. \tag{2.6}$$

The function $f(x)$ whose squared integrable is written in terms of Fibonacci polynomial as [80]

$$f(x) = \sum_{k=1}^{\infty} c_k F_k, \tag{2.7}$$

where

$$c_k = \sum_{j=0}^{\infty} \frac{k(-1)^j f^{(2j+k-1)}(0)}{j!(j+k)!}.$$

The series (2.7) is approximated by $(n+1)$ termed finite sum as given below

$$f_n(x) = \sum_{k=1}^{n+1} c_k F_k, \tag{2.8}$$

where the unknowns c_k's have to be determined.

2.3 Numerical solution using NSFDC scheme

The fractional order β-th derivative of the function $f_n(x)$ can be approximated in terms of Fibonacci polynomial as

$$D^{\beta}[f_n(x)] = \sum_{k=\lceil\beta\rceil+1}^{n+1} \sum_{\substack{i=\lceil\beta\rceil \\ (i+k)=odd}}^{k} c_k \frac{(\frac{k+i-1}{2})!}{(\frac{k-i-1}{2})!\Gamma(i+1-\beta)} x^{i-\beta}. \tag{2.9}$$

Let us consider the approximation of the solutions of the nonlinear coupled Burgers' equations (2.1) with the help of Fibonacci polynomial as follows

$$\begin{aligned} u(x,t) &= \sum_{i=1}^{n+1} b_i(t) F_i(x), \\ v(x,t) &= \sum_{j=1}^{n+1} c_j(t) F_j(x). \end{aligned} \tag{2.10}$$

Considering the expressions given in (2.10) at the time level m, one can obtain

$$\begin{aligned} u(x,t_m) &= \sum_{i=1}^{n+1} b_i^m F_i(x), \\ v(x,t_m) &= \sum_{j=1}^{n+1} c_j^m F_j(x). \end{aligned} \tag{2.11}$$

Now writing the nonlinear coupled Burgers' equation at time level m with the help of equation (2.11), we obtain

$$\frac{\partial u(x,t_m)}{\partial t} - \epsilon \frac{\partial^\beta u(x,t_m)}{\partial x^\beta} + a_0 u(x,t_m)\frac{\partial u(x,t_m)}{\partial x} + a_1\Big(u(x,t_m)\frac{\partial v(x,t_m)}{\partial x} + v(x,t_m)\frac{\partial u(x,t_m)}{\partial x}\Big) = 0, \tag{2.12}$$

$$\frac{\partial v(x,t_m)}{\partial t} - \epsilon \frac{\partial^\beta v(x,t_m)}{\partial x^\beta} + b_0 v(x,t_m)\frac{\partial v(x,t_m)}{\partial x} + b_1\Big(u(x,t_m)\frac{\partial v(x,t_m)}{\partial x} + v(x,t_m)\frac{\partial u(x,t_m)}{\partial x}\Big) = 0. \tag{2.13}$$

As we all agree that dealing with nonlinear equations is complicated in comparison to dealing with linear equations. So the nonlinear terms of the above equations (2.12) and (2.13) can be linearized with the help of Taylor's expansion as

$$\begin{aligned} u(x,t_m)\frac{\partial u(x,t_m)}{\partial x} &= u(x,t_m)\frac{\partial u(x,t_{m-1})}{\partial x} + u(x,t_{m-1})\frac{\partial u(x,t_m)}{\partial x} - u(x,t_{m-1})\frac{\partial u(x,t_{m-1})}{\partial x}, \\ v(x,t_m)\frac{\partial v(x,t_m)}{\partial x} &= v(x,t_m)\frac{\partial v(x,t_{m-1})}{\partial x} + v(x,t_{m-1})\frac{\partial v(x,t_m)}{\partial x} - v(x,t_{m-1})\frac{\partial v(x,t_{m-1})}{\partial x}, \\ u(x,t_m)\frac{\partial v(x,t_m)}{\partial x} &= u(x,t_m)\frac{\partial v(x,t_{m-1})}{\partial x} + u(x,t_{m-1})\frac{\partial v(x,t_m)}{\partial x} - u(x,t_{m-1})\frac{\partial v(x,t_{m-1})}{\partial x}, \\ v(x,t_m)\frac{\partial u(x,t_m)}{\partial x} &= v(x,t_m)\frac{\partial u(x,t_{m-1})}{\partial x} + v(x,t_{m-1})\frac{\partial u(x,t_m)}{\partial x} - v(x,t_{m-1})\frac{\partial u(x,t_{m-1})}{\partial x}, \end{aligned} \tag{2.14}$$

Now on using above equations (2.14) in the equations (2.12) and (2.13), we have

$$\frac{\partial u(x,t_m)}{\partial t} - \epsilon \frac{\partial^\beta u(x,t_m)}{\partial x^\beta} + R_1^m(x) = 0, \tag{2.15}$$

$$\frac{\partial v(x,t_m)}{\partial t} - \epsilon \frac{\partial^\beta v(x,t_m)}{\partial x^\beta} + R_2^m(x) = 0, \tag{2.16}$$

where

$$\begin{aligned} R_1^m =& a_0\Big(u(x,t_m)\frac{\partial u(x,t_{m-1})}{\partial x} + u(x,t_{m-1})\frac{\partial u(x,t_m)}{\partial x} - u(x,t_{m-1})\frac{\partial u(x,t_{m-1})}{\partial x}\Big) \\ &+ a_1\Big(u(x,t_m)\frac{\partial v(x,t_{m-1})}{\partial x} + u(x,t_{m-1})\frac{\partial v(x,t_m)}{\partial x} - u(x,t_{m-1})\frac{\partial v(x,t_{m-1})}{\partial x} \\ &+ v(x,t_m)\frac{\partial u(x,t_{m-1})}{\partial x} + v(x,t_{m-1})\frac{\partial u(x,t_m)}{\partial x} - v(x,t_{m-1})\frac{\partial u(x,t_{m-1})}{\partial x}\Big), \end{aligned} \tag{2.17}$$

$$\begin{aligned} R_2^m =& b_0\Big(v(x,t_m)\frac{\partial v(x,t_{m-1})}{\partial x} + v(x,t_{m-1})\frac{\partial v(x,t_m)}{\partial x} - v(x,t_{m-1})\frac{\partial v(x,t_{m-1})}{\partial x}\Big) \\ &+ b_1\Big(u(x,t_m)\frac{\partial v(x,t_{m-1})}{\partial x} + u(x,t_{m-1})\frac{\partial v(x,t_m)}{\partial x} - u(x,t_{m-1})\frac{\partial v(x,t_{m-1})}{\partial x} \\ &+ v(x,t_m)\frac{\partial u(x,t_{m-1})}{\partial x} + v(x,t_{m-1})\frac{\partial u(x,t_m)}{\partial x} - v(x,t_{m-1})\frac{\partial u(x,t_{m-1})}{\partial x}\Big). \end{aligned} \tag{2.18}$$

Now using the expressions (2.9) and (2.10) in the equations (2.15) and (2.16), we have

$$\sum_{i=1}^{n+1}\frac{db_i^m}{dt}F_i(x) - \epsilon\sum_{i=\lceil\beta\rceil+1}^{n+1}\sum_{\substack{k=\lceil\beta\rceil\\(i+k)=odd}}^{i} b_i^m\frac{(\frac{i+k-1}{2})!}{(\frac{i-k-1}{2})!\mathit{\Gamma}(k+1-\beta)}x^{k-\beta} + R_1^m(x) = 0, \tag{2.19}$$

$$\sum_{j=1}^{n+1}\frac{dc_j^m}{dt}F_j(x) - \epsilon\sum_{j=\lceil\beta\rceil+1}^{n+1}\sum_{\substack{k=\lceil\beta\rceil\\(j+k)=odd}}^{j} c_j^m\frac{(\frac{j+k-1}{2})!}{(\frac{j-k-1}{2})!\mathit{\Gamma}(k+1-\beta)}x^{k-\beta} + R_2^m(x) = 0. \tag{2.20}$$

Rewriting the above two equations (2.19) and (2.20), we get

$$\sum_{i=1}^{n+1}\frac{b_i^m - b_i^{m-1}}{\phi(h)}F_i(x) - \epsilon\sum_{i=\lceil\beta\rceil+1}^{n+1}\sum_{\substack{k=\lceil\beta\rceil\\(i+k)=odd}}^{i} b_i^m\frac{(\frac{i+k-1}{2})!}{(\frac{i-k-1}{2})!\mathit{\Gamma}(k+1-\beta)}x^{k-\beta} + R_1^m(x) = 0, \tag{2.21}$$

$$\sum_{j=1}^{n+1}\frac{c_j^m - c_j^{m-1}}{\phi(h)}F_j(x) - \epsilon\sum_{j=\lceil\beta\rceil+1}^{n+1}\sum_{\substack{k=\lceil\beta\rceil\\(j+k)=odd}}^{j} c_j^m\frac{(\frac{j+k-1}{2})!}{(\frac{j-k-1}{2})!\mathit{\Gamma}(k+1-\beta)}x^{k-\beta} + R_2^m(x) = 0. \tag{2.22}$$

The above two equations (2.21) and (2.22) together with boundary conditions will help us of finding the unknowns $b_k's$ and $c_k's$ at each time level other than zero time level. Initial conditions will help us to find the values at zero time level. Hence rewriting initial conditions (2.3) with the help of (2.11), we get

$$\sum_{i=1}^{n+1} b_i^0 F_i(x) = f_1(x), \ \sum_{j=1}^{n+1} c_i^0 F_i(x) = f_2(x), \ a \leq x \leq b. \tag{2.23}$$

Now rewriting boundary conditions (2.4) with the help of (2.11) at time level m, we obtain

$$\begin{aligned} \sum_{i=1}^{n+1} b_i^m F_i(a) = g_1(t_m), \ \sum_{i=1}^{n+1} b_i^m F_i(b) = g_2(t_m), \\ \sum_{j=1}^{n+1} c_j^m F_i(a) = h_1(t_m), \ \sum_{j=1}^{n+1} c_j^m F_i(b) = h_2(t_m). \end{aligned} \tag{2.24}$$

From the expressions (2.11) it is clear that total $(2n+1)$ unknowns are obtained at each time level m. So one can collocate equations (2.21) and (2.22) at $(n-1)$ collocation points $x = \frac{i}{n+1}, \ i = 1, 2, 3, ..., n-1$. In this way togather with four boundary conditions given in (2.24), we will have $(2n+1)$ algebraic linear equations in $b_k's$ and $c_k's$, which we can solve easily with appropriate numerical method. To start the process of calculation of unknowns, we need the $b_k's$ and $c_k's$ at zero time level which can be calculated by collocating the initial conditions (2.23) at the set collocation points and using the boundary conditions (2.24) at zero time level.

2.4 Numerical Applications

In this section, the proposed scheme has been applied on two problems having exact solutions and compared the numerical results obtained by the proposed method with the existing results through error analysis. During comparison of the numerical error, the L_2 and L_∞ errors have been calculated, which are defined as

$$L_2 = \sqrt{h \sum_{j=1}^{m} |u_j - U_j|^2}, \qquad h = x_{i+1} - x_i \tag{2.25}$$

$$L_\infty = \underset{j}{Max} |u_j - U_j|, \tag{2.26}$$

where u and U are numerical and exact solutions, respectively.

Example 1. Let us consider the following example

$$u_t - u_{xx} - 2uu_x + uv_x + vu_x = 0, \tag{2.27}$$

$$v_t - v_{xx} - 2vv_x + uv_x + vu_x = 0, \tag{2.28}$$

with the initial conditions as

$$u(x, 0) = v(x, 0) = sin\ x, \ \ -\pi \leq x \leq \pi, \tag{2.29}$$

and the boundary conditions as

$$u(-\pi, t) = u(\pi, t) = 0, \tag{2.30}$$

$$v(-\pi, t) = v(\pi, t) = 0, \ \ t \in [0, T], \tag{2.31}$$

which has the exact solutions given by

$$u(x, t) = v(x, t) = exp(-t) sin\ x.$$

The problem is solved for $n = 17$ and $\Delta t = 0.01$ with the proposed method and a comparison of error analysis between the exact solution and the numerical results obtained by different techniques is presented in Table 2.1. It is clearly observed from Table 2.1 that the proposed method performs better as compared to previously solved methods even for very less order of approximation and discretization of time ($\Delta t = 0.01$). It is also seen that in other methods viz., Onarcan and Hepson [81] have used trigonometric B-spline algorithm to achieve the minimum error of order $L_\infty \times 10^6 = 392.13$, Mittal and Jiwari [82] used differential quadrature method to get the error of order $L_\infty \times 10^6 = 74.601$, Bhatt et al. used [83] fourth-order compact schemes to find the minimum error $L_\infty \times 10^6 = 1.193$, while in the proposed scheme it is obtained as $L_\infty \times 10^6 = 1.56073e - 6$.

Example 2. Let us consider the following example

$$u_t - u_{xx} + 2uu_x + \alpha(uv_x + vu_x) = 0, \tag{2.32}$$

$$v_t - v_{xx} + 2vv_x + \beta(uv_x + vu_x) = 0, \tag{2.33}$$

which have the exact solutions as

$$u(x,t) = a_0(1 - tanh(k(x - 2kt))), \tag{2.34}$$

$$v(x,t) = a_0\Big(\frac{2\beta - 1}{2\alpha - 1} - tanh(k(x - 2kt))\Big), \tag{2.35}$$

where $k = a_0\Big(\frac{4\alpha\beta - 1}{4\alpha - 2}\Big)$, and a_0, α and β are constants. The initial and boundary conditions are extracted from the exact solutions. In this problem a comparison of errors with other existing methods are given in Table 2.2. It is very clear from the table that the proposed method is performing well as compared to an existing method [84] even for less temporal discretization. Bhatt et al. [83] have also solved

TABLE 2.1: Comparison of numerical results with different methods for Example 1.

Method	n	t	$L_2 \times 10^6$	$L_\infty \times 10^6$	Method	n	t	$L_2 \times 10^6$	$L_\infty \times 10^6$
Our proposed scheme	17	0.1	3.16812e-5	3.1173e-4	DQM [84]	131	0.1	2.26096	3.78768
		0.5	9.84944e-5	6.83079e-4			0.5	2.70172	2.61424
		1.0	1.49673e-5	1.00919e-4			1.0	2.48146	1.57097
		2.0	7.20063e-6	3.68887e-5			2.0	1.61674	1.00892
		3.0	3.31726e-6	1.54191e-5			3.0	0.86332	0.54705
		4.0	4.69968e-6	4.21068e-5			4.0	0.42518	0.27676
		5.0	2.67499e-6	1.63987e-5			5.0	0.20099	0.13728
		10.0	3.49104e-7	1.56073e-6			10.0	0.01748	0.01017
LBM [85]	64	0.1	30.3	27.48	LB [86]	64	0.1	28.3	22.826
		0.5	151.7	92.04			0.5	117.8	80.420
		1.0	303.4	111.66			1.0	170.4	66.441
		2.0	607.0	82.11			2.0	309.0	58.983
		5.0	1518.1	10.22			5.0	1015.3	8.098
		10.0	3038.6	0.13			10.0	2278.5	0.112
FDM [85]	64	0.1	80.2	72.68	Coll. [87]	200	0.1	8.2	7.4
		0.5	401.5	243.54			0.5	24.9	41.0
		1.0	803.2	295.46			1.0	30.0	82.1
		2.0	1607.1	217.52			2.0	-	-
		5.0	4022.7	27.10			5.0	-	-
		10.0	8061.7	0.36			10.0	-	-
Coll. [88]	100	0.1	32.9024	29.7713	Coll. [87]	400	0.1	2.0	1.86
		0.5	164.5015	99.7752			0.5	10.2	6.22
		1.0	328.9759	121.0234			1.0	20.4	7.56
Coll. [88]	128	0.1	30.1811	18.4560	Imp.FD [89]	200	0.1	58.6	53.0
		0.5	39.4088	61.8548			0.5	294.0	179.0
		1.0	28.0808	11.2589			1.0	591.0	217.0
Gal. [90]	64	0.1	1.3961	3.9846					
		0.5	2.4739	2.8698					
		1.0	3.5300	1.7864					
		2.0	5.5176	0.7300					
		3.0	7.8953	0.3903					

above problem with fourth-order scheme and obtained the maximum error of order 10^{-5} for $n = 16$ and $n = 20$. Khater et al. [91] have used a Chebyshev spectral collocation method to obtain the maximum absolute error of order 10^{-5} for $n = 20$. Rashid et al. [92] and Mittal and Jiwari [82] have used different types of numerical schemes to obtain absolute error order 10^{-5} for $n = 20$. But that much level of accuracy is obtained for a very less number of approximation $(n = 3)$ and less order of temporal discretization while applying the proposed method on the mentioned problem.

After the validation of the accuracy and efficiency of the proposed method while

TABLE 2.2: Comparison of numerical results with different methods for Example 2.

Method	n	Δ t	α	β	t	$L_2(u)\times 10^3$	$L_\infty(u)\times 10^3$	$L_2(v)\times 10^3$	$L_\infty(v)\times 10^3$
DQM [84]	3	0.1	0.1	0.3	0.1	0.213635	0.067569	0.109682	0.034684
					0.3	0.639770	0.202348	0.328518	0.103886
					0.5	1.064579	0.336650	0.546654	0.172867
					0.7	1.487774	0.470476	0.764093	0.241627
					1.0	2.119762	0.670328	1.088950	0.344356
Our proposed scheme	3	0.1	0.1	0.3	0.1	0.00812135	0.00898543	0.00303799	0.00446763
					0.3	0.0244485	0.0274415	0.00930321	0.012705
					0.5	0.0407458	0.045736	0.0155668	0.0212789
					0.7	0.0570096	0.0638709	0.0218201	0.0297751
					1.0	0.0813408	0.0907786	0.0311794	0.0423763
DQM [84]	3	0.01	0.3	0.03	0.1	0.250138	0.079101	0.477433	0.150978
					0.3	0.749491	0.237010	1.430031	0.452216
					0.5	1.247615	0.394531	2.379619	0.752501
					0.7	1.744516	0.551664	3.326207	1.051839
					1.0	2.487583	0.786643	4.740498	1.499077
Our proposed scheme	3	0.01	0.3	0.03	0.1	0.00507463	0.00738132	0.00848107	0.0122781
					0.3	0.0151924	0.0220816	0.025398	0.0367252
					0.5	0.0252694	0.036693	0.0422557	0.0610205
					0.7	0.0353061	0.051216	0.0590544	0.0851642
					1.0	0.0502862	0.0728356	0.084142	0.094347

applying it on the two existing integer order problems mentioned above, a derive has been taken to apply it on the nonlinear fractional-order CBEs given in (2.1) and (2.2) and to show the effect of spatial fractional order derivative on the solution profiles for different cases.

Let us consider the equations (2.1) and (2.2) under the prescribed initial conditions as

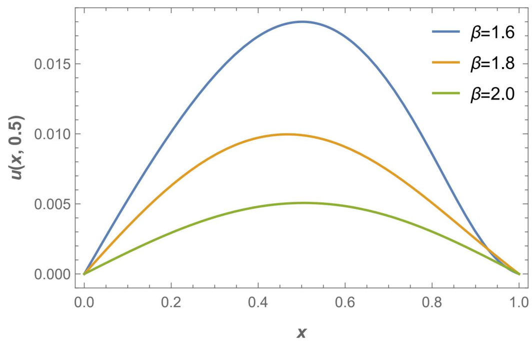
$$u(x,0) = v(x,0) = sin\,\pi x,\;\; 0 \le x \le 1, \tag{2.36}$$

and the boundary conditions as

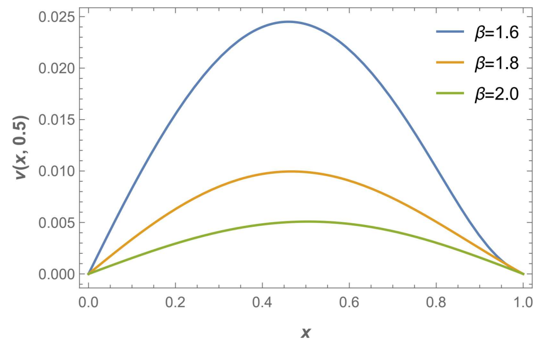
$$u(0,t) = u(1,t) = 0, \tag{2.37}$$

$$v(0,t) = v(1,t) = 0,\;\; t \in [0,T], \tag{2.38}$$

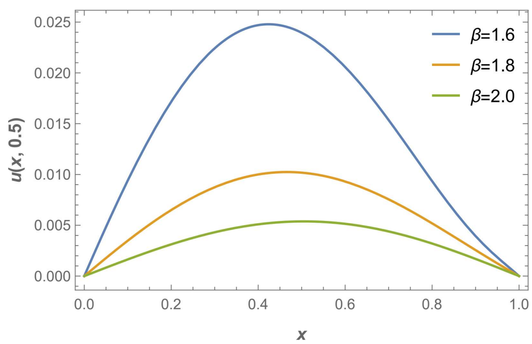
The fractional-order CBEs has been tackled with the help of proposed method and



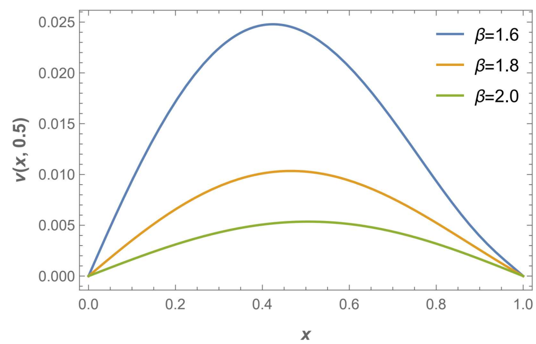
(a) for $a_0 = b_0 = 0.2$



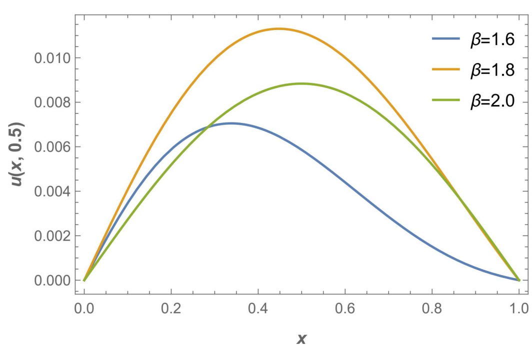
(b) for $a_0 = b_0 = 0.2$



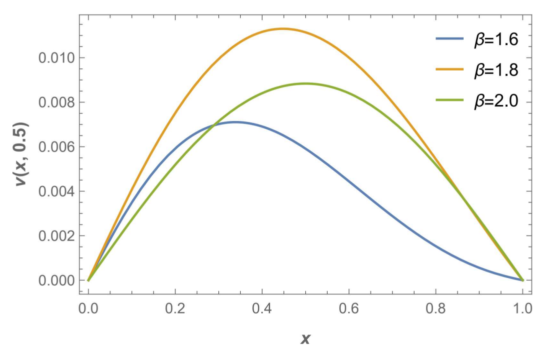
(c) for $a_0 = b_0 = 2.0$



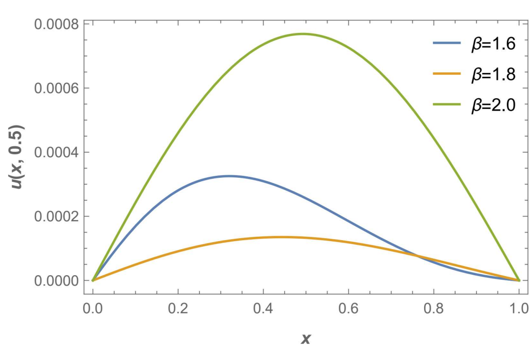
(d) for $a_0 = b_0 = 2.0$



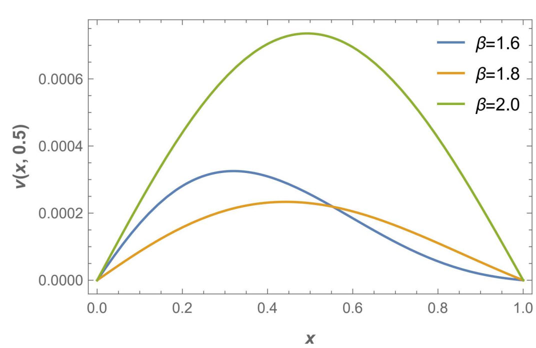
(e) for $a_0 = b_0 = 20$



(f) for $a_0 = b_0 = 20$

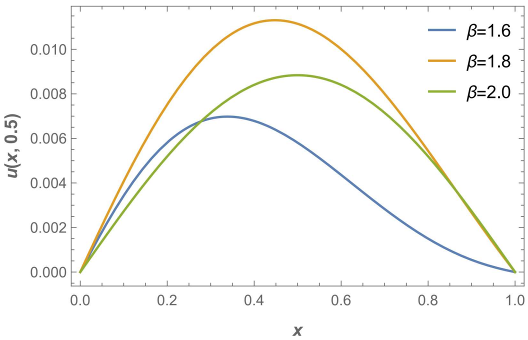


(g) for $a_0 = b_0 = 200$

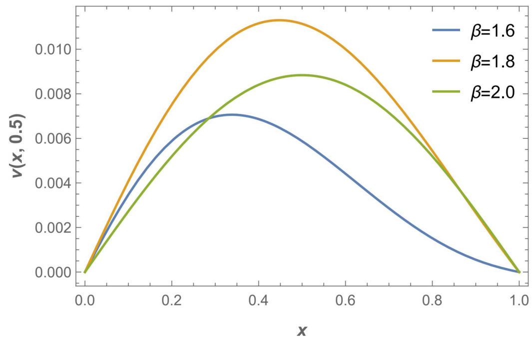


(h) for $a_0 = b_0 = 200$

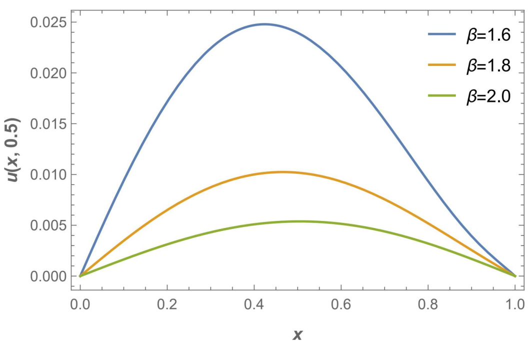
FIGURE 2.1: Simulation of fractional-order CBEs for $a_1 = b_1 = 10$ at $t = 0.5$



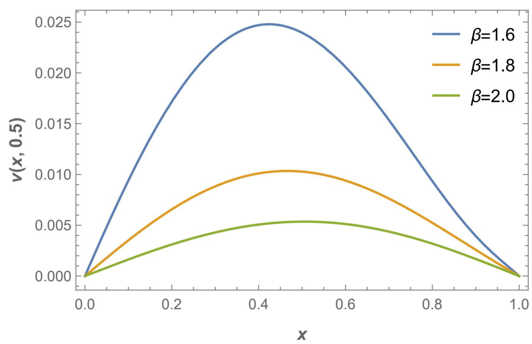
(a) for $a_1 = b_1 = 1$



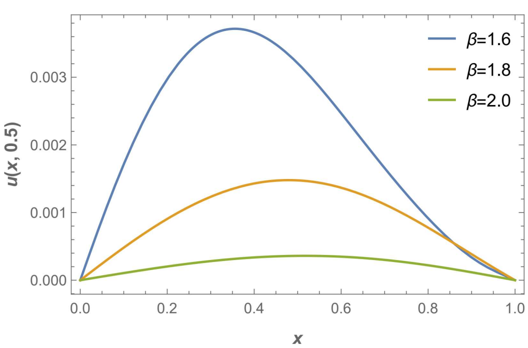
(b) for $a_1 = b_1 = 1$



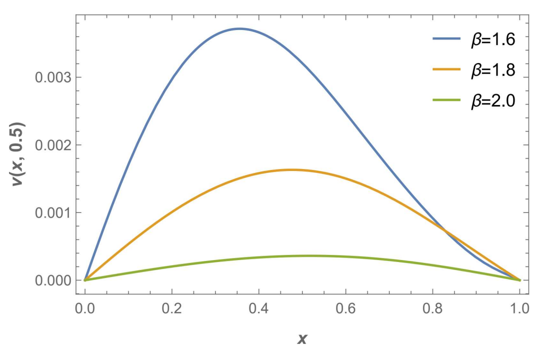
(c) for $a_1 = b_1 = 10$



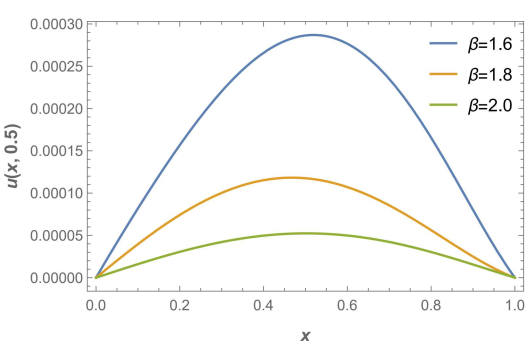
(d) for $a_1 = b_1 = 10$



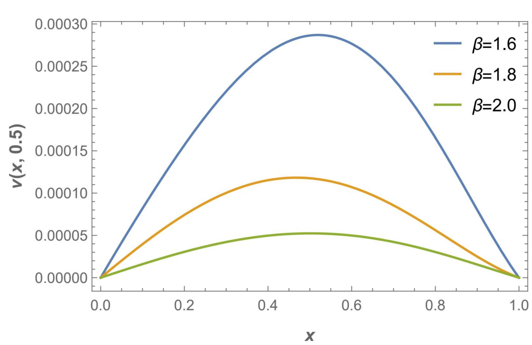
(e) for $a_1 = b_1 = 100$



(f) for $a_1 = b_1 = 100$



(g) for $a_1 = b_1 = 1000$



(h) for $a_1 = b_1 = 1000$

FIGURE 2.2: Simulation of fractional-order CBEs for $a_0 = b_0 = 2$ at $t = 0.5$

observe the behavior of $u(x,t)$ and $v(x,t)$ due to the change in spatial fractional order derivative for the different values of parameters a_0, a_1, b_0 and b_1. In the Figure 2.1, eight sub-figures have been plotted for different values of $a_0 = b_0$ for a fixed value of $a_1 = b_1 = 10$ at time $t = 0.5$. From the sub-figures a, b, c and d of the Figure 2.1, it is clear that both $u(x,t)$ and $v(x,t)$ are decreasing with increase in the order of spatial derivative. But on increasing the values of $a_0 = b_0$, both the solutions start behaving irregularly in the sub-figures 2.1(e), (f), (g) and (h). In the Figure 2.2, eight sub-figures are plotted for different values of $a_1 = b_1$ for a fixed value of $a_0 = b_0 = 2$ at time $t = 0.5$. From the sub-figures 2.2(a) and (b) of the Figure 2.2, it is clearly seen that at very low value of $a_1 = b_1 = 1$, solutions $u(x,t)$ and $v(x,t)$ are behaving irregularly but as the values of $a_1 = b_1$ are increased, the solutions start behaving regularly. From the sub-figures 2.2(c)-(h), it is also observed that for higher values of $a_1 = b_1$, the solutions start decreasing on increasing the values of spatial order derivative β.

2.5 Conclusion

It is well known that getting an exact solution of nonlinear PDEs is not always possible, and things become more complex when it comes in the form of nonlinear coupled PDEs. So the only way left is to simulate the solution numerically. In the present study, a non-standard numerical technique has been developed with the help of Fibonacci polynomials to show its performance over the other existing methods. Two numerical examples of integer order CBEs are considered to validate the effectiveness of the proposed numerical scheme and then it is applied to find the solution of a nonlinear coupled spatial fractional order Burgers' equations. The behavior of solutions of the coupled problem has been observed through graphical representation by the change in various parameters presented in the model. It can be concluded here that the proposed scheme will help the engineers and scientists working in the area of coupled nonlinear PDEs in integer as well as fractional order systems.

****