

Article

Fractional Bernstein Series Solution of Fractional Diffusion Equations with Error Estimate

Mohammed Hamed Alshbool ^{1,*} , Osman Isik ² and Ishak Hashim ³ 

¹ Department of Applied Mathematics, Abu Dhabi University, Abu Dhabi 59911, UAE

² Elementary Mathematics Education Program, Faculty of Education, Mugla Sitki Kocman University, Mugla 48000, Turkey; osmanrasitisik@hotmail.com

³ Department of Mathematical Sciences, Faculty of Science & Technology, Universiti Kebangsaan Malaysia, UKM Bangi 43600, Malaysia; ishak_h@ukm.edu.my

* Correspondence: alshbool.mohammed@gmail.com

Abstract: In the present paper, we introduce the fractional Bernstein series solution (FBSS) to solve the fractional diffusion equation, which is a generalization of the classical diffusion equation. The Bernstein polynomial method is a promising one and can be generalized to more complicated problems in fractional partial differential equations. To get the FBSS, we first convert all terms in the problem to matrix forms. Then, the fundamental matrix equation is obtained and thus, the solution is obtained. Two error estimation methods based on a residual correction procedure and the consecutive approximations are incorporated to find the estimate and bound of the absolute error. The perturbation and stability analysis of the method is given. We apply the method to some illustrative examples. The numerical results are compared with the exact solutions and known second-order methods. The outcomes of the numerical examples are very encouraging and show that the FBSS is highly useful in solving fractional partial problems. The results show the accuracy and effectiveness of the method.

Keywords: bernstein series; fractional calculus; diffusion equations; error estimate



Citation: Alshbool, M.H.; Isik, O.; Hashim, I. Fractional Bernstein Series Solution of Fractional Diffusion Equations with Error Estimate. *Axioms* **2021**, *10*, 6. <https://doi.org/10.3390/axioms10010006>

Received: 15 December 2020

Accepted: 4 January 2021

Published: 7 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Fractional derivatives have been used to model many problems in science, e.g., physics [1–3], medicine [4], hydrology [5], biomedical problems [6], dynamics of particles [7] and applied sciences [8]. The linear fractional diffusion equation is considered for scientists and engineers [9]. Fractional space derivatives are used in the modeling of anomalous diffusion. In a diffusion model, replacement of the second derivative by a fractional derivative causes enhanced diffusion, also called superdiffusion [10].

In the present study, we consider the following one-dimensional fractional diffusion equation (FDE),

$$\frac{\partial y(x, t)}{\partial t} = d(x) \frac{\partial^\alpha y(x, t)}{\partial x^\alpha} + g(x, t), \quad (1)$$

on a finite domain $L < x < R$, $1 < \alpha \leq 2$ and $d(x) > 0$. We assume the initial condition $c(x, t = 0) = F(x)$ for $L < x < R$ and the boundary conditions $c(x = L, t) = 0$ and $y(x = R, t) = b_R(t)$.

The fractional derivative in Equation (1) is the Caputo fractional derivative of order α [11]. The basic property of the Caputo derivative is as follows.

$$D_*^\alpha c = 0, \quad (c \text{ constant}),$$

$$D_*^\alpha x^\beta = \begin{cases} 0, & \text{for } \beta \in N_0 \text{ and } \beta < [\alpha], \\ \frac{\Gamma(\beta+1)}{\Gamma(\beta+1-\alpha)} x^{\beta-\alpha}, & \text{for } \beta \in N_0 \text{ and } \beta \geq [\alpha] \text{ or } \beta > [\alpha]. \end{cases}$$

Several methods have been used to solve the FDE numerically, e.g., Crank–Nicholson method, which is second-order accurate [12]; finite difference method [13]; finite element method [14]; generalized differential transform method [15]; and Galerkin method, which includes the normalized Bernstein polynomials, collocation method [16], the shifted Jacobi tau method [17], and the Chebyshev spectral-tau method [18]. Fractional Taylor vector to solve multi-term fractional differential equations [19].

One effective method to solve the problems is the Bernstein polynomials method (BPM), also called the Bernstein series solution method [20–25]. Positive linear operators of Bernstein with sequence of these operators are established to solve to solve two-variables equations [26]. The operational matrices of the BPM have been used to get the numerical solutions of a class of third-order ordinary differential equations [27]. The multi-stage BPM, a generalization of the standard BPM, was applied to fractional-order stiff systems [28]. The BPM with new modifications was employed to solve fractional differential equations [29]. Moreover, the same method was used to solve some types of ordinary differential equations [30–32]. A special type of the singular Emden-Fowler problems was solved by BPM [33]. Recently, some linear and non-linear systems of ordinary differential equations have been solved by BPM and the accuracy has been improved by a residual correction procedure [34]. A novel error estimation method for the parametric non-intrusive reduced order model based on machine learning is presented in [35].

In this paper, the fractional Bernstein series solution (FBSS) method is introduced and applied to solve Equation (1) numerically. The method comprises the fractional Bernstein polynomials and collocation method. We approximate the exact solution of Equation (1) by $f_{\alpha}p_{n,n}$

$$f_{\alpha}p_{n,n}(x, t) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} f_{\alpha} B_{i,n}(x) B_{j,n}(t),$$

such that $f_{\alpha}p_{n,n}$ satisfies Equation (1) on the collocation nodes.

This paper is organized as follows. Section 2 presents necessary definitions and theorems. Section 3, the main section, discusses the matrix forms of the solution $f_{\alpha}p_{n,n}$ with its derivatives. Then, we get the fundamental matrix equation of the FDE. Employing the conditions and applying the Gauss elimination procedure yields the unknown matrix. Thus, we obtain the FBSS. In Section 4, two different error analysis methods are presented to get the upper bound of the absolute error with the corrected FBSS. Moreover, an upper bound obtained by the generalized Taylor theorem is presented. To decide the stability, the perturbation and stability analysis of the method is done in Section 5. Section 6 provides the numerical results to illustrate the FBSS method for different n values. We compare the method with other methods used to solve the problem. The results show that the present method gives accurate solutions and is also efficient. We test the effect of small perturbations to the approximate solutions both theoretically and practically for various values of n . Section 7 summarizes the results.

2. Preliminaries and Notations

Bernstein polynomials of n -th degree are given by the following equation [29].

$$B_{t,n}(x) = \binom{n}{t} \frac{x^t (R - x)^{n-t}}{R^n}, \quad t = 0, 1, 2, \dots, n \quad x \in [0, R], \tag{2}$$

Similarly, the fractional Bernstein polynomials $B_{k,n}^{\alpha}(x)$ are constituted by using $x \rightarrow x^{\alpha}$, so Equation (2) becomes

$$\begin{aligned} B_{t,n}^{\alpha}(x) &= \binom{n}{t} \frac{x^{t\alpha} (R - x^{\alpha})^{n-t}}{R^n}, \quad 1 < \alpha < 2, \\ t &= 0, 1, 2, \dots, n \quad x \in [0, R], \end{aligned} \tag{3}$$

We will use the definition of Caputo’s derivative, which is the modification of the definition of Riemann-Liouville. It has some advantages for solving initial value problems [36].

Definition 1 ([37,38]). The Riemann–Liouville integral operator of order $\alpha > 0$ for $a \geq 0$ is defined as follows

$$(J_a^\alpha f)(x) = \frac{1}{\Gamma(\alpha)} \int_a^x (x-t)^{\alpha-1} f(t) dt, \quad x > a,$$

$$(J_a^0 f)(x) = f(x).$$

Definition 2 ([37,38]). The Caputo fractional derivative of f of order $\alpha > 0$ for $a \geq 0$ is given as follows,

$$(D_a^\alpha f)(x) = (J_a^{m-\alpha} f^{(m)})(x) = \frac{1}{\Gamma(m-\alpha)} \int_a^x \frac{f^{(m)}(t)}{(x-t)^{\alpha+1-m}} dt$$

for $m - 1 < \alpha \leq m, m \in \mathbb{N}, x \geq a$.

We will use the following multivariate fractional Taylor’s theorem [39] to bound the absolute error.

Theorem 1. For a compact and convex domain $D \subset \mathbb{R}^2$, let $D^{k\alpha} f \in C(D)$ for $k = 0, 1, \dots, m + 1$ where

$$D^{k\alpha} f = D^{k\alpha-n} D^n f, \quad n \text{ is the smallest integer exceeding } k\alpha$$

$$D^n f = \left(\Delta x \frac{\partial}{\partial x} + \Delta y \frac{\partial}{\partial y} \right)^n f.$$

If $(x_0, y_0) \in D$, then

$$f(x, y) = \sum_{k=0}^m \frac{D^{k\alpha} f(x_0, y_0)}{\Gamma(k\alpha + 1)} + \frac{D^{(m+1)\alpha} f(\xi, \eta)}{\Gamma((m+1)\alpha + 1)}$$

$$= P_m^\alpha(x, t) + R_m^\alpha(\xi, \eta)$$

where $\xi = x_0 + \theta \Delta x, \eta = y_0 + \theta \Delta y, 0 < \theta < 1$ and

$$P_m^\alpha(x, t) = \sum_{k=0}^m \frac{D^{k\alpha} f(x_0, y_0)}{\Gamma(k\alpha + 1)} \quad (\text{Truncated mult. frac. Taylor series}) \tag{4}$$

$$R_m^\alpha(\xi, \eta) = \frac{D^{(m+1)\alpha} f(\xi, \eta)}{\Gamma((m+1)\alpha + 1)} \quad (\text{Remainder term}).$$

3. Numerical Method

Let

$$f_\alpha p_{n,n}(x, t) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} f_\alpha B_{i,n}(x) B_{j,n}(t)$$

be the FBSS of Equation (1). Let us find the matrix forms of

$$f_\alpha p_{n,n}^\alpha = \frac{\partial^\alpha f_\alpha p_{n,n}}{\partial x^\alpha}, \quad 1 < \alpha \leq 2 \text{ and } \frac{\partial f_\alpha p_{n,n}}{\partial t}.$$

First note that $f_\alpha p_{n,n}$ can be written as follows:

$$f_\alpha p_{n,n}(x, t) = f_\alpha \mathbf{B}_n(x) \mathbf{Q}_n(t) \mathbf{A},$$

where

$$f_\alpha \mathbf{B}_n(x) = \left[B_{0,n}^\alpha(x) \quad B_{1,n}^\alpha(x) \quad \cdots \quad B_{n,n}^\alpha(x) \right],$$

$$\mathbf{Q}_n(t) = \begin{pmatrix} \mathbf{B}_n(t) & 0 & \cdots & 0 \\ 0 & \mathbf{B}_n(t) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{B}_n(t) \end{pmatrix},$$

and

$$\mathbf{A} = [a_{00} \quad a_{01} \cdots a_{0n} \quad a_{10} \quad a_{11} \cdots a_{1n} \cdots a_{n1} \quad a_{n2} \cdots a_{nn}]^T.$$

We write $f_\alpha \mathbf{B}_n(x)$ as

$$f_\alpha \mathbf{B}_n(x) = f_\alpha \mathbf{X}(x) \mathbf{D}^T,$$

where

$$\mathbf{D} = \begin{pmatrix} d_{00} & d_{01} & \cdots & d_{0n} \\ d_{10} & d_{11} & \cdots & d_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n0} & d_{n1} & \cdots & d_{nn} \end{pmatrix}, \quad f_\alpha \mathbf{X}(x) = [1 \quad x^\alpha \quad x^{2\alpha} \cdots x^{n\alpha}],$$

$$d_{ij} = \begin{cases} \frac{(-1)^{j-i}}{R^j} \binom{n}{i} \binom{n-i}{j-i}, & i \leq j \\ 0, & i > j \end{cases}.$$

Therefore, we can write $f_\alpha p_{n,n}$ and $f_\alpha p_{n,n}^\alpha$ in the forms

$$f_\alpha p_{n,n}(x, t) = f_\alpha \mathbf{B}_n(x) \mathbf{Q}_n(t) \mathbf{A}, \tag{5}$$

$$f_\alpha p_{n,n}^\alpha(x, t) = f_\alpha \mathbf{B}_n^\alpha(x) \mathbf{Q}_n(t) \mathbf{A}; \tag{6}$$

respectively. As $f_\alpha \mathbf{B}_n^\alpha(x)$ can be written as

$$f_\alpha \mathbf{B}_n^{(\alpha)}(x) = f_\alpha \mathbf{X}^{(\alpha)}(x) \mathbf{D}^T, \tag{7}$$

for $f_\alpha \mathbf{X}^{(\alpha)}(x)$, the relation

$$f_\alpha \mathbf{X}^{(\alpha)}(x) = f_\alpha \mathbf{X}(x) \mathbf{C}(\mathbf{ff}), \tag{8}$$

is obtained where

$$\mathbf{C}(\mathbf{ff}) = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \frac{\Gamma(2)}{\Gamma(2-\alpha)} x^{-\alpha} & 0 & \cdots & 0 \\ 0 & 0 & 0 & \frac{\Gamma(3)}{\Gamma(3-\alpha)} x^{-\alpha} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \frac{\Gamma(n+1)}{\Gamma(n+1-\alpha)} x^{-\alpha} \end{pmatrix}.$$

Substituting Equation (8) into Equation (7) we get

$$f_\alpha \mathbf{B}_n^{(\alpha)}(x) = f_\alpha \mathbf{X}(x) \mathbf{C}(\mathbf{ff}) \mathbf{D}^\top. \tag{9}$$

Note that $\mathbf{Q}_n(t)$ can be given as

$$\mathbf{Q}_n(t) = \bar{\mathbf{Y}}_n(t) \bar{\mathbf{D}}, \tag{10}$$

where

$$\bar{\mathbf{Y}}_n(\mathbf{t}) = \begin{pmatrix} \mathbf{Y}(t) & 0 & \dots & 0 \\ 0 & \mathbf{Y}(t) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{Y}(t) \end{pmatrix}, \quad \mathbf{Y}(t) = [1 \quad t \quad t^2 \dots t^n],$$

$$\bar{\mathbf{D}} = \begin{pmatrix} \mathbf{D}^\top & 0 & \dots & 0 \\ 0 & \mathbf{D}^\top & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{D}^\top \end{pmatrix}.$$

The term $\frac{\partial \mathbf{Q}_n(t)}{\partial t}$ can be written as:

$$\frac{\partial \mathbf{Q}_n(t)}{\partial t} = \frac{\partial \bar{\mathbf{Y}}_n(t)}{\partial t} \bar{\mathbf{D}}, \tag{11}$$

where

$$\frac{\partial \bar{\mathbf{Y}}_n(t)}{\partial t} = \bar{\mathbf{Y}}_n(\mathbf{t}) \bar{\mathbf{B}}, \tag{12}$$

and

$$\bar{\mathbf{B}} = \begin{pmatrix} \mathbf{B} & 0 & \dots & 0 \\ 0 & \mathbf{B} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{B} \end{pmatrix},$$

and

$$\mathbf{B} = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \dots & \dots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix}.$$

Substituting Equation (12) into Equation (11) we get

$$\frac{\partial \mathbf{Q}_n(t)}{\partial t} = \bar{\mathbf{Y}}(\mathbf{t}) \bar{\mathbf{B}} \bar{\mathbf{D}}. \tag{13}$$

Substituting Equations (9) and (13) into Equation (5) yields the matrix forms of $f_\alpha p_{n,n}(x, t)$, $f_\alpha p_{n,n}^\alpha(x, t)$ and $\frac{\partial f_\alpha p_{n,n}}{\partial t}$ as

$$f_\alpha p_{n,n}(x, t) = f_\alpha \mathbf{X}(x) \mathbf{D}^T \bar{\mathbf{Y}}(t) \bar{\mathbf{D}} \mathbf{A}, \tag{14}$$

$$f_\alpha p_{n,n}^\alpha(x, t) = f_\alpha \mathbf{X}(x) \mathbf{C}(\mathbf{ff}) \mathbf{D}^T \bar{\mathbf{Y}}(t) \bar{\mathbf{D}} \mathbf{A}, \tag{15}$$

$$\frac{\partial f_\alpha p_{n,n}}{\partial t} = f_\alpha \mathbf{X}(x) \mathbf{D}^T \bar{\mathbf{Y}}(t) \bar{\mathbf{B}} \bar{\mathbf{D}} \mathbf{A}, \tag{16}$$

respectively. The use of (15) and (16) in (1) gives the fundamental matrix equation

$$[f_\alpha \mathbf{X}(x) \mathbf{D}^T \bar{\mathbf{Y}}(t) \bar{\mathbf{B}} \bar{\mathbf{D}} - d(x) f_\alpha \mathbf{X}(x) \mathbf{C}(\mathbf{ff}) \mathbf{D}^T \bar{\mathbf{Y}}(t) \bar{\mathbf{D}}] \mathbf{A} = g(x, t). \tag{17}$$

Inserting the collocation points $\{(x_i, y_j) : 0 \leq i, j \leq n\}$ in Equation (17) yields the system

$$\mathbf{W} \mathbf{A} = \mathbf{G}, \tag{18}$$

where m -th row of \mathbf{W} is obtained from $\{(x_k, y_l), k = \lfloor \frac{m}{n+1} \rfloor, l = m - k(n+1) - 1$ and

$$[\mathbf{G}]_{1m} = g(x_k, t_l), \quad r = \left\lfloor \frac{m}{n+1} \right\rfloor, \quad l = m - k(n+1) - 1.$$

For the conditions, we first put $t = 0$, $x = L$, and $x = R$ into Equation (14), respectively. Then, we obtain the following matrix relations by substituting the collocation nodes,

$$\begin{aligned} \mathbf{C}_1 \mathbf{A} &= \mathbf{G}_1, \\ \mathbf{C}_2 \mathbf{A} &= \mathbf{G}_2, \\ \mathbf{C}_3 \mathbf{A} &= \mathbf{G}_3, \end{aligned} \tag{19}$$

where

$$\begin{aligned} [\mathbf{C}_1]_{1,i} &= \mathbf{X}(x_i) \mathbf{D}^T \bar{\mathbf{Y}}(0) \bar{\mathbf{D}}, \\ [\mathbf{C}_2]_{1,i} &= \mathbf{X}(L) \mathbf{D}^T \bar{\mathbf{Y}}(t_i) \bar{\mathbf{D}}, \\ [\mathbf{C}_3]_{1,i} &= \mathbf{X}(R) \mathbf{D}^T \bar{\mathbf{Y}}(t_i) \bar{\mathbf{D}}, \\ [\mathbf{G}_1]_{1,i} &= F(x_i), \\ [\mathbf{G}_2]_{1,i} &= 0, \\ [\mathbf{G}_3]_{1,i} &= b_R(t_i). \end{aligned}$$

Combining $[\mathbf{W}, \mathbf{G}]$ and $[\mathbf{C}_1, \mathbf{G}_1]$, and $[\mathbf{C}_2, \mathbf{G}_2]$ and $[\mathbf{C}_3, \mathbf{G}_3]$, we obtain a new system $[\tilde{\mathbf{W}}, \tilde{\mathbf{G}}]$:

$$[\tilde{\mathbf{W}}, \tilde{\mathbf{G}}] = \begin{pmatrix} \mathbf{W} & , & \mathbf{G} \\ \mathbf{C}_1 & , & \mathbf{G}_1 \\ \mathbf{C}_2 & , & \mathbf{G}_2 \\ \mathbf{C}_3 & , & \mathbf{G}_3 \end{pmatrix}.$$

Applying the Gauss elimination method to the augmented matrix $[\tilde{\mathbf{W}}, \tilde{\mathbf{G}}]$ and deleting the zero rows gives the system $[\bar{\mathbf{W}}, \bar{\mathbf{G}}]$, where $\bar{\mathbf{W}}$ is square matrix and has full rank. Then, the unknown matrix \mathbf{A} can be obtained as

$$\mathbf{A} = \bar{\mathbf{W}}^{-1} \bar{\mathbf{G}}. \tag{20}$$

4. Error Analysis

In this section, first an upper bound for the absolute error is given. Then, we give two error estimation methods that can be applied easily and are useful practically. The first

one is the residual correction procedure. The second one is specifying the consecutive approximations which is similar to the error analysis of the RK4 method.

Theorem 2. Let $f_\alpha p_{n,n}$ and y be the FBSS and the exact solution of Equation (1), respectively. By using the above notations, the absolute error is bounded as follows,

$$|y(x, t) - f_\alpha p_{n,n}(x, t)| \leq |R_n^\alpha(x, t)| + |P_n^\alpha(x, t) - f_\alpha p_{n,n}(x, t)|$$

provided that $D^{k\alpha} f \in C(D)$, where D is a rectangular domain contains the collocation nodes.

Proof. Adding and subtracting the term $P_n^\alpha(x, t)$, the two-dimensional truncated generalized Taylor polynomial defined in (4), into the left-hand side and applying triangle inequality yields the desired result. \square

Let R be the function defined as

$$R(x, t) := \frac{\partial f_\alpha p_{n,n}(x, t)}{\partial t} - d(x) \frac{\partial^\alpha f_\alpha p_{n,n}(x, t)}{\partial x^\alpha}.$$

Adding the term R into both sides of Equation (1) yields the following fractional differential equation for the absolute error

$$\frac{\partial e_{n,n}(x, t)}{\partial t} = d(x) \frac{\partial^\alpha e_{n,n}(x, t)}{\partial x^\alpha} - \frac{\partial f_\alpha p_{n,n}(x, t)}{\partial t} + d(x) \frac{\partial^\alpha f_\alpha p_{n,n}(x, t)}{\partial x^\alpha} + g(x, t), \tag{21}$$

where $e_{n,n} = y - f_\alpha p_{n,n}$. The initial and boundary conditions for the problem are converted to

$$\begin{aligned} e_{n,n}(x, 0) &= 0, \\ e_{n,n}(L, t) &= e_{n,n}(R, t) = 0. \end{aligned} \tag{22}$$

We get an approximate solution, $e_{n,n}^{m,m}$, for the absolute error by applying the method to Equation (21) with the conditions (22) on the nodes $\{(x_i, t_j) : 0 \leq i, j \leq m\} \subset \Omega$. Then, the absolute error $e_{n,n}$ can be estimated by $e_{n,n}^{m,m}$ provided that $\|e_{n,n} - e_{n,n}^{m,m}\| < \varepsilon$ is small.

Corollary 1. Let $f_\alpha p_{n,n}$ be the FBSS. Then, $f_\alpha p_{n,n} + e_{n,n}^{m,m}$ is another approximate solution, corrected FBSS, of Equation (1) and its error function is $e_{n,n} - e_{n,n}^{m,m}$. Moreover, if

$$\|e_{n,n} - e_{n,n}^{m,m}\| < \|y - e_{n,n}\|,$$

then $f_\alpha p_{n,n} + e_{n,n}^{m,m}$ is a better approximation than $p_{n,n}$ in any given norm $\|\cdot\|$.

Let $f_\alpha p_{n,n}$ and $f_\alpha p_{s,s}$ be any two FBSS of Equation (1), y be the exact solution of Equation (1). Then, by using the triangle inequality, we find the following inequality,

$$\|e_{n,n}\| - \|e_{m,m}\| = \|y - f_\alpha p_{n,n}\| - \|y - f_\alpha p_{m,m}\| \leq \|f_\alpha p_{n,n} - f_\alpha p_{m,m}\|.$$

If $\|e_{m,m}\| < \|e_{n,n}\|$, then we can write as follows

$$\|e_{n,n}\| = C \|e_{m,m}\|, \quad C > 1.$$

Therefore, we can bound the error as

$$\begin{aligned} \|e_{n,n}\| - \|e_{m,m}\| &= (C - 1) \|e_{m,m}\| \leq \|f_\alpha p_{m,m} - f_\alpha p_{n,n}\| \text{ or} \\ \|e_{n,n}\| - \|e_{m,m}\| &= \left(1 - \frac{1}{C}\right) \|e_{n,n}\| \leq \|f_\alpha p_{m,m} - f_\alpha p_{n,n}\|. \end{aligned}$$

Therefore, we can bound the error by

$$\frac{1}{(C-1)} \|f_\alpha p_{m,m} - f_\alpha p_{n,n}\| \text{ or } \frac{C}{(C-1)} \|f_\alpha p_{m,m} - f_\alpha p_{n,n}\|. \tag{23}$$

Then, we can bound the error $\|e_{m,m}\|$ well in case of $C \geq 2$, even when the exact solution is not known. A similar argument may be proposed when the error sequence is decreasing (or increasing). Thus, in case of a decreasing (or increasing) error sequence, one of the following bounds is satisfied

$$\|e_{n,n}\| \leq \frac{1}{C-1} \|f_\alpha p_{n+1,n+1} - f_\alpha p_{n,n}\|$$

or

$$\|e_{n,n}\| \leq \frac{C}{C-1} \|f_\alpha p_{n+1,n+1} - f_\alpha p_{n,n}\|.$$

5. Numerical Stability

In this section, we will study the perturbation analysis with stability estimation of the linear systems obtained by the FBSS method, which is similar to that in [40], for a given problem. Perturbing the initial or boundary conditions yields the following perturbed solutions, $f_\alpha p_{n,n}^{per}(x, t)$. There occur two cases:

Case 1: Only the vector $\bar{\mathbf{G}}$ on the right hand side of (20) is perturbed, i.e.,

$$\bar{\mathbf{W}}\mathbf{A} = \bar{\mathbf{G}} + \Delta\bar{\mathbf{G}}. \tag{24}$$

Let us show the perturbed solution of (24) as $\mathbf{A}^p = \mathbf{A} + \Delta\mathbf{A}$, where $\Delta\mathbf{A}$ is the perturbation of the solution resulting from the perturbations in the initial and boundary conditions. Then, the change in the solution caused by the initial change is bounded as [41]

$$\frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|} \leq \text{cond}(\bar{\mathbf{W}}) \frac{\|\Delta\bar{\mathbf{G}}\|}{\|\bar{\mathbf{G}}\|}.$$

Case 2: The perturbations might be occurred both $\bar{\mathbf{W}}$ and $\bar{\mathbf{G}}$,

$$(\bar{\mathbf{W}} + \Delta\bar{\mathbf{W}})\mathbf{A} = \bar{\mathbf{G}} + \Delta\bar{\mathbf{G}}. \tag{25}$$

As the same notation in Case 1, the change in \mathbf{A} caused by perturbing the initials is bounded above as [41]

$$\frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|} \leq \frac{\text{cond}(\bar{\mathbf{W}})}{1 - \text{cond}(\bar{\mathbf{W}}) \frac{\|\Delta\bar{\mathbf{W}}\|}{\|\bar{\mathbf{W}}\|}} \left(\frac{\|\Delta\bar{\mathbf{W}}\|}{\|\bar{\mathbf{W}}\|} + \frac{\|\Delta\bar{\mathbf{G}}\|}{\|\bar{\mathbf{G}}\|} \right).$$

Thus, for Case 1,

$$\begin{aligned} \left| f_\alpha p_{n,n}(x, t) - f_\alpha^{per} p_{n,n}(x, t) \right| &= |f_\alpha \mathbf{B}_n(x) \mathbf{Q}_n(t) (\mathbf{A} - \mathbf{A}^p)| \\ &\leq \|f_\alpha \mathbf{B}_n(x)\| \|\mathbf{Q}_n(t)\| \|\Delta\mathbf{A}\| \\ &\leq \|f_\alpha \mathbf{B}_n(x)\| \|\mathbf{Q}_n(t)\| \text{cond}(\bar{\mathbf{W}}) \frac{\|\Delta\bar{\mathbf{G}}\| \|\mathbf{A}\|}{\|\bar{\mathbf{G}}\|}. \end{aligned} \tag{26}$$

and therefor we can specify the effect of the little changes in the initial and boundary conditions on the FBSS by measuring $\text{cond}(\bar{\mathbf{W}})$. We omit the result for Case 2 by simplicity.

6. Numerical Results and Discussion

In this section, three examples are provided to illustrate the properties and effectiveness of the technique.

6.1. Example 1

Let us consider the FDE [17]

$$\frac{\partial u(x, t)}{\partial t} = d(x) \frac{\partial^{1.8} u(x, t)}{\partial x^{1.8}} + g(x, t), \tag{27}$$

where $0 < x < 1$ and $\alpha = 1.8$, the diffusion coefficient is

$$d(x) = \Gamma(2.2) \frac{x^{2.8}}{6}$$

and the source function

$$g(x, t) = -(1 + x)e^{-t}x^3.$$

The initial condition is

$$u(x, 0) = x^3$$

and the boundary conditions are

$$u(0, t) = 0, \quad u(1, t) = e^{-t}, \quad t > 0.$$

The exact solution to this problem is

$$u(x, t) = e^{-t}x^3.$$

By applying the procedure in Section 3, the matrix equation for Equation (27) is found as

$$[\mathbf{X}(x)\mathbf{D}^T\tilde{\mathbf{Y}}(t)\tilde{\mathbf{B}}\tilde{\mathbf{D}} - \Gamma(2.2)\frac{x^{2.8}}{6}\mathbf{X}(x)\mathbf{C}(\mathbf{ff})\mathbf{D}^T\tilde{\mathbf{Y}}(t)\tilde{\mathbf{D}}]\mathbf{A} = -(1 + x)e^{-t}x^3. \tag{28}$$

The collocation points that will be used are the Chebyshev interpolation nodes

$$\{(x_i, y_j) : 0 \leq i, j \leq n, \quad x_i = \frac{1}{2} + \frac{1}{2} \cos\left(\frac{2i - 1}{2n}\right)\pi, \quad y_j = \frac{1}{2} + \frac{1}{2} \cos\left(\frac{2j - 1}{2n}\right)\pi\}.$$

By substituting the collocation points in Equation (28), we will obtain \mathbf{W} matrix. The conditions matrices for $(x, 0) = x^3, u(0, t) = 0, u(1, t) = e^{-t}$ are obtained as

$$\begin{aligned} p_{n,n}(x_i, 0) &= \mathbf{X}(x_i)\mathbf{D}^T\tilde{\mathbf{Y}}(0)\tilde{\mathbf{D}}\mathbf{A} = x_i^3, \\ p_{n,n}(0, t_j) &= \mathbf{X}(0)\mathbf{D}^T\tilde{\mathbf{Y}}(t_j)\tilde{\mathbf{D}}\mathbf{A} = 0, \\ p_{n,n}(1, t_j) &= \mathbf{X}(1)\mathbf{D}^T\tilde{\mathbf{Y}}(t_j)\tilde{\mathbf{D}}\mathbf{A} = e^{-t_j}, \quad 0 \leq i, j, k \leq n. \end{aligned}$$

Then, $[\tilde{\mathbf{W}}, \tilde{\mathbf{G}}]$ is obtained by combining these matrices. Thus, we obtain the coefficient matrix \mathbf{A} . As we perform the method for different n values, we obtain n different FBSS. All calculations are done using the Maple program. The results for $n = 5, 10, n = 15,$ and $n = 20$ are given in Table 1. The absolute errors are graphed in Figure 1. Table 2 show the absolute errors between consecutive approximations. The exact and approximate solutions with $n = 15$ are graphed in Figure 2. The values of error functions and the exact solution, at time $t = 1$, are given in Table 3 and graphed in Figure 3. The upper bounds of absolute error at time $t = 1$ are given in Figure 4. A comparison of the method with the numerical methods in [16,18] is made for $t = 1$. The absolute error for $n = 5$ with the errors obtained

by the corrected FBSSs are given in Table 4. We can say that increasing n yields better approximation results and the method provides better results. From Table 2, we deduce that the absolute error $e_{n,n}$ can be estimated approximately by the difference between $e_{n,n}$ and $e_{n+1,n+1}$. From Table 3, we can see the method gives more accurate results than the methods in [16,18] for $t = 1$. From Table 4, increasing m yields more accurate corrected approximate solutions. Table 5 shows the error norms (L_2, L_∞) resulting with CPU time (in seconds) used in the Maple program to find the numerical solutions for different n values. We provide the stability results for the method for some n values in Table 6. Increasing n gives the condition numbers which are increasing. It can be said that the approximate solutions will be more stable around $n = 10$ by using both the theoretical upper bounds and the numerical results. The method produces approximately 10^5 as a condition number. For $n \gg 10$, the solutions will be more sensitive to small variations since the number of conditions is large. Thus, around $n = 10$, we can say that the method works well for this problem.

Table 1. Comparisons of the absolute errors for different n for Example 1.

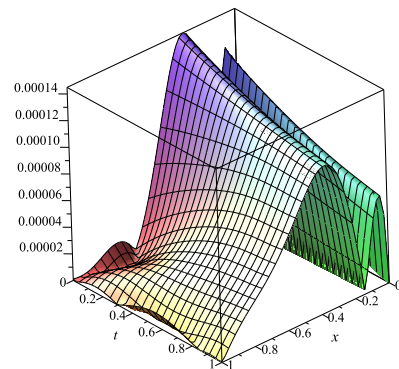
x	t	$ e_{5,5} $	$ e_{10,10} $	$ e_{15,15} $	$ e_{20,20} $
0.0	0.0	8.096×10^{-5}	1.030×10^{-6}	9.528×10^{-8}	1.678×10^{-8}
0.2	0.2	7.314×10^{-5}	2.828×10^{-7}	9.191×10^{-9}	6.885×10^{-9}
0.4	0.4	1.148×10^{-4}	3.347×10^{-7}	2.655×10^{-8}	7.076×10^{-10}
0.6	0.6	5.966×10^{-5}	9.071×10^{-7}	2.046×10^{-9}	5.669×10^{-9}
0.8	0.8	3.015×10^{-5}	8.664×10^{-8}	1.116×10^{-8}	9.309×10^{-8}
1.0	1.0	7.439×10^{-7}	1.744×10^{-10}	1.789×10^{-10}	5.641×10^{-11}

Table 2. Upper bounds for the absolute errors for $n = 9$ and $n = 10$ of Example 1.

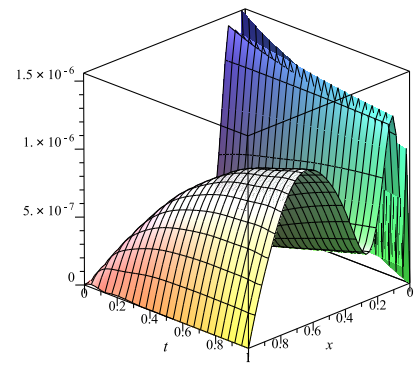
x	t	$ y - y_9 $	$ y - y_{10} $	$ y_9 - y_{10} $
0.0	0.0	2.138×10^{-6}	1.030×10^{-6}	1.107×10^{-6}
0.2	0.2	6.193×10^{-7}	2.828×10^{-7}	9.022×10^{-6}
0.4	0.4	2.488×10^{-7}	3.347×10^{-7}	2.153×10^{-6}
0.6	0.6	2.278×10^{-6}	9.071×10^{-7}	1.371×10^{-6}
0.8	0.8	1.527×10^{-6}	8.664×10^{-8}	6.661×10^{-6}
1.0	1.0	2.344×10^{-9}	1.744×10^{-10}	2.519×10^{-9}

Table 3. Absolute errors on $[0, 1]$, with $n = 5, n = 10$ and $n = 15$ for Example 1 at time $t = 1$.

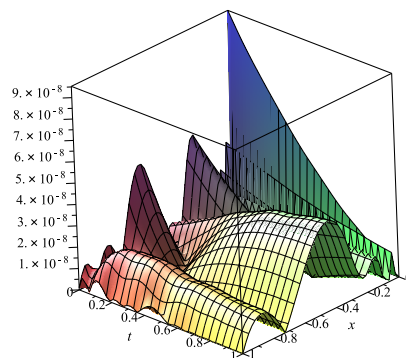
x	$n = 5$	$n = 10$	$n = 15$	Method in [16]	Method in [18]
0.1	7.883×10^{-7}	2.705×10^{-9}	1.552×10^{-10}	2.7×10^{-8}	1.9×10^{-6}
0.2	4.005×10^{-5}	5.618×10^{-7}	2.877×10^{-9}	6.5×10^{-7}	1.4×10^{-7}
0.3	1.017×10^{-4}	3.809×10^{-7}	2.154×10^{-8}	5.1×10^{-7}	2.6×10^{-6}
0.4	1.080×10^{-4}	6.936×10^{-7}	4.0434×10^{-8}	3.2×10^{-6}	3.8×10^{-6}
0.5	9.053×10^{-5}	8.913×10^{-7}	3.881×10^{-8}	5.8×10^{-6}	3.4×10^{-6}
0.6	6.758×10^{-5}	8.504×10^{-7}	2.398×10^{-8}	6.1×10^{-6}	2.0×10^{-6}
0.7	4.838×10^{-5}	4.054×10^{-7}	5.763×10^{-9}	3.1×10^{-6}	4.5×10^{-7}
0.8	3.355×10^{-5}	7.251×10^{-7}	1.025×10^{-8}	2.2×10^{-6}	3.3×10^{-7}
0.9	1.872×10^{-6}	2.869×10^{-7}	1.335×10^{-9}	6.1×10^{-6}	9.7×10^{-8}



Absolute error of numerical solutions of Example 1, for $n=5$.



Absolute error of numerical solutions of Example 1, for $n=10$



Absolute error of numerical solutions of Example 1, for $n=15$

Figure 1. The absolute errors for Example 1: $n = 5, n = 10$ and $n = 15$.

Table 4. The absolute errors with the corrected FBSSs $n = 5$, and $m = 10, 15, 20$ of Example 1.

x	t	$e_{5,5}$	$e_{5,5}^{10,10}$	$e_{5,5}^{15,15}$	$e_{5,5}^{20,20}$
0.0	0.0	8.096×10^{-5}	1.825×10^{-6}	2.621×10^{-7}	4.519×10^{-8}
0.2	0.2	7.314×10^{-5}	8.021×10^{-5}	7.156×10^{-5}	7.191×10^{-6}
0.4	0.4	1.148×10^{-4}	1.084×10^{-4}	1.029×10^{-4}	9.817×10^{-5}
0.6	0.6	5.966×10^{-5}	5.071×10^{-5}	3.681×10^{-5}	2.295×10^{-6}
0.8	0.8	3.015×10^{-5}	2.947×10^{-5}	8.609×10^{-6}	4.658×10^{-6}
1.0	1.0	7.439×10^{-7}	7.224×10^{-7}	1.158×10^{-7}	7.392×10^{-7}

Table 5. Error norms L_2, L_∞ and CPU for different values of n for Example 1.

n	L_2	L_∞	CPU
3	4.403×10^{-2}	2.343×10^{-2}	6.30
5	2.454×10^{-4}	1.148×10^{-4}	10.5
8	4.328×10^{-6}	8.541×10^{-6}	15.5
10	2.582×10^{-6}	1.472×10^{-6}	18.3
13	4.079×10^{-7}	2.186×10^{-7}	90.5
15	1.074×10^{-7}	9.528×10^{-8}	129.7
18	1.181×10^{-7}	7.121×10^{-8}	472.1
20	1.061×10^{-7}	7.067×10^{-8}	750.5

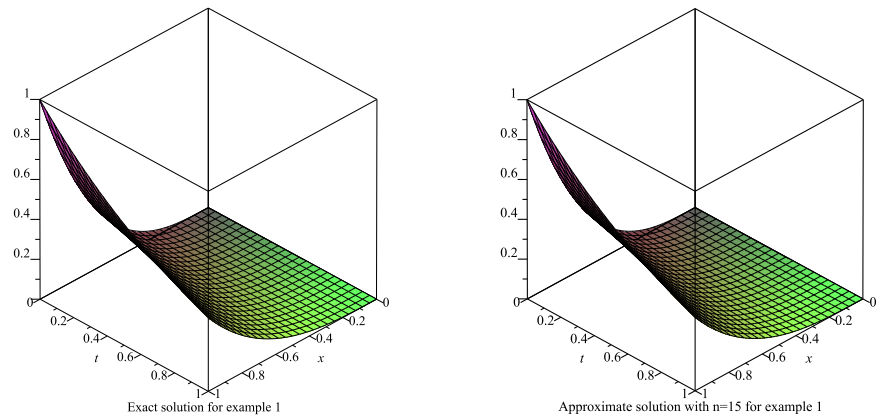


Figure 2. The exact solution and FBSSs for Example 1.

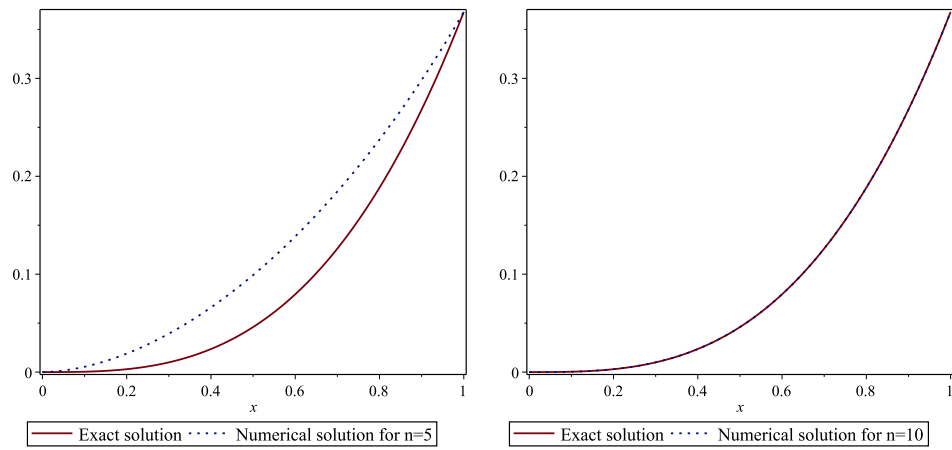


Figure 3. FBSSs and the exact solution for Example 1, with $n = 5$ and $n = 10$ at time $t = 1$.

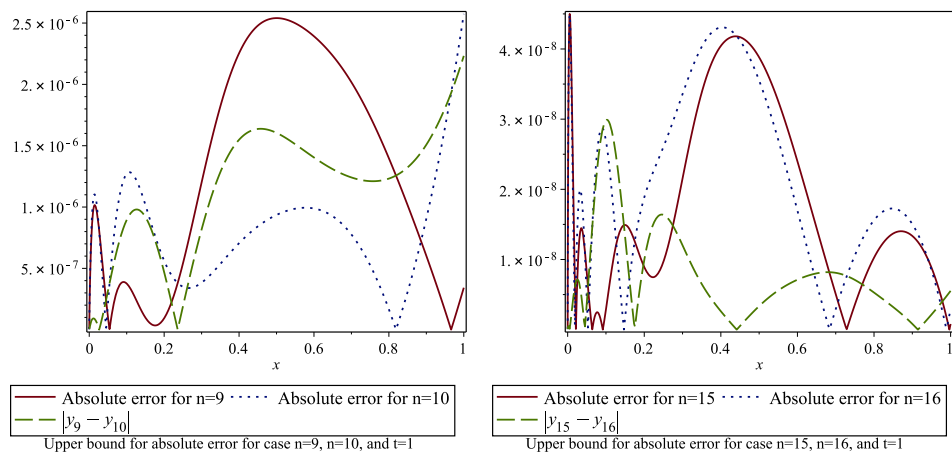


Figure 4. Upper bounds of the absolute errors for Example 1, with $n = 9, 10$ and $n = 15, 16$ at time $t = 1$.

Table 6. Stability of the system related to the method for different values of n for Example 1.

	$n = 5$	$n = 10$	$n = 15$
$\text{cond}(\bar{W})$	673.48	6.33×10^5	1.68×10^9
$\ \Delta A\ $	1.0253×10^{-16}	1.5394×10^{-15}	4.9009×10^{-9}
$\ A\ $	1.0000	1.0000	0.9999
$\ \Delta \bar{G}\ $	10^{-16}	10^{-16}	10^{-16}
$\ \bar{G}\ $	1.0775	2.1642	4.1801
Upper Bound obtained by (26)	2×10^{-10}	5.9987×10^{-7}	3.9567×10^{-2}

6.2. Example 2

Let us consider the FDE [17]

$$\frac{\partial u(x, t)}{\partial t} = d(x) \frac{\partial^{1.4} u(x, t)}{\partial x^{1.4}} + g(x, t), \tag{29}$$

where $0 < x < 1$ and $\alpha = 1.4$. The diffusion coefficient is

$$d(x) = \frac{1}{24} \Gamma(5 - 1.4) x^{1.4},$$

the source function

$$g(x, t) = -2e^{-t} x^4.$$

The initial and boundary conditions are as follows, respectively,

$$u(x, 0) = x^4$$

$$u(0, t) = 0, \quad u(1, t) = e^{-t}, \quad t > 0.$$

The exact solution of this problem is given by

$$u(x, t) = e^{-t} x^4.$$

We perform the method for $n = 7, 10$ and $n = 15$. The results are given in Table 7. The absolute errors are graphed in Figure 5. A comparison of the method with the numerical method in [17] is given in Table 7. We can say that increasing n yields better approximation results and the method provides better results. The method gives more accurate results than the method in [17]. From Table 8, we conclude that the absolute error $e_{n,n}$ can be bounded approximately by $|e_{n,n} - e_{n+1,n+1}|$. The absolute error for $n = 5$ with the errors obtained by the corrected FBSSs are given in Table 9. The absolute errors for $n = 11$ and $n = 16$ at time $t = 1$ are given in Figure 6. The errors with their upper bounds at $t = 1$ are given in Figure 7. Table 10 shows the error norms (L_2, L_∞), resulting with CPU time (in seconds) used in the Maple program to find FBSSs for different n values. The method again gives approximately 10^5 as a condition number. Therefore, around $n = 10$, we can say from Table 11 that the method produces more stable approximations for this problem.

6.3. Example 3

Let us consider the fractional convection diffusion equation [42]

$$\frac{\partial u(x, t)}{\partial t} + d(x) \frac{\partial u(x, t)}{\partial x} = c(x) \frac{\partial^2 u(x, t)}{\partial x^2} + g(x, t), \tag{30}$$

where $0 < x < 1$ and. The diffusion coefficient is

$$d(x) = \frac{x}{3}, \quad c(x) = \frac{x^2}{6}.$$

the source function

$$g(x, t) = x^3 \cosh(t).$$

The initial and boundary conditions are as follows, respectively,

$$u(x, 0) = 0$$

$$u(0, t) = 0, \quad u(1, t) = \sinh(t).$$

The exact solution of this problem is given by

$$u(x, t) = x^3 \sinh(t).$$

Table 7. The absolute errors for different n values with a comparison with the method in [17] for Example 2.

x	t	$n = 8$	$n = 10$	$n = 15$	Method in [17]
0.1	0.1	1.421×10^{-7}	7.187×10^{-9}	1.343×10^{-13}	2.7×10^{-6}
0.2	0.2	7.425×10^{-8}	8.404×10^{-9}	6.504×10^{-11}	1.1×10^{-5}
0.3	0.3	3.189×10^{-8}	24.59×10^{-8}	2.747×10^{-10}	1.2×10^{-5}
0.4	0.4	2.147×10^{-7}	1.310×10^{-8}	3.070×10^{-10}	1.2×10^{-5}
0.5	0.5	2.152×10^{-7}	1.253×10^{-8}	5.414×10^{-10}	1.5×10^{-5}
0.6	0.6	3.730×10^{-8}	1.076×10^{-8}	6.843×10^{-10}	1.8×10^{-5}
0.7	0.7	3.411×10^{-7}	1.787×10^{-8}	7.885×10^{-10}	1.8×10^{-5}
0.8	0.8	2.716×10^{-7}	3.792×10^{-8}	6.409×10^{-10}	1.3×10^{-5}
0.9	0.9	1.262×10^{-7}	1.125×10^{-10}	4.291×10^{-12}	6.6×10^{-6}

Table 8. Upper bounds of the absolute errors, for $n = 10, n = 11$ and Example 2.

x	t	$ y - y_{10} $	$ y - y_{11} $	$ y_{10} - y_{11} $
0.0	0.0	3.201×10^{-9}	9.861×10^{-10}	7.911×10^{-10}
0.2	0.2	8.404×10^{-9}	4.976×10^{-9}	2.924×10^{-8}
0.4	0.4	1.310×10^{-8}	2.389×10^{-8}	2.576×10^{-8}
0.6	0.6	1.076×10^{-8}	3.348×10^{-8}	1.682×10^{-7}
0.8	0.8	3.792×10^{-8}	4.166×10^{-8}	1.076×10^{-6}
1.0	1.0	9.600×10^{-10}	7.996×10^{-11}	2.110×10^{-6}

Table 9. The absolute errors with the corrected FBSSs for $n = 5$ and $m = 10, 15, 20$ of Example 2.

x	t	$e_{5,5}$	$e_{5,5}^{10,10}$	$e_{5,5}^{15,15}$	$e_{5,5}^{20,20}$
0.0	0.0	3.401×10^{-3}	5.042×10^{-7}	3.603×10^{-7}	6.510×10^{-8}
0.2	0.2	5.411×10^{-4}	2.017×10^{-4}	2.174×10^{-4}	2.009×10^{-5}
0.4	0.4	6.232×10^{-4}	1.646×10^{-5}	1.600×10^{-5}	2.641×10^{-5}
0.6	0.6	1.356×10^{-3}	2.266×10^{-5}	3.024×10^{-5}	4.091×10^{-5}
0.8	0.8	1.201×10^{-3}	2.752×10^{-5}	3.847×10^{-5}	6.655×10^{-5}
1.0	1.0	1.413×10^{-5}	1.143×10^{-6}	2.427×10^{-6}	4.372×10^{-7}

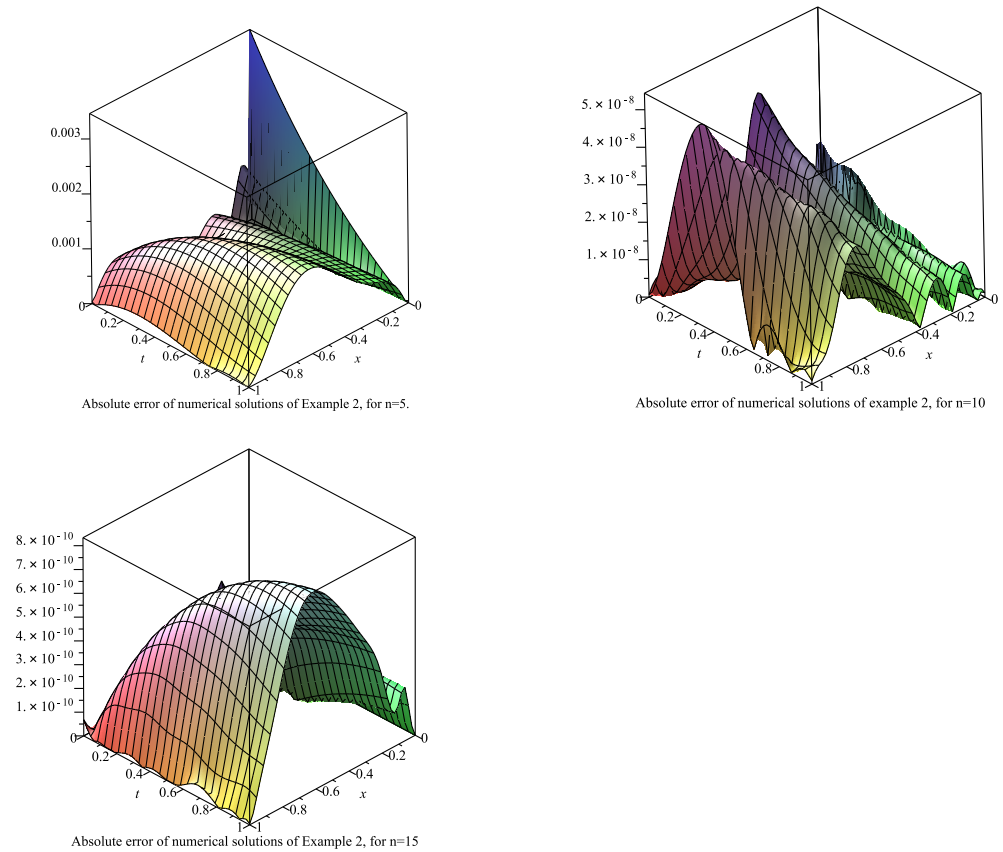


Figure 5. The absolute errors for Example 2, with $n = 5$, $n = 10$ and $n = 15$.

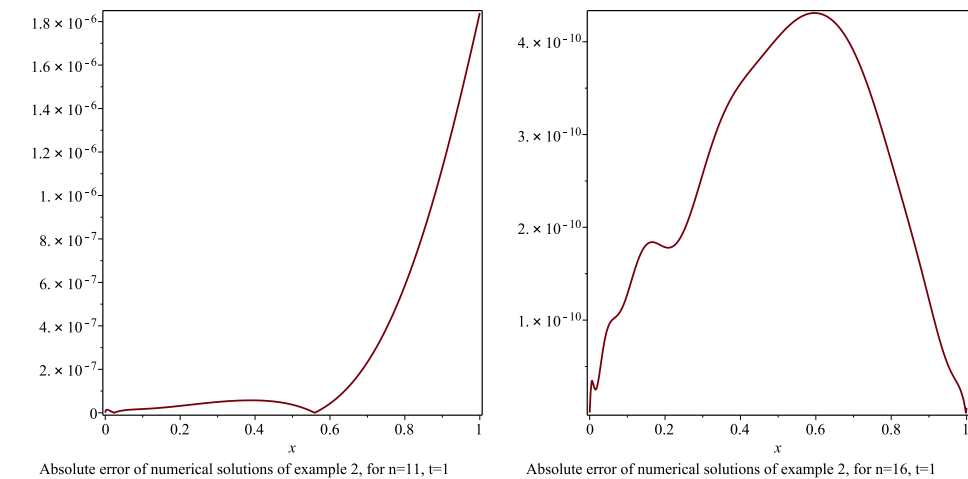


Figure 6. The absolute errors for Example 2, with $n = 11$, $n = 16$, and $t = 1$.

By applying the technique in Section 3, for different values of n , the FBSSs for the problem are founded. From Tables 12 and 13, we can see obviously that the results obtained by FBSS and the exact solution of the problem are agreement with each other. We compare our results with [42], can see our results are more accurate. The condition numbers for the systems that related to the method for various n are given in Table 14. We can say from Table 14 that the method is insensitive to small variations for $n = 10$.

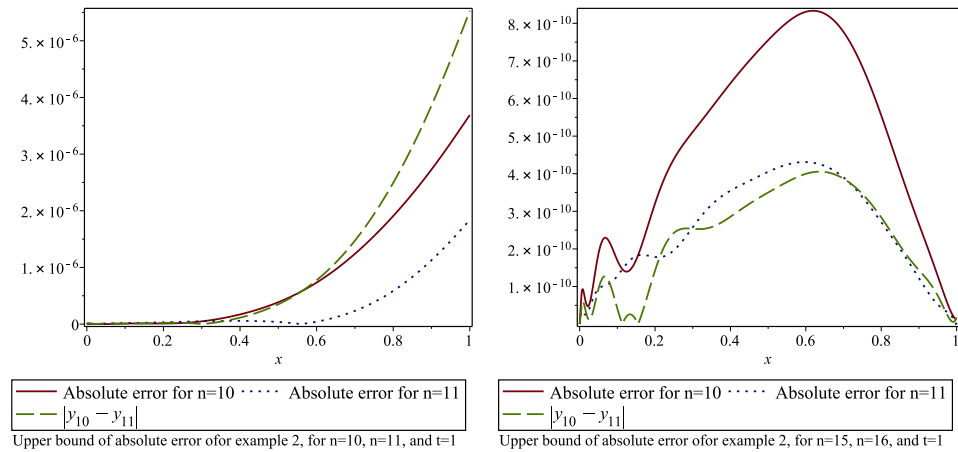


Figure 7. Upper bounds of absolute errors for Example 2, with $n = 10, n = 11,$ and $n = 15, n = 16,$ for case $t = 1.$

Table 10. Error norms L_2, L_∞ and CPU for different values of n for Example 2.

n	L_2	L_∞	CPU
3	2.294×10^{-1}	1.255×10^{-1}	4.5
5	4.544×10^{-3}	3.461×10^{-3}	7.19
8	5.733×10^{-7}	3.411×10^{-7}	9.5
10	5.538×10^{-8}	3.792×10^{-8}	19.6
13	5.249×10^{-9}	2.287×10^{-9}	62.0
15	1.413×10^{-9}	7.885×10^{-10}	162.3
18	2.916×10^{-10}	1.497×10^{-10}	445
20	8.245×10^{-11}	4.117×10^{-11}	700

Table 11. $\text{cond}(\bar{W})$ for different values of n for Example 2.

	$n = 3$	$n = 5$	$n = 8$	$n = 10$	$n = 13$	$n = 15$
$\text{cond}(\bar{W})$	105.67	851.57	72.61×10^3	8.99×10^5	5.03×10^7	9.18×10^8

Table 12. Comparisons of the absolute errors for $n = 3, 5, 7$ of Example 3.

x	[42]	Our Method	[42]	Our Method	[42]	Our Method
	$n = 3$	$n = 3$	$n = 5$	$n = 5$	$n = 7$	$n = 7$
0.1	1.63×10^{-5}	2.05×10^{-6}	3.06×10^{-6}	1.20×10^{-7}	2.68×10^{-6}	6.02×10^{-11}
0.2	4.85×10^{-5}	5.21×10^{-6}	2.08×10^{-6}	1.65×10^{-7}	1.42×10^{-6}	1.50×10^{-10}
0.3	5.24×10^{-5}	2.70×10^{-6}	2.60×10^{-7}	3.40×10^{-7}	2.06×10^{-6}	1.23×10^{-10}
0.4	3.67×10^{-5}	9.91×10^{-5}	9.20×10^{-7}	4.09×10^{-7}	4.05×10^{-7}	6.05×10^{-10}
0.5	1.04×10^{-5}	2.53×10^{-4}	9.70×10^{-7}	6.17×10^{-7}	6.90×10^{-7}	1.49×10^{-10}
0.6	1.76×10^{-5}	5.04×10^{-4}	1.10×10^{-7}	1.29×10^{-7}	6.00×10^{-7}	2.28×10^{-9}
0.7	3.85×10^{-5}	8.35×10^{-4}	1.10×10^{-6}	2.60×10^{-6}	7.00×10^{-8}	3.13×10^{-9}
0.8	4.33×10^{-5}	1.20×10^{-4}	1.92×10^{-6}	4.09×10^{-6}	3.90×10^{-7}	5.34×10^{-9}
0.9	2.33×10^{-5}	2.19×10^{-4}	2.14×10^{-6}	4.03×10^{-6}	1.84×10^{-6}	8.06×10^{-9}

Table 13. Absolute errors for different values of n of Example 3.

x	$n = 10$	$n = 13$	$n = 15$
0.1	1.92×10^{-15}	5.65×10^{-18}	3.13×10^{-17}
0.2	1.93×10^{-15}	9.15×10^{-18}	7.82×10^{-18}
0.3	2.12×10^{-15}	2.14×10^{-17}	4.31×10^{-17}
0.4	2.64×10^{-14}	6.28×10^{-17}	1.81×10^{-17}
0.5	6.36×10^{-14}	1.26×10^{-17}	1.49×10^{-16}
0.6	9.33×10^{-14}	1.34×10^{-17}	3.22×10^{-16}
0.7	1.65×10^{-13}	4.82×10^{-17}	2.47×10^{-16}
0.8	2.35×10^{-13}	8.36×10^{-17}	1.78×10^{-16}
0.9	3.52×10^{-13}	4.98×10^{-17}	6.67×10^{-16}

Table 14. $\text{cond}(\bar{W})$ for different values of n for Example 3.

	$n = 3$	$n = 5$	$n = 7$	$n = 10$	$n = 13$	$n = 15$
$\text{cond}(\bar{W})$	47.24	308.70	2775.84	63340.64	3.05×10^6	2.92×10^7

7. Conclusions

In this paper, we proposed the FBSS to solve the FDE numerically. The method can be easily implemented and is effective. It comprises fractional Bernstein polynomials and the collocation method. First, the fundamental matrix equation is obtained and then it is solved by the Gauss elimination procedure. Applying the initial and boundary conditions, we get the solution for the given n value. Two error estimations, residual correction procedure and estimations are obtained by the difference of the consecutive approximations. We applied the method to some examples. Generally, the results demonstrate that the proposed method achieves better approximation accuracy than some other well-known methods. From the examples, we can bound approximately the absolute error $e_{n,n}$ by $|e_{n,n} - e_{n+1,n+1}|$. On the other hand, one can obtain more accurate results by using the residual correction procedure. Increasing m gives more accurate corrected FBSS. For the stability of the method, we can decide which size n yields more stable approximations by specifying at the number of conditions of the related system. For the examples, the method produces condition numbers approximately as 10^5 for $n \approx 10$ approximately. Thus, around $n = 10$, we can say that the method is suitable for these examples. The subjects of our future works can be exemplified by applying fractional Bernstein series solution for solving fractional integral differential equations and chaotic fractional order systems.

Author Contributions: Conceptualization, M.H.A. and O.I.; methodology, M.H.A. and O.I.; software, M.H.A. and O.I.; validation, I.H. and O.I.; writing—original draft preparation, M.H.A. and O.I.; writing—review and editing, I.H.; supervision, I.H.; funding acquisition, M.H.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Abu Dhabi University grant number 18-51-45005.

Acknowledgments: The authors acknowledge with thanks, the Department of Mathematics and Statistics at the Abu Dhabi University for making their facilities available for the research.

Conflicts of Interest: The authors declare that there is no conflict of interest.

References

1. Barkai, E.; Metzler, R.; Klafter, J. From continuous time random walks to the fractional Fokker-Planck equation. *Phys. Rev. E* **2000**, *61*, 132–138. [[CrossRef](#)] [[PubMed](#)]
2. Blumen, A.; Zumofen, G.; Klafter, J. Transport aspects in anomalous diffusion: Lévy walks. *Phys. Rev. A* **1989**, *40*, 3964–3973. [[CrossRef](#)] [[PubMed](#)]
3. Salehi, F.; Saeedi, H.; Moghadam, M.M. A Hahn computational operational method for variable order fractional mobile-immobile advection-dispersion equation. *Math. Sci.* **2018**, *12*, 91–101. [[CrossRef](#)]

4. Cardoso, L.C.; Dos Santos, F.L.P.; Camargo, R.F. Analysis of fractional-order models for hepatitis B. *Comput. Appl. Math.* **2018**, *37*, 4570–4586. [[CrossRef](#)]
5. Baeumer, B.; Benson, D.A.; Meerschaert, M.M.; Wheatcraft, S.W. Subordinated advection-dispersion equation for contaminant transport. *Water Resour. Res.* **2001**, *37*, 1543–1550. [[CrossRef](#)]
6. Magin, R.L. *Fractional Calculus in Bioengineering*; Begell House Publishers: Redding, CA, USA, 2006.
7. Tarasov, V.E. *Fractional Dynamics: Applications of Fractional Calculus to Dynamics of Particles, Fields and Media*; Springer: Berlin/Heidelberg, Germany, 2010.
8. Mainardi, F. *Fractional Calculus and Waves in Linear Viscoelasticity: An Introduction to Mathematical Models*; Imperial College Press: London, UK, 2010.
9. Povstenko, Y. *Linear Fractional Diffusion-Wave Equation for Scientists and Engineers*; Birkhäuser: New York, NY, USA, 2015.
10. Sousa, E. Finite difference approximations for a fractional advection diffusion problem. *J. Comput. Phys.* **2009**, *228*, 4038–4054. [[CrossRef](#)]
11. Miller, K.S.; Ross, B. *An Introduction to the Fractional Calculus and Fractional Differential Equations*; Wiley: Hoboken, NJ, USA, 1993.
12. Tadjeran, C.; Meerschaert, M.M.; Scheffler, H.P. A second-order accurate numerical approximation for the fractional diffusion equation. *J. Comput. Phys.* **2006**, *213*, 205–213. [[CrossRef](#)]
13. Meerschaert, M.M.; Tadjeran, C. Finite difference approximations for fractional advection-dispersion flow equations. *J. Comput. Appl. Math.* **2004**, *172*, 65–77. [[CrossRef](#)]
14. Fix, G.J.; Roof, J.P. Least squares finite-element solution of a fractional order two-point boundary value problem. *Comput. Math. Appl.* **2004**, *48*, 1017–1033. [[CrossRef](#)]
15. Cetinkaya, A.; Kiyamaz, O. The solution of the time-fractional diffusion equation by the generalized differential transform method. *Math. Comput. Model.* **2013**, *57*, 2349–2354. [[CrossRef](#)]
16. Baseri, A.; Babolian, E.; Abbasbandy, S. Normalized Bernstein polynomials in solving space-time fractional diffusion equation. *Adv. Differ. Equ.* **2017**, *2017*, 346. [[CrossRef](#)]
17. Mohammed, O.H.; Fadhel, S.F.; Al-Safi, M.G.S. Shifted Jacobi Tau Method for Solving the Space Fractional Diffusion Equation. *IOSR-JM* **2014**, *10*, 34–44. [[CrossRef](#)]
18. Doha, E.H.; Bhrawy, A.H.; Ezz-Eldien, S.S. Numerical approximations for fractional diffusion equations via a Chebyshev spectral-tau method. *Cent. Eur. J. Phys.* **2013**, *11*, 1494–1503. [[CrossRef](#)]
19. Avci, I.; Mahmudov, N.I. Numerical Solutions for Multi-Term Fractional Order Differential Equations with Fractional Taylor Operational Matrix of Fractional Integration. *Mathematic* **2020**, *8*, 96.
20. Pandey, R.K.; Kumar, N. Solution of Lane-Emden type equations using Bernstein operational matrix of differentiation. *New Astron.* **2012**, *17*, 303–308. [[CrossRef](#)]
21. Yousefi, S.A.; Behroozifar, M. Operational matrices of Bernstein polynomials and their applications. *Int. J. Syst. Sci.* **2010**, *41*, 709–716. [[CrossRef](#)]
22. Yuzbas, S. Numerical solutions of fractional Riccati type differential equations by means of the Bernstein polynomials. *Appl. Math. Comput.* **2013**, *219*, 6328–6343.
23. Isik, O.R.; Sezer, M.; Guney, Z. A rational approximation based on Bernstein polynomials for high order initial and boundary values problems. *Appl. Math. Comput.* **2011**, *217*, 9438–9450.
24. Isik, O.R.; Sezer, M.; Guney, Z. Bernstein series solution of linear second-order partial differential equations with mixed conditions. *Math. Methods Appl. Sci.* **2014**, *37*, 609–619. [[CrossRef](#)]
25. Lorentz, G.G. *Bernstein Polynomials*; Chelsea Publishing Company: New York, NY, USA, 1986.
26. Büyükyazici, I.; Ibikli, E. The approximation properties of generalized bernstein polynomials of two variables. *Appl. Math. Comp.* **2004**, *156*, 367–380. [[CrossRef](#)]
27. Khataybeh, S.N.; Hashim, I.; Alshbool, M. Solving directly third-order ODEs using operational matrices of Bernstein polynomials method with applications to fluid flow equations. *J. King Saud Univ. Sci.* **2019**, *31*, 822–826. [[CrossRef](#)]
28. Alshbool, M.H.T.; Hashim, I. Multistage Bernstein polynomials for the solutions of the fractional order stiff systems. *J. King Saud Univ. Sci.* **2016**, *28*, 280–285. [[CrossRef](#)]
29. Alshbool, M.H.T.; Bataineh, A.S.; Hashim, I.; Isik, O.R. Solution of fractional-order differential equations based on the operational matrices of new fractional Bernstein functions. *J. King Saud Univ. Sci.* **2017**, *29*, 1–18. [[CrossRef](#)]
30. Alshbool, M.H.T.; Bataineh, A.S.; Hashim, I.; Isik, O.R. Approximate solutions of singular differential equations with estimation error by using Bernstein polynomials. *Int. J. Pure Appl. Math.* **2015**, *100*, 109–125. [[CrossRef](#)]
31. Bhatti, M.I.; Bracken, P. Solutions of differential equations in a Bernstein polynomial basis. *J. Comput. Appl. Math.* **2007**, *205*, 272–280. [[CrossRef](#)]
32. Alshbool, M.; Hashim, I. Bernstein polynomials for solving nonlinear stiff system of ordinary differential equations. In *AIP Conference Proceedings*; AIP Publishing LLC: New York, NY, USA, 2015; Volume 1678, p. 060015.
33. Bencheikh, A.; Chiter, L.; Abbassi, H. Bernstein polynomials method for numerical solutions of integro-differential form of the singular Emden-Fowler initial value problems. *Int. J. Math. Comput. Sci.* **2018**, *17*, 66–75. [[CrossRef](#)]
34. Alshbool, M.H.T.; Shatanawi, W.; Hashim, I.; Sarr, M. Residual Correction Procedure with Bernstein Polynomials for Solving Important Systems of Ordinary Differential Equations. *Cent. Eur. J. Phys.* **2013**, *11*, 1494–1503.

35. Xiao, D. Error estimation of the parametric non-intrusive reduced order model using machine learning. *Comput. Meth. Appl. Mech. Eng.* **2019**, *355*, 513–534. [[CrossRef](#)]
36. Kiryakova, V. *Generalized Fractional Calculus and Applications*; Longman Scientific & Technical: Harlow, UK, 1994.
37. Podlubny, I. *Fractional Differential Equations*; Academic Press: San Diego, CA, USA, 1999.
38. Kilbas, A.A.; Srivastava, H.M.; Trujillo, J.J. *Theory and Applications of Fractional Differential Equations*; Elsevier: Amsterdam, The Netherlands, 2006.
39. Cheng, J. On Multivariate Fractional Taylor's and Cauchy' Mean Value Theorem. *J. Math. Study* **2019**, *52*, 38–52. [[CrossRef](#)]
40. Hu, H.Y.; Chen, J.S.; Chi, S.W. Perturbation and stability analysis of strong form collocation with reproducing kernel approximation. *Int. J. Numer Methods Eng.* **2011**, *88*, 157–179. [[CrossRef](#)]
41. Atkinson, K.E. *An Introduction to Numerical Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2008.
42. Xie, J.; Huang, Q.; Yang, X. Numerical solution of the one-dimensional fractional convection diffusion equations based on Chebyshev operational matrix. *Springerplus* **2016**, *5*, 1–16. [[CrossRef](#)] [[PubMed](#)]