Egy. J. Pure & Appl. Sci. 2024; 62(2):44-51

Egyptian Journal of Pure and Applied Science

Integrating Bernstein and Improved Block-Pulse Functions for Solving Linear Fredholm Integro-Differential Equations

Mohamed A. Ramadan [1*], Heba S. Osheba [1]

[1] *Mathematics and Computer Science Department, Faculty of Science, Menoufia University, Egypt.*

ARTICLE INFO

Received 4 February 2024
Accepted 2 March 2024

Keywords
Integral equations,
Approximation of functions,
Orthogonal polynomials,
Accuracy.

Correspondence
Mohamed A. Ramadan

E-mail*
(Corresponding Author)
ramadanmohamed13@yahoo.com

ABSTRACT

Mathematical modeling of real-life problems usually results in some form of functional equations, e.g. algebraic equations, differential equations, integral equations and others. The occurrence of differential equations and integral equations is common in many areas of the sciences and engineering. In particular, the conversion of boundary value problems in differential equations to integro-differential equations, with limits of integration, considered as constant, is termed Fredholm integro-differential equations In this study, issues involving linear Fredholm integro-differential equations are numerically solved using a hybrid of orthogonal functions. To solve these problems, a hybrid method combining improved block-pulse functions and Bernstein is proposed. This blended method is test by the authors in a previous work is of good agreement with the exact solution. To convert the solution of integro-differential equations to the solution of algebraic equations, the operational matrices of derivative for this function, together with the hybrid functions, are presented. To demonstrate the practicality and accuracy of the proposed approach in this study, we provide some test problems Examples are given to highlight the accuracy and effectiveness of the proposed method.

1. Introduction

There are many uses for integro-differential equations, including modelling of spatiotemporal processes in the natural sciences, electrostatics, and control theory of industrial mathematics, engineering, and mathematics, as well as modelling of epidemics [1]. Integral equations do not have analytical solutions, or they are difficult to find. This has led to the development of numerous numerical techniques for solving integral equations.

As a result; we employ a variety of numerical techniques to roughly solve these equations. The differential transforms method [2], the Haar wavelets method [3-5], the hybrid Legendre polynomials and block-pulse functions approach [6, 7], the triangular functions [8], the single-term Walsh series method [9], the wavelet-Galerkin method [10], and the compact finite difference method [11] are a few of these methods.

Y. H. Youssri, A. G. Atta [12] proposed novel spectral algorithm utilizing Fibonacci polynomials to numerically solve both linear and nonlinear integro-differential equations with fractional-order derivatives. A. G. Atta and Y. H. Youssriin [13] introduced new spectral collocation approach is applied to obtain precise numerical approximation using new basis functions based on shifted first-kind Chebyshev polynomials for solving the nonlinear time-fractional partial integro-differential equation with a weakly singular kernel. Moreover, Y. H. Youssri and R. M. Hafez reported a collocation algorithm for the numerical solution of a Volterra–Fredholm integral equation, using shifted Chebyshev collocation method, for more details, see [14].

The solution to the linear Fredholm integro-differential problem given in this study uses a hybrid function made up of a combination of Bernstein polynomials and improved block pulse functions (1.1),

$$\sum_{i=0}^{s} p_i(x)u^{(i)}(x) = f(x) + \lambda \int_0^1 k(x,t)u(t)\,dt\,, \tag{1.1}$$

with the initial conditions

$$u^{(i)}(0) = \alpha_i \quad , \quad 0 \le i \le s-1$$

Where $k(x,t)$ is the kernel of the integral, $f(x)$ and $p_i(x)$ are known analytic functions, $u^{(i)}(x)$ is the *ith* derivative of the unknown function that will be determined, s is a positive integer and (λ, α_i) are suitable constants.

To make it more reader-friendly, we shall detail our proposed hybrid strategy in what follows. The following is the structure of this paper. In Section 2, we introduce a hybrid approach for connecting the Bernstein and Improved Block-Pulse functions. In section 3, the suggested method for numerically approximating linear Fredholm Integro-differential Equations on the basis of HBIBP is described. The section's numerical examples illustrate the accuracy and reliability of our suggested method for solving second-kind linear Fredholm Integro-differential Equations. We also add our concluding remarks.

2. Hybrid Bernstein improved block-pulse functions definition (HBIBPFs) [15, 25]

Improved Block-Pulse functions and Bernstein polynomials are used to create the complete orthogonal function $HBIBP_{i,j}(x)$, and then this set formas complete orthogonal system.

$HBIBP_{i,j}(x)$ where $j = 0,1,\dots,M, i = 1,2,\dots,N+1$, $HBIBP_{i,j}(x)$ have two arguments i and j.

Order of IBPFs (Improved Block-Pulse Functions) and degree of BPs (Bernstein Polynomials) are represented by i and j, respectively. $HBIBP(x)$ defined as follows on the interval [0, 1].:

$$HBIBP_{i,j}(x) = \begin{cases} B_{j,M}\left(\frac{2x}{h}\right) & , x \in \left[0, \frac{h}{2}\right), \\ 0 & , \ otherwise, \end{cases} \quad for\ i = 1, j = 0,1,\dots,M \tag{2.1}$$

$$HBIBP_{i,j}(x) = \begin{cases} B_{j,M}\left(\frac{x}{h} + \frac{3}{2} - i\right) & , x \in \left[(i-2)h + \frac{h}{2}, (i-1)h + \frac{h}{2}\right) \\ 0 & , \ otherwise, \end{cases} \quad for\ i = 2,3,\dots,N, j = 0,1,\dots,M \tag{2.2}$$

$$HBIBP_{i,j}(x) = \begin{cases} B_{j,M}\left(\frac{2x}{h} - \frac{2}{h} + 1\right) & , x \in \left[1 - \frac{h}{2}, 1\right), \\ 0 & , \ otherwise, \end{cases} \quad for\ i = N+1, j = 0,1,\dots,M \tag{2.3}$$

Thus, our new basis is $\{HBIBP_{1,0}, HBIBP_{1,1}, \dots, HBIBP_{N+1,M}\}$. We can approximate the function to the base function, where N is any positive integer and $h = \frac{1}{N}$. In the following part, we will specifically address the issue of approximating such functions.

2.1. Function approximation

A function $u(x)$ can be expressed as follows using the $HBIBP(x)$ basis:

$$u(x) \simeq \sum_{i=1}^{N+1} \sum_{j=0}^{M} c_{i,j}.HBIBP_{i,j}(x) = C^T HBIBP(x)\,, \tag{2.4}$$

where

$$HBIBP(x) = [HBIBP_{1,0}, HBIBP_{1,1}, \dots, HBIBP_{N+1,M}]^T\,, \tag{2.5}$$

and

$$C = [c_{1,0}, c_{1,1}, \dots, c_{N+1,M}]^T, \tag{2.6}$$

we have

$$C^T < HBIBP(x), HBIBP(x) > = < u(x), HBIBP(x), \tag{2.7}$$

then

$$C = L^{-1} < u(x), HBIBP > , \tag{2.8}$$

where $\langle .,. \rangle$ represents the standard inner product and L is an $((N+1)(M+1) \times (N+1)(M+1))$ matrix that is said the dual matrix that is

$$L = < HBIBP(x), HBIBP(x) >$$
$$= \int_0^1 HBIBP(x).HBIBP^T(x)dx$$

$$= \begin{pmatrix} L_1 & 0 & 0 & \cdots & 0 \\ 0 & L_2 & 0 & \cdots & 0 \\ 0 & 0 & L_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & L_{n+1} \end{pmatrix}, \tag{2.9}$$

$L_i (i = 1,2,\dots,n+1)$ is given as follows:

$$(L_1)_{i+1,j+1} = \int_0^{\frac{h}{2}} B_{i,M}\left(\frac{2x}{h}\right) B_{j,M}\left(\frac{2x}{h}\right) dx = \frac{h}{2}\int_0^1 B_{i,M}(x)B_{j,M}(x)dx$$

$$= \frac{h\binom{M}{i}\binom{M}{j}}{2(2M+1)\binom{2M}{i+j}}, \quad for\ i,j = 0,\dots,M, \qquad (2.10)$$

$$(L_r)_{i+1,j+1} = \int_{(i-2)h+\frac{h}{2}}^{(i-1)h+\frac{h}{2}} B_{i,M}\left(\frac{x}{h}+\frac{3}{2}-i\right) B_{j,M}\left(\frac{x}{h}+\frac{3}{2}-i\right) dx,$$

$$for\ r = 2,\dots,N$$

$$= h\int_0^1 B_{i,M}(x)B_{j,M}(x)dx = \frac{h\binom{M}{i}\binom{M}{j}}{(2M+1)\binom{2M}{i+j}}, \quad for\ i,j = 0,\dots,M, \qquad (2.11)$$

$$(L_{N+1})_{i+1,j+1} = \int_{1-\frac{h}{2}}^{1} B_{i,M}\left(\frac{2x}{h}-\frac{2}{h}+1\right) B_{j,M}\left(\frac{2x}{h}-\frac{2}{h}+1\right) dx$$

$$= \frac{h}{2}\int_0^1 B_{i,M}(x)B_{j,M}(x)dx = \frac{h\binom{M}{i}\binom{M}{j}}{2(2M+1)\binom{2M}{i+j}}, \quad for\ i,j = 0,\dots,M, \qquad (2.12)$$

The function $k(x,t) \in L^2([0,1]\times[0,1])$ can also be approximated as follow:

$$k(x,t) = HBIBP^T(x).K.HBIBP(t)$$

where K is an $(M+1)(N+1)$ matrix that we can obtain as follows:

$$K = L^{-1}\langle HBIBP(x), \langle k(x,t), HBIBP(t)\rangle\rangle L^{-1}.$$

2.2. Operational matrix of product

Consider this $C^T = [C_1^T, C_2^T, \dots, C_{N+1}^T]$ is an arbitrary $1\times(N+1)(M+1)$ matrix which C_i^T is $1\times(M+1)$ matrix for $i = 1,2,\dots,N+1$, then $\hat{C}$ is $(N+1)(M+1)\times(N+1)(M+1)$ operational matrix of product whenever

$$C^T HBIBP(x) HBIBP(x)^T \simeq HBIBP(x)^T\hat{C} \qquad (2.13)$$

We know

$$C^T B(x)B(x)^T \simeq B(x)^T\hat{C}_i, \quad i = 1,2,\dots,N+1,$$

Which $\hat{C}_i$ is the product of the Bernstein polynomials shown in [20, 21], then

$$C^T HBIBP(x)HBIBP(x)^T =$$

$$C^T\begin{bmatrix} HBIBP_{1,m}(x)HBIBP_{1,m}(x)^T & \overline{0} & \cdots & \overline{0} \\ \overline{0} & HBIBP_{2,m}(x)HBIBP_{2,m}(x)^T & \cdots & \overline{0} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{0} & \overline{0} & \cdots & HBIBP_{N+1,m}(x)HBIBP_{N+1,m}(x)^T \end{bmatrix}$$

$$= \begin{bmatrix} C_1 HBIBP_{1,m}(x)HBIBP_{1,m}(x)^T & \overline{0} & \cdots & \overline{0} \\ \overline{0} & C_2HBIBP_{2,m}(x)HBIBP_{2,m}(x)^T & \cdots & \overline{0} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{0} & \overline{0} & \cdots & C_{N+1}HBIBP_{N+1,m}(x)HBIBP_{N+1,m}(x)^T \end{bmatrix}$$

$$= \begin{bmatrix} HBIBP_{1,m}(x)^T\hat{C}_1 & \overline{0} & \cdots & \overline{0} \\ \overline{0} & HBIBP_{2,m}(x)^T\hat{C}_2 & \cdots & \overline{0} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{0} & \overline{0} & \cdots & HBIBP_{N+1,m}(x)^T\hat{C}_{N+1} \end{bmatrix}$$

$= HBIBP(x)^T\hat{C}$, for $m = 0,\dots,M$, where

$$\hat{C} = \begin{bmatrix} \hat{C}_1 & \overline{0} & \dots & \overline{0} \\ \overline{0} & \hat{C}_2 & \dots & \overline{0} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{0} & \overline{0} & \dots & \hat{C}_{N+1} \end{bmatrix},$$

with $\overline{0}$ is $(m+1)\times(m+1)$ matrix.

2.3. Operational Integration Matrix

The coefficient matrix $\bar{P}$ It should be possible to integrate more HBIBP functions into HBIBP functions. Following are the formulas for the operational integration matrix $\bar{P}$:

$$\int_0^x HBIBP(t)dt \simeq \bar{P}HBIBP(x), \qquad 0 \le x \le 1 \qquad (2.14)$$

where $\bar{P}$ is $(N+1)(M+1)$ square matrix and in Eq. (2.1)-(2.3), $HBIBP(x)$ is defined. It's simple to see that:

$$\int_0^1 B_{i,m}(x)dx = \frac{1}{m+1}, \quad i = 0,1,\dots,m$$

Then

$$\int_0^1 B_{i,m}(kx)dx = \frac{1}{k(m+1)}, \quad i = 0,1,\dots,m$$

.

On the other hand we know

$$\int_0^x B(t)dt \simeq PB(x)$$

where [23] and [24] provide information on how to obtain this matrix, where P is the Bernstein function's operational integration matrix of $B(x)$.

$$\int_0^x HBIBP_{i,j}(t)dt = \left[\frac{P}{2N}, \frac{\bar{1}}{2N(m+1)}, \cdots \frac{\bar{1}}{2N(m+1)}\right] HBIBP(x), \quad for\ i = 1, j = 0,1,\dots,M$$

$$\int_0^x HBIBP_{i,j}(t)dt = \left[\underbrace{\bar{0}, \quad \cdots \quad \bar{0},}_{i\ times} \frac{P}{N}, \frac{\bar{1}}{N(m+1)}, \cdots \frac{\bar{1}}{N(m+1)}\right] HBIBP(x),$$

$$for\ i = 2,3,\dots,N, j = 0,1,\dots,M.$$

$$\int_0^x HBIBP_{i,j}(t)dt = \left[\underbrace{\bar{0}, \cdots \bar{0},}_{i\ times} \frac{P}{2N}\right] HBIBP(x), \qquad for\ i = N+1, j = 0,1,\dots,M,$$

Where $\bar{0}$ is zero matrix of $(M+1)\times(M+1)$ dimension and $\bar{1}$ is a matrix that all of its elements is one of dimension $(M+1)\times(M+1)$.

The following is to how $\bar{P}$ is obtained:

Assume $Q_1 = \frac{P}{2N}$, $Q_2 = \frac{P}{N}$, $Z_1 = \frac{\bar{1}}{2N(m+1)}$ and $Z_2 = \frac{\bar{1}}{N(m+1)}$

$$\bar{P} = \begin{pmatrix} Q_1 & Z_1 & Z_1 & \cdots & Z_1 \\ \bar{0} & Q_2 & Z_2 & \cdots & Z_2 \\ \bar{0} & \bar{0} & Q_2 & \cdots & Z_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{0} & \bar{0} & \bar{0} & \cdots & Q_1 \end{pmatrix}$$

2.4. Operational differentiation matrix

The following section provides the operational matrix for differentiation $\bar{D}$ by:

$$\frac{dHBIBP(x)}{dx} = \bar{D}HBIBP(x).$$

We have

$\frac{dB(x)}{dx} = DB(x)$,

where D is the operational differentiation matrix of $B(x)$. Information on acquiring this matrix can be found in [23, 24].

$$\frac{dHBIBP_{i,j}(x)}{dx} = [2ND, \quad \bar{0}, \quad \cdots \quad \bar{0}]HBIBP(x),$$

$$for\ i = 1, j = 0,1, \dots, M$$

$$\frac{dHBIBP_{i,j}(x)}{dx} = \Big[\underbrace{\bar{0}, \quad \cdots \quad \bar{0},}_{i-1\ times} \quad ND, \quad \bar{0}, \quad \cdots \quad \bar{0}\Big]HBIBP(x),$$

$for\ i = 2,3, \dots, N, j = 0,1, \dots, M$

$$\frac{dHBIBP_{i,j}(x)}{dx} = \Big[\underbrace{\bar{0}, \quad \cdots \quad \bar{0},}_{i-1\ times} \quad 2ND\Big]HBIBP(x),$$

$$for\ i = N+1, j = 0,1, \dots, M,$$

where $\bar{0}$ is a matrix $(M+1) \times (M+1)$ that all of its elements is 0.

So

$$\bar{D} = N\begin{pmatrix} 2D & \bar{0} & \bar{0} & \dots & \bar{0} \\ \bar{0} & D & \bar{0} & \dots & \bar{0} \\ \bar{0} & \bar{0} & D & \dots & \bar{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{0} & \bar{0} & \bar{0} & \cdots & 2D \end{pmatrix}$$

3. Outline of solution

This section explains how to solve the sth-order linear Fredholm integro-differential equation with conditions (1.1) as initial condtions.

First, an approximation of the function u(x) is derived using

$$u(x) \simeq U^T HBIBP(x) = HBIBP^T(x)\mathrm{U} \quad , \quad (3.1)$$

where U is a (N+1)(M+1)-vector that is unknown and *HBIBP* is defined in (2.1) - (2.4) .

Second step, the functions $u^{(i)}(x)$, $i = 0, 1, \dots, s$ are approximately represented using

$$u^{(i)}(x) \simeq U^T(HBIBP(x))^{(i)} = U^T \bar{D}^i HBIBP(x) \ ,$$
$$i = 0,1, \dots, s \qquad (3.2)$$

Where $\bar{D}$ is the operational derivative matrix with the dimensions $(M+1)(N+1) \times (M+1)(N+1)$, $\bar{D}^i$ is operational derivative matrix with power i and $\frac{d}{dx}HBIBP(x) = \bar{D}HBIBP(x)$.

Third step, The function $k(x,t)$ is apprximated by

$$k(x,t) \simeq HBIBP^T(x)KHBIBP(t) \qquad (3.3)$$

where K is a $(N+1)(M+1) \times (N+1)(M+1)$-matrix.

Forth step: Substituting approximation eq. (3.1)-(3.3) into linear Fredholm integro-differential equation (1.1) produces

$$\sum_{i=0}^{s} p_i(x)u^{(i)}(x) = f(x) + \lambda \int_0^1 k(x,t)u(t)\,dt \,,$$

$$\sum_{i=0}^{s} HBIBP^T(x)(\bar{D}^i)^T U$$
$$= f(x) + \lambda \int_0^1 HBIBP^T(x).K.HBIBP(t)HBIBP^T(t)U\,dt \quad ,$$

$$\sum_{i=0}^{s} HBIBP^T(x)(\bar{D}^i)^T U$$
$$= f(x) + \lambda.HBIBP^T(x).K\int_0^1 HBIBP(t)HBIBP^T(t)U\,dt \quad ,$$

Where $L = \int_0^1 HBIBP(t).HBIBP^T(t)dt$ is defined in Eq. (2.9).

$$\sum_{i=0}^{s} HBIBP^T(x)(\bar{D}^i)^T U = f(x) + \lambda.HBIBP^T(x).K.L.U \,,$$

Therefore,

$$\sum_{i=0}^{s} HBIBP^T(x)(\bar{D}^i)^T U - \lambda.HBIBP^T(x).K.L.U = f(x), \qquad (3.5)$$

We collocate Eq (3.5) in order to find U in Newton-Cotes nodal points $(M+1)(N+1) - 1$ as

$$x_i = \frac{2i-1}{2(M+1)(N+1)} \quad , i = 1, \dots, (M+1)(N+1) - 1 \qquad (3.6)$$

From Eq. (3.5) using collocation point (3.6) and the conditions, there are $(M+1)(N+1)$ unknowns and $(M+1)(N+1)$linear equations in our system. We can obtain the unknown vector by resolving the above linear system. U and Eq. (3.1) can be used to find the u(x) solution. In the expansion of the $HBIBP$ function, see (3.1)

4. Numerical examples

To demonstrate the practicality and accuracy of the proposed approach in this study, we provide some test problems this section. The MATLAB software was used for all calculations (R2018b).

Example 1

Consider the first order integro-differential equation of Fredholm type [16-20]

$$u'(x) = (x+1)e^x - x + \int_0^1 xu(t)dt \quad , \quad 0 \le x,t \le 1,$$

with initial condition $u(0) = 0$.

The exact result of this problem is $u(x) = xe^x$.

In Table 1, Comparing composite Chebyshev, CAS wavelet, differential transformation, improved homotopy perturbation, sequential, and composite Chebyshev methods producing absolute errors in solutions. Table 1 demonstrates how closely the results of the current methodology approach the exact solution.

Example 2

Consider the first order integro-differential equation of Fredholm type [16, 17, 19 and 20]

$$u'(x) = 1 - \frac{1}{3}x + \int_0^1 xu(t)dt \quad , \quad 0 \le x, t \le 1,$$

with initial condition $u(0) = 0$. The exact solution to this problem is $u(x) = x$.

Example 3

Consider the Fredholm integro-differential equation of first order [21, 22]

$$u'(x) = e^{-x} + e^{-1} - 1 + \int_0^1 u(t)dt \quad , \quad 0 \le x, t \le 1,$$

with initial condition $u(0) = 1$. The exact solution to this problem is $u(x) = e^{-x}$.

Similarly, Table 3 and Fig. 1 exhibit the numerical solution of OAFM, block pulse functions, and the provided technique for Example 3. together with the absolute errors. It is evident that, when compared to the other two methods, our method is more accurate.

Table 1. Caparison of the Absolute error for present method with $M = 1$, $N = 2$.and some existing method

x	Absolute errors					
	CAS wavelet method [16]	DT method [17]	Improved homotopy perturbation [18]	Sequential Bases approach [19]	A new Schauder bases [20]	present method
0.1	1.34×10^{-3}	1.00×10^{-2}	0.23×10^{-5}	1.01×10^{-7}	1.8512×10^{-6}	$6.34569244 \times 10^{-16}$
0.2	1.15×10^{-3}	2.78×10^{-2}	0.92×10^{-5}	4.82×10^{-7}	1.3012×10^{-6}	$8.97251194 \times 10^{-17}$
0.3	5.67×10^{-3}	5.08×10^{-2}	0.20×10^{-4}	1.01×10^{-6}	1.4019×10^{-6}	$8.30585546 \times 10^{-17}$
0.4	5.93×10^{-2}	7.08×10^{-2}	0.37×10^{-4}	1.61×10^{-6}	2.1531×10^{-6}	$2.92574121 \times 10^{-16}$
0.5	1.32×10^{-2}	9.71×10^{-2}	0.57×10^{-4}	2.30×10^{-6}	5.0330×10^{-7}	$9.08359580 \times 10^{-16}$
0.6	4.39×10^{-2}	1.09×10^{-1}	0.83×10^{-4}	3.09×10^{-6}	2.5560×10^{-6}	$5.81364121 \times 10^{-16}$
0.7	1.41×10^{-2}	1.04×10^{-1}	0.11×10^{-3}	9.97×10^{-6}	2.2070×10^{-6}	$3.69494696 \times 10^{-16}$
0.8	1.34×10^{-2}	6.94×10^{-2}	0.14×10^{-3}	4.90×10^{-6}	2.5090×10^{-6}	$2.93069440 \times 10^{-16}$
0.9	1.32×10^{-2}	1.00×10^{-2}	0.18×10^{-3}	6.13×10^{-6}	3.4610×10^{-6}	$3.47676839 \times 10^{-15}$

Table 2. Comparison of the absolute errors for Example 2 determined using the proposed methods versus a different method.

x	Absolute errors					
	CAS wavelet method [13]	DT method[14]	The method [14]	Schauder bases [16]	A new Schauder bases [17]	Present method
0.1	$2.17942375 \times 10^{-4}$	$1.66666667 \times 10^{-3}$	2.06509×10^{-4}	3.7900×10^{-6}	9.6605×10^{-6}	3.27418×10^{-18}
0.2	$6.38548213 \times 10^{-4}$	$6.09388620 \times 10^{-3}$	8.04069×10^{-4}	1.5160×10^{-5}	6.0920×10^{-6}	1.24146×10^{-17}
0.3	$7.91370487 \times 10^{-4}$	$1.32017875 \times 10^{-2}$	1.72624×10^{-3}	3.4110×10^{-5}	5.5695×10^{-6}	1.62345×10^{-17}
0.4	$2.15586005 \times 10^{-2}$	$2.29140636 \times 10^{-2}$	2.86044×10^{-3}	6.0640×10^{-5}	8.0930×10^{-6}	9.00400×10^{-18}
0.5	$4.99358429 \times 10^{-3}$	$3.51578404 \times 10^{-2}$	4.04527×10^{-3}	9.4750×10^{-5}	2.6125×10^{-6}	3.41061×10^{-17}
0.6	$2.21728810 \times 10^{-2}$	$6.69648304 \times 10^{-2}$	9.18472×10^{-3}	1.3644×10^{-4}	6.0030×10^{-6}	1.55524×10^{-17}
0.7	$1.05645449 \times 10^{-4}$	$7.12430514 \times 10^{-2}$	5.06663×10^{-3}	1.8571×10^{-4}	1.3896×10^{-6}	1.52795×10^{-17}
0.8	$1.43233681 \times 10^{-3}$	$8.63983845 \times 10^{-2}$	5.65279×10^{-3}	2.4256×10^{-4}	1.7810×10^{-7}	1.52795×10^{-17}
0.9	$2.07747461 \times 10^{-2}$	$1.08103910 \times 10^{-1}$	4.10753×10^{-3}	3.0699×10^{-4}	1.3004×10^{-6}	1.74623×10^{-17}

Table 3. The numerical approximation and absolute errors for Example 3 using the proposed methods vs a different method.

x	Exact solution	method [21]	OAFM[22]	Presented method	Absolute errors		
					method [21]	OAFM [22]	Presented method
0.1	0.904837418	0.910993	0.904837	0.904837418	6.15558e-03	4.18036e-07	3.27418×10^{-14}
0.2	0.818730753	0.804005	0.818731	0.818730753	1.47258e-02	2.46922e-07	1.24146×10^{-13}
0.3	0.740818221	0.755324	0.740818	0.740818221	1.45058e-02	2.20682e-07	1.62345×10^{-13}
0.4	0.670320046	0.666636	0.670320	0.670320046	3.68405e-03	4.60356e-08	9.00400×10^{-14}
0.5	0.606530660	0.588375	0.606530	0.606530660	1.81557e-02	6.59713e-07	3.41061×10^{-13}
0.6	0.548811636	0.552766	0.548812	0.548811636	3.95436e-03	3.63906e-07	1.55524×10^{-13}
0.7	0.496585304	0.487894	0.496585	0.496585304	8.69130e-03	3.03791e-07	1.52795×10^{-13}
0.8	0.449328964	0.458378	0.449329	0.449328964	9.04904e-03	3.58828e-08	1.52795×10^{-13}
0.9	0.406569660	0.404606	0.406570	0.406569660	1.96366e-03	3.40259e-07	1.74623×10^{-13}

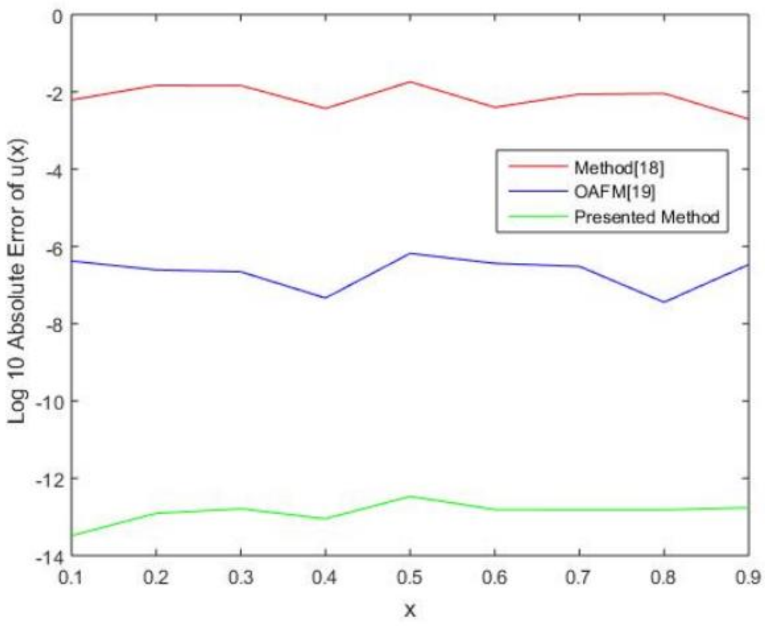

Fig. 1 Absolute error comparison for Example 3 with M=1,N=2 for the presented method with method [21] and OAF method [22]

5. Conclusion

This study attempts to numerically solve the linear Fredholm integro-differential equations using the hybrid Bernstein and improved block-pulse functions presented in [15, 25]. The proposed (HBIBPFs) are tested for accuracy and applicability using illustrative examples. According to the numerical results, the proposed hybrid approach's accuracy is superior to that of composite Chebyshev, CAS wavelet, improved homotopy perturbation, sequential, and composite Chebyshev methods and new Schauder bases. The method is particularly promising for handling more diversified nonlinear integro-differential equations that the authors are researching, as evidenced by the numerical findings.

6. References

1. **H., R., Thiem. (1977).** A model for spatio spread of an epidemic, J. Math. Bio **4, 337–351.**
2. **P., Darania., and A., Ebadian. (2007).** A method for the numerical solution of the integro-differential equations, Appl. Math. Comput. **188, 657–668**
3. **U., Lepik. (2006).** Haar wavelet method for nonlinear integro-differential equations, Appl. Math. Comput. **176, 324–333.**
4. **M., Erfanian., and M., Gachpazan. (2015).** Rationalized Haar wavelet bases to approximate solution of nonlinear Fredholm integral equations with error analysis, Appl. Math. Comput. **265, 304–312.**
5. **M., Erfanian., M., Gachpazan, and H., Beiglo. (2015).** Solving mixed Fredholm–Volterra integral equations by using the operational matrix of RH wavelets, SeMA J. **69, 25–36.**
6. **K., Maleknejad., B., Basirat., and E., Hashemizadeh. (2011).** Hybrid Legendre polynomials and Block-pulse functions approch for nonlinear Volterr–Fredholm integro-differential equations, Comput. Math. Appl. **61, 2821–2828.**
7. **E., Babolian., Z., Masouria., and S., Hatamzadeh-Varmazyar. (2008).** New direct method to solve nonlinear Volterra–Fredholm integral and integro-differential equations using operational matrix with block-pulse functions, Prog. Electromagn. Res. **8, 59–76.**
8. **E., Babolian., Z., Masouria., and S., Hatamzadeh-Varmazyar. (2009).** Numerical solution of nonlinear Volterra–Fredholm integro-differential equations via direct method using triangular function, Comput. Math. Appl. **58,239–247.**
9. **B., Sepehriana., and M., Razzaghi. (2004).** Single-term Walsh series method for the Volterra integro-differential equations, Eng. Anal. Bound. Elem. **28, 1315–1319.**
10. **A., Avudainayagam., and C., Vani. (2009).** Wavelet–Galerkin method for integro-differential equations, Word App. Sci. J. **7, 50–56.**
11. **J., Zhao., and R., M., Corless. (2006).** Compact finite difference method for integro-differential equations, Appl. Math. Comput. **177, 271–288.**
12. **Y., H., Youssri., & A., G., (2024).** Atta. Fejér-Quadrature Collocation Algorithm for Solving Fractional Integro-Differential Equations via Fibonacci Polynomials, Contemporary Mathematics, Volume **5** Issue **1,** **https://doi.org/10.37256/cm.5120244054.**
13. **G., Atta., and Y., H., Youssri. (2022).** Advanced shifted first-kind Chebyshev collocation approach for solving the nonlinear time-fractional partial integro-differential equation with a weakly singular kernel, Computational and Applied Mathematics , **41:381.**
14. **Y., H., Youssri., and R., M., Hafez. (2020).** Chebyshev collocation treatment of Volterra–Fredholm integral equation with error analysis, Arab. J. Math. , **9:471–480.**
15. **M., A., Ramadan., & H., S., Osheba. (2020).** A New Hybrid Orthonormal Bernstein and Improved Block-Pulse Functions Method for Solving Mathematical Physics and Engineering Problems, Alexandria Engineering Journal, Volume **59**, Issue **5**, **October**, Pages **3643-3652.**
16. **H., Danfu., and S., Xufeng. (2007).** Numerical solution of integro-differential equations by using CAS wavelet operational matrix of integration, Appl. Math. Comput. **194, 460–466.**
17. **P., Darania., and A., Ebadian. (2007).** A method for the numerical solution of the integro-differential equations, Appl.Math. Comput. **188, 657–668.**
18. **E., Yusufoglu. (Agadjanov). (2009).** Improved homotopy perturbation ˇ method for solving Fredholm type integro-differential equations, Chaos, Solitons and Fractals, **vol. 41, no. 1, pp. 28–37.**
19. **M., I., Berenguer., M., V., Fernandez Munoz., A., I., Garralda- Guillem., and M., Ruiz Galan. (2012).** A sequential approach for solving the Fredholm integro-differential equation, Applied Numerical Mathematics, **vol. 62, no. 4, pp. 297–304.**

20. **Erfanian., M., Gachpazan., M., & Beiglo., H. (2017).** A new sequential approach for solving the integro-differential equation via Haar wavelet bases. *Computational Mathematics and Mathematical Physics*, ***57*(2), 297-305.**

21. **Rahmani., Leyla., Bijan., Rahimi., and Mohammad., Mordad. (2011).** Numerical Solution of Volterra-Fredholm Integro-Differential Equation by Block Pulse Functions and Operational Matrices. Gen., **4(2), 37-48.**

22. **L., Zada., M., Al-Hamami., R., Nawaz., S., Jehanzeb., A., Morsy., A., A., Abdel-Aty., and K., S., Nisar. (2021).** A new approach for solving fredholm integro-differential equations. *Information Sciences Letters*, ***10*(3), 3.**

23. **S., A., Yousefi., & M., Behroozifar. (2010).** Operational matrices of Bernstein polynomials and their applications, Int. J. Syst. Sci. **41 709-716.**

24. **M., Behroozifar., S., A., Yousefi. (2013).** Numerical solution of delay differential equations via operational matrices of hybrid of block-pulse functions and Bernstein polynomials, Computational Methods for Differential Equations **1.2 78-95.**

25. **Ji-Huan., He., Mahmoud., H., Taha., Mohamed., A., Ramadan., and Galal., M., Moatimid. (2022).** A Combination of Bernstein and Improved Block-Pulse Functions for Solving a System of Linear Fredholm Integral Equations, Mathematical Problems in Engineering, Volume, Article ID **6870751, 12** pages **https://doi.org/10.1155/2022/6870751.**