A Reliable Numerical Treatment of Differential Equations via Hybrid Bernstein and Improved Block-Pulse Functions

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

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

ARTICLE INFO
Received 31 December 2023
Accepted 22 January 2024

ABSTRACT
Many branches of practical mathematics rely heavily on numerical solutions to initial value problems, boundary value problems, and eigenvalue problems for ordinary and partial differential equations. Recently, the authors attempted to solve integral equations using hybrid Bernstein functions and improved block pulse functions. However, this is the first study to present a technical coupling between hybrid Bernstein and improved block-pulse functions for solving differential equations. The current method transforms differential equations into an algebraic system that can be solved with conventional methods. To validate the new method, certain numerical examples are supplied. The findings demonstrated that the method is both promising and highly accurate. The numerical findings reveal that the suggested hybrid approach outperforms the use of Bernstein polynomials, multi derivative hybrid block methods, block-pulse function, and other methods indicated in the numerical section in terms of accuracy. The proposed method can be implemented for more kinds of differential equations. The proposed method is applicable to a broader range of differential equations. The numerical results show that the proposed method are highly accurate and effective.

1. Introduction
Since ordinary differential equations (ODEs) are used to simulate a wide range of physical, biological, and social events, solving ODEs is crucial in many branches of research and engineering. We can forecast how these occurrences will change over time and comprehend their behavior by solving ODEs.

The ability to analyses a system's underlying dynamics even with sparse or insufficient data is one of the main advantages of solving ODEs. Solving ODEs may show how various variables interact and change over time in a system by capturing the relationships between them. Direct extraction of this information from experimental data is frequently challenging or impossible.
ODE solutions can also be used to control systems or forecast the results of certain operations. ODEs are utilized, for instance, in engineering to develop control systems that manage the actions of machinery or processes. ODEs are used in medicine to simulate the transmission of diseases and assess the efficacy of various therapies. In conclusion, the significance of solving ODEs lies in their ability to shed light on complicated system behavior that cannot be discovered from data collection alone. It enables us to assess the results of various actions, optimize systems, and forecast how a system will change over time. Despite the fact that multiple orthogonal and non-orthogonal polynomials have been created and applied to linear and nonlinear differential equations such as Chebyshev, Legendre, Jacobi, Bernstein, Bernoulli, Genocchi, Lucas, Laguerre, Hermite, and Bell.

In recent years, several researchers have proposed direct approaches for solving higher-order ODEs. Majid et al. [1] utilized Jacobi iteration and direct methods with varying step sizes to solve second-order ODEs. Awoyemi and Idowu [2] devised a type of hybrid collocation direct approach for solving third-order ODEs. Olabode and Yusuph [3] used power series collocation and interpolation to develop a three-step block technique for solving third-order ODEs. Waeleh et al. [4] devised a block technique based on numerical integration and interpolation to solve higher-order ODEs, and they gave fourth- and fifth-order approximations. Olabode and Alabi [5] proposed a linear multistep technique for solving fourth-order ODEs that employs interpolation and the collocation of power series approximation solutions.


In this article, we first show the relationship between the Bernstein polynomials and the enhanced block pulse basis. Using this relation, we created the operational matrix for integration and product of B-polynomials. They are used for solving ordinary differential equations. The current method turns an ordinary differential equation to a series of algebraic equations. We used the proposed method on three test issues and compared the results to the precise solutions and other ways, demonstrating that it is very effective and convenient. Then we utilise them to solve differential equations

$$\sum_{j=0}^{s} \rho_j(x)y^{(j)}(x) = g(x), \quad 0 \leq x \leq 1,$$  \hspace{0.5cm} (1.1)

With the initial condition

$$y^{(k)}(0) = b_k, \quad 0 \leq k \leq s - 1,$$  \hspace{0.5cm} (1.2)

Where $g(x)$ and $\rho_j(x), j = 0, \ldots, s$ are given functions and $y(x)$ is the unknown function to be determined. The key feature of this technique is that it lowers these equations to those of an easily solvable algebraic equation, significantly simplifying them.

The paper is structured as follows: Section 2 presents a combination of the Bernstein and Improved Block-Pulse functions. Section 3 presents an approach for numerically approximating linear and nonlinear differential equations using the HBIBP basis. The accuracy and applicability are determined by solving several linear and nonlinear differential equations. Finally, the last part summarizes the findings.
2. Hybrid Bernstein improved block-pulse functions (HBIBPFs) \[\text{[13]}\]

**Definition 2.1:** HBIBP\(_{i,j}(x)\) is a combination of Bernstein polynomials and Improved Block-Pulse functions where all are orthogonal and complete, and then the set is a complete orthogonal system. HBIBP\(_{i,j}(x)\) where \(j = 0, 1, \ldots, M, i = 1, 2, \ldots, N + 1\), HBIBP\(_{i,j}(x)\) have two arguments \(i\) and \(j\) are the order of IBPFs and degree of BPs, respectively.

\(\text{HBIBP}(x)\) defined on the interval \([0, 1]\) as follows:

\[
\text{HBIBP}_{i,j}(x) = \begin{cases} 
B_{i,j} \left( \frac{x}{h} \right), & x \in \left(0, \frac{1}{h} \right), \\
0, & \text{otherwise},
\end{cases}
\]

\(i = 1, 2, \ldots, M, j = 0, 1, \ldots, N\).

\[
\text{HBIBP}_{i,j}(x) = \begin{cases} 
B_{i,j} \left( \frac{x}{h} + \frac{1}{2} \right), & x \in \left(\frac{1}{h} + \frac{1}{2}, 1 \right), \\
0, & \text{otherwise},
\end{cases}
\]

\(i = 1, 2, \ldots, N, j = 0, 1, \ldots, M\).

\[
\text{HBIBP}_{i,j}(x) = \begin{cases} 
B_{i,j} \left( \frac{x}{h} + \frac{1}{2} + 1 \right), & x \in \left(1, 1 + \frac{1}{h} \right), \\
0, & \text{otherwise},
\end{cases}
\]

\(i = 1, 2, \ldots, N, j = 0, 1, \ldots, M\).

Thus, our new basis is \(\{\text{HBIBP}_{1,0}, \text{HBIBP}_{1,1}, \ldots, \text{HBIBP}_{N+1,M}\}\) and we can approximate function to the base function, where \(N\) is an arbitrary positive integer and \(h = \frac{1}{N}\). In the next section we really are dealing with the problem of approximating such functions.

2.1. Function approximation

Using the HBIBP basis, a function \(u(x)\) can be represented as:

\[
u(x) = \sum_{i=1}^{N+1} \sum_{j=0}^{M} c_{i,j} \cdot \text{HBIBP}_{i,j}(x) = C^T \text{HBIBP}(x),
\]

where

\[
\text{HBIBP}(x) = [\text{HBIBP}_{1,0}, \text{HBIBP}_{1,1}, \ldots, \text{HBIBP}_{N+1,M}]^T,
\]

and

\[
C = [c_{1,0}, c_{1,1}, \ldots, c_{N+1,M}]^T.
\]

We have

\[
C^T < \text{HBIBP}(x), \text{HBIBP}(x) > = \langle u(x), \text{HBIBP}(x) \rangle,
\]

then

\[
C = L^{-1} \langle u(x), \text{HBIBP} \rangle,
\]

where \(\langle \cdot, \cdot \rangle\) is 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

\[
L = \langle \text{HBIBP}(x), \text{HBIBP}(x) \rangle = \int_0^1 \text{HBIBP}(x) \cdot \text{HBIBP}^T(x) \, dx
\]

\[
= \begin{pmatrix} L_1 & 0 & 0 & \cdots & 0 \\
0 & L_2 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & L_n & 0 \\
0 & 0 & \cdots & 0 & L_{n+1}
\end{pmatrix},
\]

\(L_i(i = 1, 2, \ldots, n + 1)\) is defined as follows,

\[
(L_i)_{i + j + 1} = \frac{2^{i+j}}{2(2M + 1)} \int_0^1 B_{i,j}(x) B_{i,j}(x) \, dx = \frac{1}{2} B_{i,j}(x) B_{i,j}(x) \, dx
\]

\[
= \frac{h^M}{2(2M + 1)} \binom{M}{i+j} (i+j)!, \quad \text{for } i = 0, \ldots, M;
\]

\[
(L_u)_{i + j+1} = \int_0^1 B_{i,j}(x) B_{i,j}(x) \, dx = \frac{1}{2} B_{i,j}(x) B_{i,j}(x) \, dx
\]

\[
= \frac{h^M}{2(2M + 1)} \binom{M}{i+j} (i+j)!, \quad \text{for } i = 0, \ldots, M;
\]

\[
(L_{u+1})_{i + j+1} = \int_0^1 B_{i,j}(x) B_{i,j}(x) \, dx = \frac{1}{2} B_{i,j}(x) B_{i,j}(x) \, dx
\]

\[
= \frac{h^M}{2(2M + 1)} \binom{M}{i+j} (i+j)!, \quad \text{for } i = 0, \ldots, M;
\]

We can also approximate the function \(k(x, t) \in L^2([0, 1] \times [0, 1])\) as follow:

\[
k(x, t) = \text{HBIBP}^T(x) \cdot K \cdot \text{HBIBP}(t)
\]

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

\[
K = L^{-1}(\text{HBIBP}(x), (k(x, t), \text{HBIBP}(t))) L^{-1}.
\]

2.2. Operational product matrix

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

\[
C^T \text{HBIBP}(x) \cdot \text{HBIBP}(x)^T = \hat{C} \text{HBIBP}(x)^T \cdot \text{HBIBP}(x)
\]

We know

\[
C^T B(x) \cdot B(x) = B(x)^T \hat{C} B(x), \quad i = 1, 2, \ldots, N + 1,
\]

which \(\hat{C}_i\) is operational matrix of product of Bernstein polynomials presented in [14, 15], then

\[
C^T \text{HBIBP}(x) \cdot \text{HBIBP}(x)^T = \begin{pmatrix}
\text{HBIBP}_{1,0}(x) & \text{HBIBP}_{1,1}(x) & \cdots & \text{HBIBP}_{1,M}(x) \\
\text{HBIBP}_{2,0}(x) & \text{HBIBP}_{2,1}(x) & \cdots & \text{HBIBP}_{2,M}(x) \\
\vdots & \vdots & \ddots & \vdots \\
\text{HBIBP}_{N+1,0}(x) & \text{HBIBP}_{N+1,1}(x) & \cdots & \text{HBIBP}_{N+1,M}(x)
\end{pmatrix}
\]

\[
= \text{HBIBP}(x)^T \hat{C} \text{HBIBP}(x),
\]

for \(m = 0, \ldots, M\), where

2.3. Operational Integration Matrix

HBIBP functions can be integrated using the coefficient matrix $P'$. The operational matrix for integration $P'$ is given by

$$
\int_0^x HBI(x) \, dx = P HBI(x), \quad 0 \leq x \leq 1
$$

where $P$ is $(N + 1)(M + 1)$ square matrix and $HBI(x)$ is defined in Eq. (2.1)-(2.3). It is easy to see that:

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

Then

$$
\int_0^x B_i(mx) \, dx = \frac{1}{m+1}, \quad i = 0, 1, \ldots, m,
$$

On the other hand we know

$$
\int_0^x B(t) \, dt = PB(x)
$$

which $P$ is the operational matrix of integration of Bernstein function $(B(x))$ and information on obtaining this matrix are given in [14] and [15].

$$
\int_0^x HBI(x) \, dx = \left[ \begin{array}{c} P_1 \\ \vdots \\ P_M \end{array} \right] HBI(x),
$$

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

$$
\int_0^x HBI(x) \, dx = \left[ \begin{array}{c} 0 \\ \vdots \\ 0 \end{array} \right] HBI(x),
$$

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

$$
\int_0^x HBI(x) \, dx = \left[ \begin{array}{c} 0 \\ \vdots \\ 0 \end{array} \right] HBI(x),
$$

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

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

So

$$
\mathbf{P} = \begin{pmatrix}
A_1 & B_1 & B_1 & \cdots & B_1 \\
0 & A_2 & B_2 & \cdots & B_2 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & A_1
\end{pmatrix}
$$

3. Solution Methodology and Illustrative Examples

3.1. Solution Methodology

This section presents the derivation of the method for solving the linear differential equation (1.1) with the initial conditions Eq. (1.2). If we approximate $g(x), p_j(x), j = 0, \ldots, s$ and $y^{(s)}(x)$ as follows:

$$
g(x) = G HBI(x)
$$

$$
p_j(x) = P_j HBI(x), \quad j = 0, \ldots, s,
$$

$$
y^{(s)}(x) = C HBI(x)
$$

Where $G, P_j, j = 0, \ldots, s$ and $C$ are the coefficients which are defined similarly to Eq. (2.8). With $s$-times integrating from Eq. (3.2) with respect to $x$ between $x = 0$ to $x = x$, using Eq. (2.14) and the initial conditions Eq. (1.2), we will have

2.4. Operational differentiation matrix

The operational matrix of differentiation $D$ is given by

$$
\frac{d(HBI(x))}{dx} = D HBI(x)
$$

We have

$$
\frac{dB(x)}{dx} = D B(x).
$$

Which $D$ is the operational matrix of differentiation of $B(x)$ and information on obtaining this matrix is given in [14, 15].

$$
\frac{dHBI(x)}{dx} = \left[ \begin{array}{c} 2N & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array} \right] HBI(x),
$$

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

$$
\frac{dHBI(x)}{dx} = \left[ \begin{array}{c} 0 \\ \vdots \\ 0 \end{array} \right] HBI(x),
$$

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

$$
\frac{dHBI(x)}{dx} = \left[ \begin{array}{c} 0 \\ \vdots \\ 0 \end{array} \right] HBI(x),
$$

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

Therefore, the operational matrix of integration $P$ is obtained as follows:

Assume $A_1 = \frac{1}{2N}, A_2 = \frac{1}{N}, B_1 = \frac{1}{2N(m+1)}$ and $B_2 = \frac{1}{N(m+1)}$

$$
\mathbf{P} = \begin{pmatrix}
A_1 & B_1 & B_1 & \cdots & B_1 \\
0 & A_2 & B_2 & \cdots & B_2 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & A_1
\end{pmatrix}
$$
Let \( y'^{i-1}(x) = b_{i-1} + C^T P H B I B P(x) \) and \( y'^{i-2}(x) = b_{i-2} + b_{i-1} x + C^T P H B I B P(x) \) be the unknown functions.

Substituting Eq. (3.4) into Eq. (3.3), we have
\[
 y'(x) = b_1 + b_0 + \frac{b_0}{2!} x^2 + \cdots + \frac{b_0}{(s-2)!} x^{s-2} + c_{s-1}^{s-1} P H B I B P(x)
\]
\[
 y(x) = b_0 + b_0 x + \frac{b_0}{2!} x^2 + \cdots + \frac{b_0}{(s-1)!} x^{s-1} + c_{s}^{s} P H B I B P(x)
\]

Replacing Eq. (3.6) and Eq. (3.7) into (1.1), we obtain
\[
 y'(x) = \left( \sum_{j=0}^{s} P_{j}^{s} H B I B P(x) \right) y(x) = \sum_{j=0}^{s} P_{j}^{s} H B I B P(x) \cdot \sum_{j=0}^{s} H B I B P(x) = \sum_{j=0}^{s} P_{j}^{s} H B I B P(x) \cdot \sum_{j=0}^{s} H B I B P(x) = \sum_{j=0}^{s} \tilde{P}_{j} \tilde{Q}_{j} = G\x \quad (3.8)
\]

Using Eq. (2.13), we have
\[
 \sum_{j=0}^{s} H B I B P(x) \cdot \tilde{P}_{j} \tilde{Q}_{j} = H B I B P(x) \cdot G\x
\]

Therefore, we get
\[
 \sum_{j=0}^{s} \tilde{P}_{j} \tilde{Q}_{j} = G\x
\]

The unknown Vector \( C \) can be obtained by solving Eq. (3.10), once \( C \) is known, \( y(x) \) can be calculated from Eq. (3.7).

3.2. Illustrative examples

Example 1: Application to the Bessel differential equation [14]

Consider the following zero-order Bessel differential equation (O’Neil 1987).
\[
 x y'' + y' + xy = 0
\]

with initial conditions \( y(0) = 1 \), \( y'(0) = 0 \).

The exact solutions known as the Bessel function of the first kind of order zero denoted by \( J_0(x) \)
\[
 J_0(x) = \sum_{q=0}^{\infty} \left( -1 \right)^q \left( \frac{x}{2} \right)^{2q} \frac{(q)!^2}{q!}
\]

First, we approximate the unknown function \( y''(x) \) by
\[
 y''(x) \approx \sum_{i=0}^{N} c_{i} H B I B P_{i}(x) = C^T H B I B P(x)
\]

Using Eq. (2.14) and initial conditions we have
\[
 y'(x) = C^T P H B I B P(x)
\]
\[
 y(x) = C^T P^2 H B I B P(x) + C^T P H B I B P(x)
\]

Where 1 = \( 2^T H B I B P(x) \) and we can express function \( x \) as
\[
 x = E^T H B I B P(x)
\]

Substituting Eq. (3.11) – (3.13) in the Bessel differential equation we obtain
\[
 E^T H B I B P(x) \cdot C^T H B I B P(x) + C^T P H B I B P(x) + E^T H B I B P(x) \cdot [C^T P^2 H B I B P(x) + Z^T H B I B P(x)] = 0
\]

Replacing Eq. (3.11) into (1.1), we obtain
\[
 E^T H B I B P(x) \cdot H B I B P(x) = \sum_{j=0}^{s} \tilde{P}_{j} \tilde{Q}_{j} = G
\]

The unknown Vector \( C \) can be obtained by solving Eq. (3.10), once \( C \) is known, \( y(x) \) can be calculated from Eq. (3.7).

Example 2: Consider the nonlinear boundary layer equation \([17, 18, 19]\)
\[
 y'''(x) + y'(x) + xy = 0
\]

with initial conditions \( y(0) = 0 \), \( y'(0) = 0 \), \( y''(0) = A \).

This equation is well-known as the Blasius equation. Solving the Blasius equation yields the value \( y''(0) \), which is used to evaluate shear stress at the plate.

Blasius' equation has been solved using several approaches, such as series expansions, Runge Kutta, differential transformation, and others.

First, we approximate the unknown function \( y'''(x) \) by
\[
 y'''(x) \approx \sum_{i=0}^{N} c_{i} H B I B P_{i}(x) = C^T H B I B P(x)
\]

Using Eq. (2.14) and initial conditions we have
\[
 y''(x) = C^T P H B I B P(x) + a^T H B I B P(x) = C^T P H B I B P(x) + a^T H B I B P(x)
\]

Where 1 = \( 2^T H B I B P(x) \) and we can express function \( x \) as
\[
 x = E^T H B I B P(x)
\]

Substituting Eq. (3.11) – (3.13) in the Bessel differential equation we obtain
\[
 E^T H B I B P(x) \cdot C^T H B I B P(x) + C^T P H B I B P(x) + E^T H B I B P(x) \cdot [C^T P^2 H B I B P(x) + Z^T H B I B P(x)] = 0
\]

Replacing Eq. (3.11) into (1.1), we obtain
\[
 E^T H B I B P(x) \cdot H B I B P(x) = \sum_{j=0}^{s} \tilde{P}_{j} \tilde{Q}_{j} = G
\]

The unknown Vector \( C \) can be obtained by solving Eq. (3.10), once \( C \) is known, \( y(x) \) can be calculated from Eq. (3.7).
Where we approximate $\tilde{\lambda}, \tilde{\mu}, \tilde{\nu}$, then

$$Y = G T H B I B P(x)$$

then we get

$$2C T H B I B P(x) + H B I B P(x) T \tilde{G} R = 0,$$

Thus we have

$$\left. C = -\frac{1}{2} \tilde{G} R. \right\}$$

Our methodology's findings are widely acknowledged (see Table 2) when compared to the hybrid block approach [17], the Bernstein polynomials method [18], and the Bernoulli polynomials operational matrix [19] for the constant $A = 1$.

**Example 3:** Next consider the following problem [20, 21 and 22]

$$y'''(x) + 5y''(x) + 7y'(x) + 3y = 0, \quad (3.25)$$

with initial conditions $y(0) = 1, y'(0) = 0, y''(0) = -1$. The exact solution is $y(x) = e^{-x} + xe^{-x}$.

First, we approximate the unknown function $y'''(x)$ by

$$y'''(x) = \sum_{i=1}^{N+1} \sum_{j=0}^{M} c_{i,j} H B I B P_{i,j}(x) = C T H B I B P(x), \quad (3.26)$$

Using Eq. (2.14) and initial conditions we have

$$y''(x) = C T P H B I B P(x) - 1 = C T P H B I B P(x) - Z T H B I B P(x) \quad (3.27)$$

Where we approximate $1 \approx Z T H B I B P(x)$ and

$$y'(x) = C T P^2 H B I B P(x) - Z T P H B I B P(x) \quad (3.28)$$

$$y(x) = C T P^3 H B I B P(x) - Z T P^2 H B I B P(x) + 1$$

$$= C T P^3 H B I B P(x) - Z T P^2 H B I B P(x) + Z T H B I B P(x) \quad (3.29)$$

Substituting Eq. (3.26)–(3.29) in Eq. (3.25) we obtain

$$y'''(x) + 5y''(x) + 7y'(x) + 3y = 0,$$

$$C T H B I B P(x) + 5\left(C T P H B I B P(x) - Z T H B I B P(x)\right)$$

$$\quad + 7\left(C T P^2 H B I B P(x) - Z T P H B I B P(x)\right)$$

$$\quad + 3\left(C T P^3 H B I B P(x) - Z T P^2 H B I B P(x) + Z T H B I B P(x)\right) = 0,$$

$$C T (I + 5\tilde{P} + 7\tilde{P}^2 + 3\tilde{P}^3) = 42T + 7ZT \tilde{P} + 32T \tilde{P}^2$$

$$\left(I + 5\tilde{P} + 7\tilde{P}^2 + 3\tilde{P}^3 \right) C = 42T + 7ZT \tilde{P} + 32T \tilde{P}^2$$

$$C = \left(I + 5\tilde{P} + 7\tilde{P}^2 + 3\tilde{P}^3 \right)^{-1} (42T + 7ZT \tilde{P} + 32T \tilde{P}^2)$$

Here, we solve the same problem using Hybrid Bernstein improved block-pulse functions (HBIBPFs), with $M = 3, N = 3$. In Table 4, a comparison is made between Absolute Errors of the approximate values using the present approach Linear Multistep Hybrid Method [20], Bernstein polynomials [21] and Hybrid Block method [22].

**Example 4:** Consider the following problem [22, 23]

$$y'''(x) - y''(x) + y'(x) - y = 0, \quad (3.30)$$

with initial conditions $y(0) = 1, y'(0) = 0, y''(0) = -1$. The exact solution is $y(x) = \cos(x)$.

First, we approximate the unknown function $y'''(x)$ by

$$y'''(x) = \sum_{i=1}^{N+1} \sum_{j=0}^{M} c_{i,j} H B I B P_{i,j}(x)$$

$$= C T H B I B P(x), \quad (3.31)$$

Using Eq. (2.14) and initial conditions we have

$$y''(x) \approx C T P H B I B P(x) - 1 = C T P H B I B P(x) - Z T H B I B P(x) \quad (3.32)$$

Where we approximate $1 \approx Z T H B I B P(x)$ and

$$y'(x) \approx C T P^2 H B I B P(x) - Z T P H B I B P(x) \quad (3.33)$$

$$y(x) \approx C T P^3 H B I B P(x) - Z T P^2 H B I B P(x) + 1$$

$$= C T P^3 H B I B P(x) - Z T P^2 H B I B P(x) + Z T H B I B P(x) \quad (3.34)$$

Substituting Eq. (3.31)–(3.34) in Eq. (3.30) we obtain

$$y'''(x) - y''(x) + y'(x) - y = 0,$$

$$C T H B I B P(x) - \left(C T P H B I B P(x) - Z T H B I B P(x)\right)$$

$$+ \left(C T P^2 H B I B P(x) - Z T P H B I B P(x)\right)$$

$$- \left(C T P^3 H B I B P(x) - Z T P^2 H B I B P(x) + Z T H B I B P(x)\right)$$

$$= 0,$$

$$C T (I - \tilde{P} + \tilde{P}^2 - \tilde{P}^3) = Z T \tilde{P} - Z T \tilde{P}^2$$

$$\left(I - \tilde{P} + \tilde{P}^2 - \tilde{P}^3 \right) C = Z T \tilde{P} - Z T \tilde{P}^2$$

$$C = \left(I - \tilde{P} + \tilde{P}^2 - \tilde{P}^3 \right)^{-1} (Z T \tilde{P} - Z T \tilde{P}^2)$$

Here, we solve this problem using Hybrid Bernstein improved block-pulse functions (HBIBPFs), with $M = 3, N = 3$. In Table 4, a comparison is made between Absolute Errors of the approximate values using the present approach, multi derivative hybrid block methods [23] and Hybrid Block method [22].
Table 1. Comparison of the absolute error for numerical results for Example 1 of the given approach with $M = 3$, $N = 3$, Bernstein polynomials method $^{[14]}$, and the exact solution of $f_0(x)$.

| $x$  | Exact solution | Bernstein-polynomials for $m = 10$ | Presented Method | Absolute Error
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Bernstein- polynomials</td>
</tr>
<tr>
<td>0</td>
<td>1.00000000000000</td>
<td>1.00000000000000038</td>
<td>1.0000000000000000</td>
<td>3.774758284e-15</td>
</tr>
<tr>
<td>0.1</td>
<td>0.99750156206604</td>
<td>0.9975015620660412</td>
<td>0.99750156206604</td>
<td>1.221245327e-15</td>
</tr>
<tr>
<td>0.2</td>
<td>0.99002497223958</td>
<td>0.9900249722395755</td>
<td>0.99002497223958</td>
<td>4.44082099e-15</td>
</tr>
<tr>
<td>0.3</td>
<td>0.97762624653830</td>
<td>0.9776262465382969</td>
<td>0.97762624653830</td>
<td>3.108624469e-15</td>
</tr>
<tr>
<td>0.4</td>
<td>0.96039822665956</td>
<td>0.960398226659563</td>
<td>0.96039822665956</td>
<td>2.997602166e-15</td>
</tr>
<tr>
<td>0.5</td>
<td>0.93846980724081</td>
<td>0.9384698072408127</td>
<td>0.93846980724081</td>
<td>2.775557562e-15</td>
</tr>
<tr>
<td>0.6</td>
<td>0.91200486349721</td>
<td>0.9120048634972114</td>
<td>0.91200486349721</td>
<td>1.443289932e-15</td>
</tr>
<tr>
<td>0.7</td>
<td>0.8812008860741</td>
<td>0.881200886074042</td>
<td>0.8812008860741</td>
<td>5.773159728e-15</td>
</tr>
<tr>
<td>0.8</td>
<td>0.84628735275048</td>
<td>0.8462873527504811</td>
<td>0.84628735275048</td>
<td>1.110223025e-15</td>
</tr>
<tr>
<td>0.9</td>
<td>0.80752379812255</td>
<td>0.8075237981225438</td>
<td>0.80752379812255</td>
<td>6.217248938e-15</td>
</tr>
<tr>
<td>1.0</td>
<td>0.76519768655797</td>
<td>0.7651976865579627</td>
<td>0.76519768655797</td>
<td>7.327471963e-15</td>
</tr>
</tbody>
</table>

Fig. 1 Graphical solutions of Example 1

Fig. 2 Comparison of the absolute error for numerical results for Example1 for the presented method with Bernstein-polynomials presented in $^{[14]}$. 
Table 2. Comparison of the approximate solution for Example 2 of the given approach with $M = 3$, $N = 3$, Hybrid block method $^{[17]}$, Bernstein polynomials method $^{[18]}$, and Bernoulli polynomials $^{[19]}$.

<table>
<thead>
<tr>
<th>$x$</th>
<th>Hybrid block method $^{[17]}$</th>
<th>Bernstein polynomials $^{[18]}$</th>
<th>Bernoulli polynomials $^{[19]}$</th>
<th>Presented Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.0049999979</td>
<td>0.0049999958</td>
<td>0.005</td>
<td>0.004999958341723</td>
</tr>
<tr>
<td>0.2</td>
<td>0.019998667</td>
<td>0.019998667</td>
<td>0.02</td>
<td>0.0199986668419935</td>
</tr>
<tr>
<td>0.3</td>
<td>0.044998481</td>
<td>0.044989879</td>
<td>0.045</td>
<td>0.0449898794745896</td>
</tr>
<tr>
<td>0.4</td>
<td>0.079991467</td>
<td>0.079957378</td>
<td>0.08</td>
<td>0.0799573779857994</td>
</tr>
<tr>
<td>0.5</td>
<td>0.124967454</td>
<td>0.124870058</td>
<td>0.125</td>
<td>0.1248700575229549</td>
</tr>
<tr>
<td>0.6</td>
<td>0.179902837</td>
<td>0.179677141</td>
<td>0.18</td>
<td>0.179677141245484</td>
</tr>
<tr>
<td>0.7</td>
<td>0.244755068</td>
<td>0.244303617</td>
<td>0.245</td>
<td>0.244303616982151</td>
</tr>
<tr>
<td>0.8</td>
<td>0.319454501</td>
<td>0.318646009</td>
<td>0.32</td>
<td>0.318646009301246</td>
</tr>
<tr>
<td>0.9</td>
<td>0.403894871</td>
<td>0.402568621</td>
<td>0.405</td>
<td>0.402568620552525</td>
</tr>
<tr>
<td>1.0</td>
<td>0.497922483</td>
<td>0.495900383</td>
<td>0.5</td>
<td>0.495900382783151</td>
</tr>
</tbody>
</table>

Table 3. Comparison of the absolute error for numerical results for Example 3 for the presented method with $M = 3$, $N = 3$, with other three numerical methods presented in $^{[20, 21, 22]}$.

<table>
<thead>
<tr>
<th>$x$</th>
<th>Exact solution</th>
<th>Linear Multistep Hybrid Method $^{[20]}$</th>
<th>Bernstein polynomials $^{[21]}$</th>
<th>Hybrid Block method $^{[22]}$</th>
<th>Presented method</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>1.0000000000</td>
<td>$1.00 \times 10^{-10}$</td>
<td>$7.42 \times 10^{-17}$</td>
<td>$6.43000E-08$</td>
<td>9.7144514655 $\times 10^{-17}$</td>
</tr>
<tr>
<td>0.2</td>
<td>0.995321160</td>
<td>$3.00 \times 10^{-10}$</td>
<td>$3.01 \times 10^{-16}$</td>
<td>$2.72000E-08$</td>
<td>9.1593399532 $\times 10^{-17}$</td>
</tr>
<tr>
<td>0.3</td>
<td>0.982476904</td>
<td>$7.00 \times 10^{-10}$</td>
<td>$6.11 \times 10^{-16}$</td>
<td>$3.05000E-08$</td>
<td>8.6042284408 $\times 10^{-17}$</td>
</tr>
<tr>
<td>0.4</td>
<td>0.963063687</td>
<td>$7.00 \times 10^{-10}$</td>
<td>$9.48 \times 10^{-16}$</td>
<td>$8.98000E-08$</td>
<td>8.0491169285 $\times 10^{-17}$</td>
</tr>
<tr>
<td>0.5</td>
<td>0.938448064</td>
<td>$6.00 \times 10^{-10}$</td>
<td>$1.26 \times 10^{-15}$</td>
<td>$4.42600E-07$</td>
<td>7.4940054162 $\times 10^{-17}$</td>
</tr>
<tr>
<td>0.6</td>
<td>0.909795990</td>
<td>$2.00 \times 10^{-10}$</td>
<td>$1.42 \times 10^{-15}$</td>
<td>$7.72600E-07$</td>
<td>6.9388939039 $\times 10^{-17}$</td>
</tr>
<tr>
<td>0.7</td>
<td>0.878098618</td>
<td>$9.00 \times 10^{-10}$</td>
<td>$1.11 \times 10^{-16}$</td>
<td>$1.95236E-06$</td>
<td>6.3837823916 $\times 10^{-17}$</td>
</tr>
<tr>
<td>0.8</td>
<td>0.844195016</td>
<td>$2.80 \times 10^{-9}$</td>
<td>$5.78 \times 10^{-16}$</td>
<td>$1.02746E-06$</td>
<td>5.8286708793 $\times 10^{-17}$</td>
</tr>
<tr>
<td>0.9</td>
<td>0.808792135</td>
<td>$5.40 \times 10^{-9}$</td>
<td>$5.93 \times 10^{-15}$</td>
<td>$1.35096E-06$</td>
<td>5.2735593670 $\times 10^{-17}$</td>
</tr>
<tr>
<td>1.0</td>
<td>0.772482354</td>
<td>$3.50 \times 10^{-9}$</td>
<td>$2.02 \times 10^{-14}$</td>
<td>$1.34706E-05$</td>
<td>4.47184478547 $\times 10^{-17}$</td>
</tr>
</tbody>
</table>
Fig. 3 Comparison of the absolute error for numerical results for Example 3 for the presented method with other three numerical methods presented in [20, 21, 22].

Table 4. Comparison of the absolute error for numerical results for Example 4 for the presented method with $M = 3$, $N = 3$. With other three numerical methods presented in [22, 23].

<table>
<thead>
<tr>
<th>$x$</th>
<th>Exact solution</th>
<th>Multi derivative Hybrid block methods [23]</th>
<th>Hybrid Block method [22]</th>
<th>Presented method</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>0.999950000</td>
<td>1.60168E-05</td>
<td>6.72000E-07</td>
<td>2.305072e-13</td>
</tr>
<tr>
<td>0.02</td>
<td>0.999800007</td>
<td>1.100991E-04</td>
<td>1.34410E-06</td>
<td>2.279087e-13</td>
</tr>
<tr>
<td>0.03</td>
<td>0.999550034</td>
<td>5.567153E-04</td>
<td>2.01700E-06</td>
<td>1.351249e-13</td>
</tr>
<tr>
<td>0.04</td>
<td>0.999200107</td>
<td>1.6332403E-03</td>
<td>2.68840E-06</td>
<td>5.823823e-14</td>
</tr>
<tr>
<td>0.05</td>
<td>0.998750260</td>
<td>3.62018361E-03</td>
<td>3.35940E-06</td>
<td>4.585688e-13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x$</th>
<th>Exact solution</th>
<th>Absolute Error</th>
<th>Relative error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Multi derivative Hybrid block methods [23]</td>
<td>Hybrid Block method [22]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Presented method</td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>0.999950000</td>
<td>1.601682E-05</td>
<td>6.72000E-07</td>
</tr>
<tr>
<td>0.02</td>
<td>0.999800007</td>
<td>1.100991E-04</td>
<td>1.34410E-06</td>
</tr>
<tr>
<td>0.03</td>
<td>0.999550034</td>
<td>5.567153E-04</td>
<td>2.01700E-06</td>
</tr>
<tr>
<td>0.04</td>
<td>0.999200107</td>
<td>1.6332403E-03</td>
<td>2.68840E-06</td>
</tr>
<tr>
<td>0.05</td>
<td>0.998750260</td>
<td>3.62018361E-03</td>
<td>3.35940E-06</td>
</tr>
</tbody>
</table>
Fig. 4 Comparison of the absolute error for numerical results for Example 4 for the presented method with $M = 3$, $N = 3$. With other three numerical methods presented in \[^{22, 23}\].

4. Conclusion

To address several second and third order initial value issues of ODEs, the hybrid Bernstein and enhanced block-pulse functions method introduced by Ramadan and Osheba \[^9\] is employed and developed. The current method transforms differential equations into an algebraic system that can be solved using standard methods. The numerical results show that the derived method outperforms some of the other methods addressed in this paper. Based on the numerical results, the method appears to be very promising for managing more general equations that the authors are investigating, such as nonlinear differential equations and oscillator differential equations, which play critical roles in natural and physical simulations.

5. References


