

Article

A Block Hybrid Method with Equally Spaced Grid Points for Third-Order Initial Value Problems

Salma A. A. Ahmedai Abd Allah , Precious Sibanda , Sicelo P. Goqo *, Uthman O. Rufai ,
Hloniphile Sithole Mthethwa  and Osman A. I. Noreldin 

School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Pietermaritzburg 3209, South Africa; 219088479@stu.ukzn.ac.za (S.A.A.A.A.); sibandap@ukzn.ac.za (P.S.); 219094325@stu.ukzn.ac.za (U.O.R.); sitholeh@ukzn.ac.za (H.S.M.); osman@aims.edu.gh (O.A.I.N.)

* Correspondence: goqos@ukzn.ac.za

Abstract: In this paper, we extend the block hybrid method with equally spaced intra-step points to solve linear and nonlinear third-order initial value problems. The proposed block hybrid method uses a simple iteration scheme to linearize the equations. Numerical experimentation demonstrates that equally spaced grid points for the block hybrid method enhance its speed of convergence and accuracy compared to other conventional block hybrid methods in the literature. This improvement is attributed to the linearization process, which avoids the use of derivatives. Further, the block hybrid method is consistent, stable, and gives rapid convergence to the solutions. We show that the simple iteration method, when combined with the block hybrid method, exhibits impressive convergence characteristics while preserving computational efficiency. In this study, we also implement the proposed method to solve the nonlinear Jerk equation, producing comparable results with other methods used in the literature.

Keywords: block hybrid method; equally spaced grid points; differential equations; nonlinear jerk equation; simple iteration method

MSC: 65L05



Citation: Ahmedai Abd Allah, S.A.A.; Sibanda, P.; Goqo, S.P.; Rufai, U.O.; Sithole Mthethwa, H.; Noreldin, O.A.I. A Block Hybrid Method with Equally Spaced Grid Points for Third-Order Initial Value Problems. *AppliedMath* **2024**, *4*, 320–347. <https://doi.org/10.3390/appliedmath4010017>

Academic Editors: Jai Prakash Jaiswal, Juan Ramón Torregrosa Sánchez and Carlo Bianca

Received: 23 November 2023

Revised: 5 February 2024

Accepted: 19 February 2024

Published: 1 March 2024



Copyright: © 2024 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

Differential equations with an appropriate set of conditions are key tools in modeling a plethora of real-world phenomena including engineering problems. It is crucial to have a clear understanding of the behavior and stability of solutions to these differential equations. Many numerical methods have been introduced to find an approximate solution for those cases. In this study, we propose a block hybrid method for the numerical solution of third-order initial value problems (IVPs) of the form

$$u''' = f(z, u, u', u''), \quad z \in [a, b], \quad (1)$$

$$u''(a) = u''_0, \quad u'(a) = u'_0, \quad u(a) = u_0, \quad (2)$$

where the prime denotes the derivative with respect to the independent variable z , and f is a continuous linear or nonlinear function in the interval $[a, b]$.

Block hybrid algorithms are a class of numerical methods that blend linear multi-step approaches with interpolation using power series. These techniques were first introduced by Gragg and Stetter [1] and Shampine and Watts [2] and involve the inclusion of an additional point within each step of the formula. This allows for more precise approximations of solutions to differential equations, as well as gives better convergence rates. Since the pioneering work of Gear [3], block methods have been extensively studied and used in the literature to solve initial value problems and boundary value problems. Motsa [4] proposed an overlapping grid hybrid block method with equally spaced and optimal grid points

for linear and nonlinear first-order IVPs. He found that the overlapping grid approach gave better performance than the standard non-overlapping grid in terms of reducing the local truncation error. Shateyi [5] applied a block hybrid method with equally spaced grid points to solve linear and nonlinear first-order IVPs. He reported that equally spaced grid points provided high rates of convergence outperforming the fourth-order Runge–Kutta method [6]. Orakwelu [7] developed an implicit block hybrid method for solving first-, second-, and third-order IVPs. He investigated the convergence rates, accuracy, and robustness of implicit block hybrid algorithms. He further investigated the performance of these algorithms when various countable off-points were imposed between grid points in the derivation process. El-Hawary and Mahmoud [8] presented the spline functions with four collocation points to solve second-order IVPs. They showed that the spline collocation scheme is convergent with order seven under certain conditions on the collocation point parameters. They found that the stability properties of the method are analyzed rigorously, and regions of absolute stability are determined based on the parameter values.

Various iterative techniques have been discussed for solving initial value and boundary value problems in ordinary and partial differential equations [9,10]. These iterative methods yield the solution or an approximation of it through a sequence of successive iterations. In the case of initial value problems, these iterative approaches can be formulated either in an integral or differential manner [11]. These approaches linearize the nonlinear equations governing the problem around the previous iteration, leading to linear differential equations at each step [12]. However, the coefficients of these equations may vary with the independent variable, necessitating numerical methods for obtaining the approximation solution.

In this study, we are motivated by the work in [4] and propose a block hybrid method for third-order IVPs with equally spaced grid points. Several strategies have been developed in the literature to solve third-order IVPs. For example, Osa et al. [13] used a multi-step block implicit hybrid method. They reported that the block method performed better than predictor–corrector methods [14] in terms of being more time-efficient, cost-effective, and accurate. Rufai and Ramos [15] modified the block method using variable step sizes instead of equally spaced points. They reported that the technique was efficient in terms of computational time and minimized truncation errors. Orakwelu et al. [16] proposed a single-step hybrid block method for solving third-order ODEs without first converting to an analogous first-order system. They reported that the scheme can be implemented without the use of starting values or predictors, avoiding the necessity for complex subroutines method. Several other authors have utilized the equal step size procedure for solving the third-order IVPs [17–19].

Block hybrid methods, used to solve differential equations, are primarily applicable to linear equations. However, when dealing with nonlinear equations, the method necessitates a preliminary step of linearization before its implementation. This prerequisite ensures the effectiveness and accuracy of methods when applied to nonlinear equations. In this study, we developed an intra-step equally spaced grid point block hybrid method that is used in conjunction with the simple iteration method (SIM) to solve third-order IVPs. They demonstrated that the SIM linearization technique generates accurate solutions for nonlinear differential equations [20]. The SIM approach is based on transforming a nonlinear ordinary differential equation into an iterative scheme made up of linear equations, which are then solved using a block hybrid method numerical approach. Linearization methods based on truncated Taylor series approximations are employed to simplify the nonlinear terms of nonlinear differential equations. Relaxation methods based on the assumption that nonlinear terms are known from previous iterations can also be used to convert a nonlinear problem to a linear discretizable problem. Newton-based linearization techniques such as the quasi-linearization method [21], local linearization method [22], Keller-box method [23], and relaxation method [24], have extensively been used to linearize differential equations. All these methods are based on one-term Taylor series expansion and are thereby susceptible to series truncation errors even before errors associated with the numerical method

used to solve the linearised problem. In this paper, we propose a method that seeks to circumvent the problem of errors associated with linearization. The proposed method uses ideas akin to those of fixed point iteration to develop iterative schemes, called Simple Iteration Methods (SIMs), for solving nonlinear differential equations. Moreover, SIMs are easy to implement. We show the effectiveness of the proposed method (referred to as the SIM-BHM) through numerical experiments, demonstrating that it gives fast convergence and accurate solutions.

2. Development of the SIM-BHM Iterative Method

This section describes the development of the SIM-BHM with equally spaced points for the numerical solution to third-order IVPs (1) and (2). Before starting the development of the hybrid block method, we need to linearize the function f using the SIM scheme as below:

2.1. The SIM Linearization Scheme

In this section, we present the development of the SIM. If the nonlinear equation $f = 0$ is rearranged in the form $z = g(z)$, then an iterative method can be written as

$$z^{s+1} = g^s, \quad s = 0, 1, 2, 3, \dots \tag{3}$$

where $g^s = g(z^s)$, and s is the number of iterative steps. This method is called fixed point iteration, the successive substitution method, or the simple iteration method. The simple iteration method is touted as a differential equation equivalent to the fixed point iteration method for solving nonlinear equations. Thus, we begin by reviewing some basic concepts of the fixed point iteration method. The following theorem gives the conditions of the fixed point iteration method.

Theorem 1. (Fixed Point Theorem).

- If g is continuous on a closed interval $[a, b]$, for all $x \in [a, b]$, then g has at least one fixed point in $[a, b]$.
- If, in addition, $g'(z)$ exists on (a, b) and a positive constant K exists with

$$|g'(z)| \leq K < 1, \text{ for all } z \in (a, b),$$

then there is exactly one fixed point in $[a, b]$.

Assume that c in (a, b) is a fixed point for g . Then, if there is an initial guess in (a, b) , the sequence

$$z^{s+1} = g^s, \quad s = 0, 1, 2, \dots$$

converges to the unique fixed point c . For proof of **Theorem 1**, see [25].

Simple Iteration Method for the ODE

The approach proposed below is a systematic approach based on a rule used to simplify products of unknown functions and their derivatives. We assume that

$$f = f(z, u, \delta, \gamma), \tag{4}$$

is a nonlinear function in u, δ , and γ . To develop the iteration scheme, Equation (4) is expressed as a sum of its linear as nonlinear components as

$$f = \mathcal{L}^*(z, u, \delta, \gamma) + \mathcal{N}^*(z, u, \delta, \gamma), \tag{5}$$

The SIM linearization scheme is developed so that the linear component \mathcal{L}^* can be written as the sum of linear terms (unknown functions) at iteration $s + 1$

$$\mathcal{L}^* = \mathcal{L}_2^*(z^s)\gamma^{s+1} + \mathcal{L}_1^*(z^s)\delta^{s+1} + \mathcal{L}_0^*(z^s)u^{s+1} \tag{6}$$

The nonlinear terms are a combination of known (at iteration s) and unknown =functions. In the nonlinear term, the function with a higher derivative is taken as unknown. Provided that the nonlinear component \mathcal{N}^* can be written as the sum of nonlinear terms

$$\mathcal{N}^* = \mathcal{N}_2(z^s, u^s, \delta^s, \gamma^s)\gamma^{s+1} + \mathcal{N}_1(z^s, u^s, \delta^s)\delta^{s+1} + \mathcal{N}_0(z^s, u^s)u^{s+1} + \mathcal{N}(z^s) \quad (7)$$

Substituting Equations (6) and (7) into Equation (5), we obtain

$$f = \mathcal{L}_2(z^s, u^s, \delta^s, \gamma^s)\gamma^{s+1} + \mathcal{L}_1(z^s, u^s, \delta^s)\delta^{s+1} + \mathcal{L}_0(z^s, u^s)u^{s+1} + \mathcal{N}(z^s), \quad (8)$$

where $\mathcal{L}_2, \mathcal{L}_1, \mathcal{L}_0$, and \mathcal{N} are known functions from a previous iteration given by

$$\begin{aligned} \mathcal{L}_2 &= \mathcal{L}_2^*(z^s) + \mathcal{N}_2(z^s, u^s, \delta^s, \gamma^s), \\ \mathcal{L}_1 &= \mathcal{L}_1^*(z^s) + \mathcal{N}_1(z^s, u^s, \delta^s), \\ \mathcal{L}_0 &= \mathcal{L}_0^*(z^s) + \mathcal{N}_0(z^s, u^s). \end{aligned}$$

Equation (8) is the linearized form of f .

The objective of this paper is to develop an alternative and novel block hybrid method approach that is based on an equally spaced grid (see Figure 1).

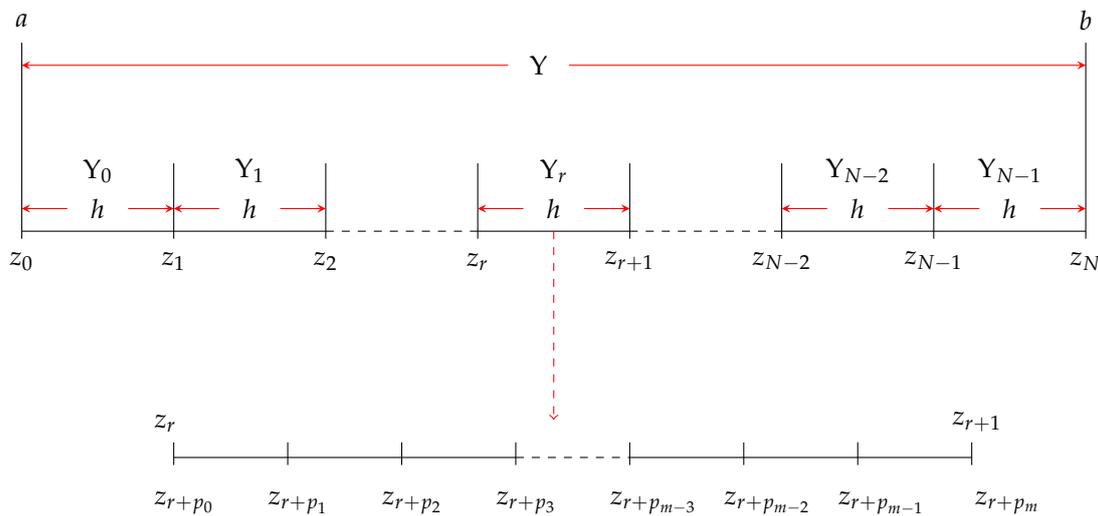


Figure 1. Illustration of the equally spaced grid points.

We define the block by Y_r and the fixed step length or size is denoted by h , where

$$h = z_{r+1} - z_r, \quad r = 0, 1, 0, \dots, N - 1, \quad (9)$$

where N is the number of blocks in $[a, b]$. For each block $Y_r = [z_r, z_{r+1}]$, we define $m + 1$ intra-step points p_j as

$$p_j = \frac{j}{m}, \quad j = 0, 1, 2, 3, \dots, m, \quad (10)$$

$$z_{r+p_j} = z_r + p_j h. \quad (11)$$

In using the BHM, we approximate the solution $u(z)$ to Equation (1) using a polynomial of the form

$$u(z) \approx U(z) = \sum_{k=0}^{m+3} a_{r,k} (z - z_r)^k. \quad (12)$$

In addition, we approximate the first and second derivatives as

$$\delta(z) \approx U'(z) = \sum_{k=1}^{m+3} a_{r,k} k(z - z_r)^{k-1}, \tag{13}$$

$$\gamma(z) \approx U''(z) = \sum_{k=2}^{m+3} a_{r,k} k(k-1)(z - z_r)^{k-2}, \tag{14}$$

respectively, where $a_{r,k}$ are $m + 4$ unknown coefficients. We apply collocation at the unknown nodes z_{r+p_j} and in Equations (10) and (11). We assume that the approximated solutions (12)–(14) satisfy Equations (1) and (2). The coefficients $a_{r,k}$ are obtained from a system of $m + 4$ equations with $m + 4$ unknowns generated from

$$U'''(z_{r+p_j}) = f_{r+p_j}, \quad j = 0, 1, 2, \dots, m. \tag{15}$$

$$U(z_r) = a_{r,0} = u_r, \quad r = 0, 1, \dots, N - 1. \tag{16}$$

$$U'(z_r) = a_{r,1} = \delta_r, \quad r = 0, 1, \dots, N - 1. \tag{17}$$

$$U''(z_r) = 2 a_{r,2} = \gamma_r, \quad r = 0, 1, \dots, N - 1. \tag{18}$$

where $f_{r+p_j} = f(z_{r+p_j}, u_{r+p_j}, \delta_{r+p_j}, \gamma_{r+p_j})$, $f_r = f(z_r, u_r, \delta_r, \gamma_r)$, $u_r = u(z_r)$, $\delta_r = \delta(z_r)$, and $\gamma_r = \gamma(z_r)$. The above system of Equations (15)–(18) can be written in matrix notation as

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 6 & 24hp_1 & 60h^2p_1^2 & 120h^3p_1^3 & \dots & (m+3)(m+2)(m+1)h^m p_1^m \\ 0 & 0 & 0 & 6 & 24hp_2 & 60h^2p_2^2 & 120h^3p_2^3 & \dots & (m+3)(m+2)(m+1)h^m p_2^m \\ 0 & 0 & 0 & 6 & 24hp_3 & 60h^2p_3^2 & 120h^3p_3^3 & \dots & (m+3)(m+2)(m+1)h^m p_3^m \\ \vdots & \dots & \vdots \\ 0 & 0 & 0 & 6 & 24h & 60h^2 & 120h^3 & \dots & (m+3)(m+2)(m+1)h^m \end{bmatrix} \begin{bmatrix} a_{r,0} \\ a_{r,1} \\ a_{r,2} \\ a_{r,3} \\ a_{r,4} \\ a_{r,5} \\ a_{r,6} \\ \vdots \\ a_{r,m+3} \end{bmatrix} = \begin{bmatrix} u_r \\ \delta_r \\ \gamma_r \\ f_r \\ f_{r+p_1} \\ f_{r+p_2} \\ f_{r+p_3} \\ \vdots \\ f_{r+1} \end{bmatrix} \tag{19}$$

We solve the above system of Equation (19) to obtain $a_{r,k}$. We use Mathematica code to obtain $a_{r,k}$. The code is as follows

```

p = Table [p_j = j/m, {j, 0, m}];
m = Length[p] - 1;
Table[z_{r+points[[j]]} = (r + p[[j]]) * h, {i, m + 1}];
U = sum[a_k * (z - z_r)^k, {k, 0, m + 3}];
dU = sum[a_k * k * (z - z_r)^{k-1}, {k, 1, m + 3}];
ddU = sum[a_k * k * (k - 1) * (z - z_r)^{k-2}, {k, 2, m + 3}];
dddU = sum[a_k * k * (k - 1) * (k - 2) * (z - z_r)^{k-3}, {k, 3, m + 3}];
initial1 = (U /. z -> z_r) == u_r;
initial2 = (dU /. z -> z_r) == delta_r;
initial3 = (ddU /. z -> z_r) == gamma_r;
Nequations = Table[(dddU /. z -> z_{r+p[[j]])} == f_{r+p[[j]]}, {j, 1, m + 1}];
Nallequations = Join[{initial1}, {initial2}, {initial3}, Nequations] // Simplify.
    
```

Substituting $a_{r,k}$ into Equations (12)–(14) and evaluating the result at collocation points (10) and (11) yield

$$u_{r+p_i} = u_r + h p_i \delta_r + \frac{1}{2} h^2 p_i^2 \gamma_r + h^3 \sum_{j=0}^m \alpha_{i,j} f_{r+p_j}, \tag{20}$$

$$\delta_{r+p_i} = \delta_r + h p_i \gamma_r + h^2 \sum_{j=0}^m \mu_{i,j} f_{r+p_j}, \tag{21}$$

$$\gamma_{r+p_i} = \gamma_r + h \sum_{j=0}^m \kappa_{i,j} f_{r+p_j}, \tag{22}$$

where $\alpha_{i,j}$, $\mu_{i,j}$, and $\kappa_{i,j}$ are known constants. By substituting Equation (8) into the BHM scheme system of Equations (20)–(22), we obtain a system of equations in matrix form

$$\mathbf{D}_{11} \mathbf{u}_{r+p}^{s+1} + \mathbf{D}_{12} \delta_{r+p}^{s+1} + \mathbf{D}_{13} \gamma_{r+p}^{s+1} = \mathbf{R}_1, \tag{23}$$

$$\mathbf{D}_{21} \mathbf{u}_{r+p}^{s+1} + \mathbf{D}_{22} \delta_{r+p}^{s+1} + \mathbf{D}_{23} \gamma_{r+p}^{s+1} = \mathbf{R}_2, \tag{24}$$

$$\mathbf{D}_{31} \mathbf{u}_{r+p}^{s+1} + \mathbf{D}_{32} \delta_{r+p}^{s+1} + \mathbf{D}_{33} \gamma_{r+p}^{s+1} = \mathbf{R}_3, \tag{25}$$

where

$$\mathbf{D}_{11} = \mathbf{I}_m - h^3 \mathbf{A} \mathbf{L}_{0,r+p}, \quad \mathbf{D}_{12} = -h^3 \mathbf{A} \mathbf{L}_{1,r+p}, \quad \mathbf{D}_{13} = -h^3 \mathbf{A} \mathbf{L}_{2,r+p},$$

$$\mathbf{D}_{21} = -h^2 \mathbf{B} \mathbf{L}_{0,r+p}, \quad \mathbf{D}_{22} = \mathbf{I}_m - h^2 \mathbf{B} \mathbf{L}_{1,r+p}, \quad \mathbf{D}_{23} = -h^2 \mathbf{B} \mathbf{L}_{2,r+p},$$

$$\mathbf{D}_{31} = -h \mathbf{C} \mathbf{L}_{0,r+p}, \quad \mathbf{D}_{32} = -h \mathbf{C} \mathbf{L}_{1,r+p}, \quad \mathbf{D}_{33} = \mathbf{I}_m - h \mathbf{C} \mathbf{L}_{2,r+p},$$

$$\mathbf{R}_1 = \mathbf{u}_r + h \mathbf{P} \delta_r + \frac{1}{2} h^2 \mathbf{P}^2 \gamma_r + h^3 \mathbf{A}_0 \mathbf{F}_r + h^3 \mathbf{A} \mathbf{N}_r,$$

$$\mathbf{R}_2 = \delta_r + h \mathbf{P} \gamma_r + h^2 \mathbf{B}_0 \mathbf{F}_r + h^2 \mathbf{B} \mathbf{N}_r,$$

$$\mathbf{R}_3 = \gamma_r + h \mathbf{C}_0 \mathbf{F}_r + h \mathbf{C} \mathbf{N}_r.$$

In Equations (23)–(25), I_m is an identity matrix of size $(m \times m)$, and all other matrices of size $(m \times m)$ are defined as follows

$$A = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,m} \\ \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{m,1} & \alpha_{m,2} & \cdots & \alpha_{m,m} \end{bmatrix}, \quad B = \begin{bmatrix} \mu_{1,1} & \mu_{1,2} & \cdots & \mu_{1,m} \\ \mu_{2,1} & \mu_{2,2} & \cdots & \mu_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{m,1} & \mu_{m,2} & \cdots & \mu_{m,m} \end{bmatrix}$$

$$C = \begin{bmatrix} \kappa_{1,1} & \kappa_{1,2} & \cdots & \kappa_{1,m} \\ \kappa_{2,1} & \kappa_{2,2} & \cdots & \kappa_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \kappa_{m,1} & \kappa_{m,2} & \cdots & \kappa_{m,m} \end{bmatrix}, \quad A_0 = \begin{bmatrix} \alpha_{1,0} & 0 & \cdots & 0 \\ 0 & \alpha_{2,0} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_{m,0} \end{bmatrix},$$

$$B_0 = \begin{bmatrix} \mu_{1,0} & 0 & \cdots & 0 \\ 0 & \mu_{2,0} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mu_{m,0} \end{bmatrix}, \quad C_0 = \begin{bmatrix} \kappa_{1,0} & 0 & \cdots & 0 \\ 0 & \kappa_{2,0} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \kappa_{m,0} \end{bmatrix},$$

$$L_{2,r+p} = \begin{bmatrix} \mathcal{L}_2(z_{r+p_1}^s, u_{r+p_1}^s, \delta_{r+p_1}^s, \gamma_{r+p_1}^s) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathcal{L}_2(z_{r+p_m}^s, u_{r+p_m}^s, \delta_{r+p_m}^s, \gamma_{r+p_m}^s) \end{bmatrix},$$

$$L_{1,r+p} = \begin{bmatrix} \mathcal{L}_1(z_{r+p_1}^s, u_{r+p_1}^s, \delta_{r+p_1}^s) & 0 & \cdots & 0 \\ 0 & \mathcal{L}_1(z_{r+p_2}^s, u_{r+p_2}^s, \delta_{r+p_2}^s) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathcal{L}_1(z_{r+p_m}^s, u_{r+p_m}^s, \delta_{r+p_m}^s) \end{bmatrix},$$

$$L_{0,r+p} = \begin{bmatrix} \mathcal{L}_0(z_{r+p_1}^s, u_{r+p_1}^s) & 0 & \cdots & 0 \\ 0 & \mathcal{L}_0(z_{r+p_2}^s, u_{r+p_2}^s) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathcal{L}_0(z_{r+p_m}^s, u_{r+p_m}^s) \end{bmatrix},$$

$$P = \begin{bmatrix} p_1 & 0 & \cdots & 0 \\ 0 & p_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & p_m \end{bmatrix}$$

whereas vectors of size $(m \times 1)$ are defined as follows

$$u_{r+p}^{s+1} = \begin{bmatrix} u_{r+p_1}^{s+1} \\ u_{r+p_2}^{s+1} \\ \vdots \\ u_{r+p_m}^{s+1} \end{bmatrix}, \quad \delta_{r+p}^{s+1} = \begin{bmatrix} \delta_{r+p_1}^{s+1} \\ \delta_{r+p_2}^{s+1} \\ \vdots \\ \delta_{r+p_m}^{s+1} \end{bmatrix}, \quad \gamma_{r+p}^{s+1} = \begin{bmatrix} \gamma_{r+p_1}^{s+1} \\ \gamma_{r+p_2}^{s+1} \\ \vdots \\ \gamma_{r+p_m}^{s+1} \end{bmatrix}, \quad u_r = \begin{bmatrix} u_r^s \\ u_r^s \\ \vdots \\ u_r^s \end{bmatrix}, \quad \delta_r = \begin{bmatrix} \delta_r^s \\ \delta_r^s \\ \vdots \\ \delta_r^s \end{bmatrix},$$

$$\gamma_r = \begin{bmatrix} \gamma_r^s \\ \gamma_r^s \\ \vdots \\ \gamma_r^s \end{bmatrix}, \quad \mathbf{F}_r = \begin{bmatrix} \mathcal{L}_2(z_r^s, u_r^s, \delta_r^s, \gamma_r^s) \gamma_r^s + \mathcal{L}_1(z_r^s, u_r^s, \delta_r^s) \delta_r^s + \mathcal{L}_0(z_r^s, u_r^s) u_r^s + \mathcal{N}(z_r^s) \\ \mathcal{L}_2(z_r^s, u_r^s, \delta_r^s, \gamma_r^s) \gamma_r^s + \mathcal{L}_1(z_r^s, u_r^s, \delta_r^s) \delta_r^s + \mathcal{L}_0(z_r^s, u_r^s) u_r^s + \mathcal{N}(z_r^s) \\ \vdots \\ \mathcal{L}_2(z_r^s, u_r^s, \delta_r^s, \gamma_r^s) \gamma_r^s + \mathcal{L}_1(z_r^s, u_r^s, \delta_r^s) \delta_r^s + \mathcal{L}_0(z_r^s, u_r^s) u_r^s + \mathcal{N}(z_r^s) \end{bmatrix},$$

$$\mathbf{N}_r = \begin{bmatrix} \mathcal{N}(z_r^s) \\ \mathcal{N}(z_r^s) \\ \vdots \\ \mathcal{N}(z_r^s) \end{bmatrix},$$

We may rewrite System (25) in compact form as follows

$$\begin{bmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} & \mathbf{D}_{13} \\ \mathbf{D}_{21} & \mathbf{D}_{22} & \mathbf{D}_{23} \\ \mathbf{D}_{31} & \mathbf{D}_{32} & \mathbf{D}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{r+p}^{s+1} \\ \delta_{r+p}^{s+1} \\ \gamma_{r+p}^{s+1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \mathbf{R}_3 \end{bmatrix}. \tag{26}$$

System (26) may now be written in the form

$$\mathbf{D}\mathbf{U} = \mathbf{R}, \tag{27}$$

where the matrix \mathbf{D} is of size $(3m \times 3m)$. The approximate solutions are found by solving Equation (27) as

$$\mathbf{U} = \mathbf{inv}(\mathbf{D})\mathbf{R}. \tag{28}$$

where \mathbf{inv} is the inverse of the matrix.

3. Analysis of the Developed SIM-BHM

This section provides the fundamental properties of the proposed SIM-BHM.

3.1. Order and Error Constant

Minimizing the truncation error is a fundamental goal in numerical analysis utilized as a measure to improve the accuracy of the method.

Proposition 1. *In the integration block $[z_r, z_{r+1}]$, for the block hybrid method defined as*

$$u_{r+p_i} = u_r + h p_i \delta_r + \frac{1}{2} h^2 p_i^2 \gamma_r + h^3 \sum_{j=0}^m \alpha_{i,j} f_{r+p_j}, \tag{29}$$

the local truncation error is given by

$$\mathfrak{L}_i[u(z_r); h] = \frac{h^{m+4}}{(m+4)!} \left[p_i^{m+4} - (m+4)(m+3)(m+2) \sum_{j=1}^m \alpha_{i,j} p_i^{m+1} \right] u^{(m+4)}(z_r) + O(h^{m+5}). \tag{30}$$

Proof. Given that u is a sufficiently differentiable function, the local truncation error can be analyzed using the linear operator \mathfrak{L} defined as

$$\mathfrak{L}_i[u(z_r); h] = C_{i,0}u(z_r) + C_{i,1}hu'(z_r) + C_{i,2}h^2u''(z_r) + C_{i,3}h^3u'''(z_r) + \dots + C_{i,q}h^q u^{(q)}(z_r), \tag{31}$$

where $C_{i,0}, C_{i,1}, \dots, C_{i,q}$ are constant. The method is said to be order q if

$$\hat{C}_0 = \hat{C}_1 = \dots = \hat{C}_{q+2} = 0, \quad \text{and} \quad \hat{C}_{q+3} \neq 0, \tag{32}$$

where $\hat{C}_q = [C_{1,q}, C_{2,q}, \dots, C_{m,q}]^T$. The vector $\hat{C}_{q+3} = [C_{1,q+3}, C_{2,q+3}, \dots, C_{m,q+3}]^T$ is the error constant of the method. Using Equation (29), the truncation error of the BHM for third-order can be written in terms of the linear operator as

$$\mathfrak{L}_i[u(z_r); h] = u(z_r + hp_i) - u(z_r) - hp_i u'(z_r) - \frac{1}{2} h^2 p_i^2 u''(z_r) - h^3 \sum_{j=0}^m \alpha_{ij} u'''(z_r + p_j h), \tag{33}$$

Expanding the terms $u(z_r + hp_i)$ and $u'''(z_r + hp_i)$ using Taylor series about z_r and then substituting them into Equation (33), we obtain

$$\mathfrak{L}_i[u(z_r); h] = \sum_{k=1}^K \frac{h^{k+2} p_i^{k+2}}{(k+2)!} u^{(k+2)}(z_r) - \sum_{k=1}^K \frac{kh^{k+2}}{k!} \sum_{j=0}^m \alpha_{ij} p_j^{k-1} u^{(k+2)}(z_r) + O(h^{K+1}), \tag{34}$$

Expanding Equation (34), we obtain

$$\begin{aligned} \mathfrak{L}_i[u(z_r); h] &= \sum_{k=1}^{m+1} \frac{h^{k+2} p_i^{k+2}}{(k+2)!} u^{(k+2)}(z_r) - \sum_{k=1}^{m+1} \frac{kh^{k+2}}{k!} \sum_{j=0}^m \alpha_{ij} p_j^{k-1} u^{(k+2)}(z_r) \\ &+ \sum_{k=m+2}^K \frac{h^{k+2} p_i^{k+2}}{(k+2)!} u^{(k+2)}(z_r) - \sum_{k=m+2}^K \frac{kh^{k+2}}{k!} \sum_{j=0}^m \alpha_{ij} p_j^{k-1} u^{(k+2)}(z_r) + O(h^{K+1}), \end{aligned} \tag{35}$$

where m is a positive integer. This may be written as

$$\begin{aligned} \mathfrak{L}_i[u(z_r); h] &= \sum_{k=3}^{m+1} \frac{h^k}{k!} \left[p_i^k - k(k-1)(k-2) \sum_{j=0}^m \alpha_{ij} p_j^{k-3} \right] u^{(k)}(z_r) \\ &+ \sum_{k=m+4}^K \frac{h^k}{k!} \left[p_i^k - k(k-1)(k-2) \sum_{j=0}^m \alpha_{ij} p_j^{k-3} \right] u^{(k)}(z_r) + O(h^{K+1}), \end{aligned} \tag{36}$$

Following [4,26], we note that

$$\sum_{j=0}^m \alpha_{ij} p_j^{k-3} = \frac{p_i^k}{k(k-1)(k-2)}, \quad \text{for } k = 4, \dots, m+1. \tag{37}$$

Substituting Equation (37) into Equation (36), we obtain

$$\mathfrak{L}_i[u(z_r); h] = \frac{h^{m+4}}{(m+4)!} \left[p_i^{m+4} - (m+4)(m+3)(m+2) \sum_{j=1}^m \alpha_{ij} p_i^{m+1} \right] u^{(m+4)}(z_r) + O(h^{m+5}). \tag{38}$$

Using Equation (38), the error constant vector is

$$\hat{C}_{m+4} = [C_{1,m+4}, C_{2,m+4}, \dots, C_{m,m+4}]^T, \tag{39}$$

where

$$C_{i,m+4} = \frac{p_i^{m+4}}{(m+4)!} - \frac{1}{(m+1)!} \sum_{j=1}^m \alpha_{ij} p_i^{m+1}. \tag{40}$$

□

Table 1 shows the truncation error and order for $m = 2, 3, 4, 5, 6$. In the table, we note that increasing the number of intra-step points improves the order of the method.

Table 1. Truncation errors for the SIM-BHM.

m	p_i	Truncation Error	q
2	$\frac{1}{2}, 1$	$\frac{u^{(6)}z_r h^6}{9216}, \frac{u^{(6)}z_r h^6}{1440}$	3
3	$\frac{1}{3}, \frac{2}{3}, 1$	$-\frac{47u^{(7)}z_r h^7}{22044960}, -\frac{19u^{(7)}z_r h^7}{1377810}, -\frac{u^{(7)}z_r h^7}{30240}$	4
4	$\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1$	$\frac{139u^{(8)}z_r h^8}{2642411520}, \frac{u^{(8)}z_r h^8}{2949120}, \frac{243u^{(8)}z_r h^8}{293601280}, \frac{u^{(8)}z_r h^8}{645120}$	5
5	$\frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1$	$-\frac{9809u^{(9)}z_r h^9}{7087500000000}, -\frac{491u^{(9)}z_r h^9}{55371093750}, -\frac{1917u^{(9)}z_r h^9}{87500000000}, -\frac{1136u^{(9)}z_r h^9}{27685546875}, -\frac{149u^{(9)}z_r h^9}{2268000000}$	6
6	$\frac{1}{6}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{5}{6}, 1$	$\frac{4001u^{(10)}z_r h^{10}}{109709829734400}, \frac{199u^{(10)}z_r h^{10}}{857108044800}, \frac{29u^{(10)}z_r h^{10}}{267846264400}, \frac{29u^{(10)}z_r h^{10}}{26784626400}, \frac{7625u^{(10)}z_r h^{10}}{4388393189376}, \frac{u^{(10)}z_r h^{10}}{391910400}$	7

3.2. Zero Stability

The zero stability of the SIM-BHM pertains to the performance of Equation (20) as the step size h approaches zero [27].

Definition 1. Zero stability is confirmed if the roots of the characteristic polynomial $\rho(\lambda)$ are such that $|\lambda_j| \leq 1$ and all roots with $|\lambda_j| = 1$ have a multiplicity that does not exceed 2.

Using Equations (23)–(25), as $h \rightarrow 0$, the matrix system reduces to

$$\mathbf{I}_m \mathbf{u}_{r+p} - \mathbf{A}_1 \mathbf{u}_{r-1+p} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} u_{r+p_1} \\ u_{r+p_2} \\ \vdots \\ u_{r+p_m} \end{bmatrix} - \begin{bmatrix} 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} u_{r-1+p_1} \\ u_{r-1+p_2} \\ \vdots \\ u_{r-1+p_m} \end{bmatrix},$$

The characteristic polynomial $\rho(\lambda)$ is given by

$$\rho(\lambda) = \det(\lambda \mathbf{I}_m - \mathbf{A}_1) = \lambda^m (\lambda - 1).$$

In this case, the block hybrid method derived from m equally spaced intra-step points is zero-stable and consistent for any selection of intra-step points.

3.3. Absolute Stability

Definition 2. A region is said to have absolute stability or be A-stable if it contains the entire left half-plane.

A region of absolute stability for the method can be defined as

$$\Re(y) = \{y \in \mathbb{C} : \mathcal{H}(y) < 1\}. \tag{41}$$

Applying the test equation [28]

$$u''' = \hat{\lambda}^3 u, \quad \lambda < 0,$$

to the new method gives

$$u_{r+p_i} = \mathcal{H}(y)u_r, \quad y = \hat{\lambda}^3 h^3, \tag{42}$$

where the $\mathcal{H}(y)$ is given by

$$\mathcal{H}(y) = \frac{\tilde{A}_0 + y p_0 + y^2 p_0^2 + y^3 \beta_0}{\tilde{A}_1 - y^3 \alpha_1}. \tag{43}$$

The stability function $\Re(y)$ can be determined by finding the dominant eigenvalues of the matrix $\mathcal{H}(y)$. The region of absolute stability for this method is shown in Figure 2. The stability function $\Re(y)$ is to be bounded if $\Re(y) \leq 1$. The bounded region is illustrated in Figure 3. The BHM is absolute-stable as the stability region contains the entire left half complex plane for $m = 2, 3, 4, 5$.

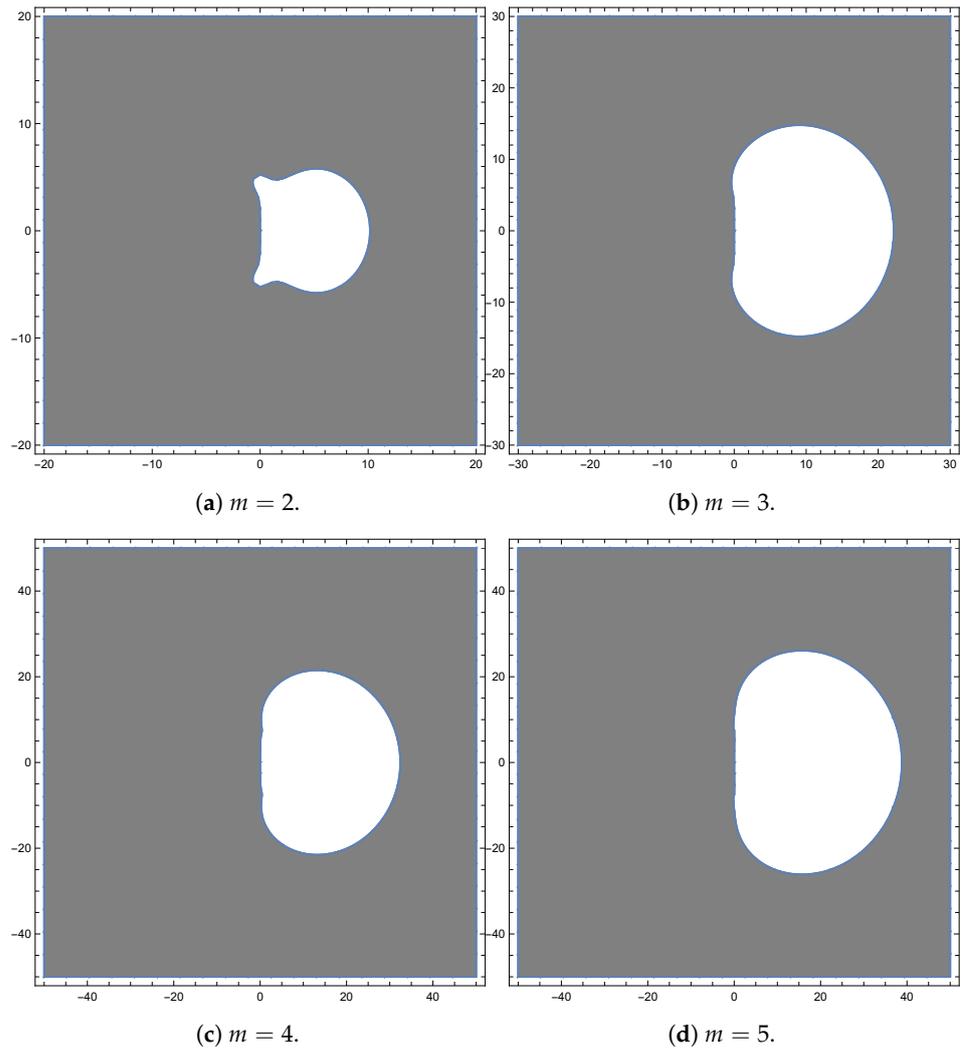


Figure 2. Region of absolute stability for $m = 2, 3, 4, 5.$

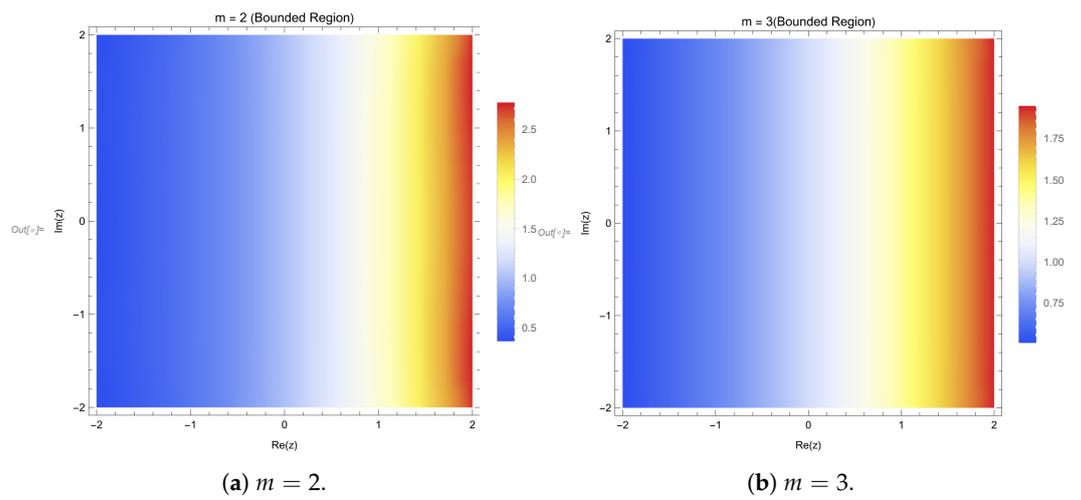


Figure 3. Cont.

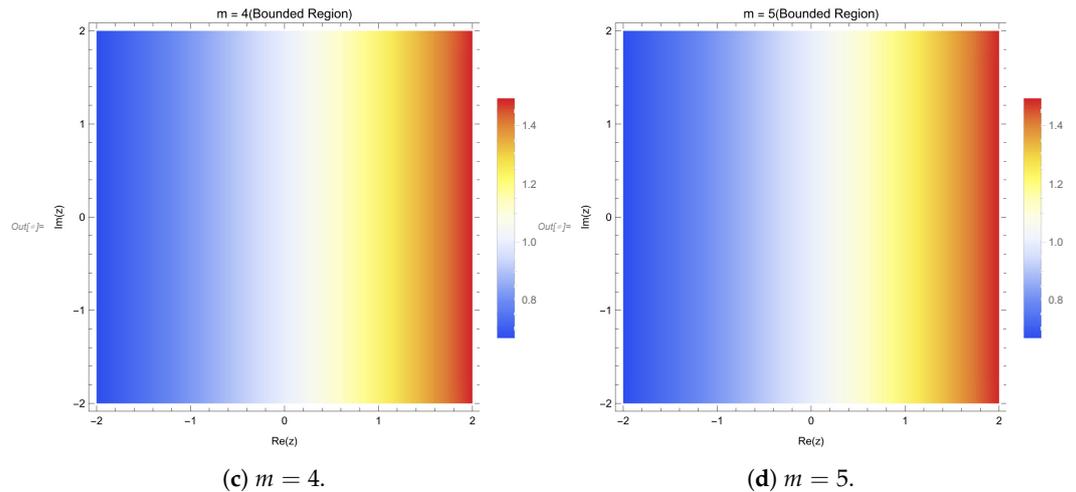


Figure 3. Region of absolute stability for $m = 2, 3, 4, 5.$

4. Implementation and Computational Procedure of the Proposed SIM-BHM

In this section, we present solutions to third-order IVPs (1)–(2) using the novel SIM-BHM iterative method. In order to evaluate the convergence of the method, we have evaluated the absolute error (AE) and absolute error estimate (AEE) between two consecutive iterations. Suppose $U(z), \delta(z)$ and $\gamma(z)$ are the approximate solutions. The absolute error is defined as

$$AE_u = |u_{r+p_i} - U_{r+p_i}|, \quad AE_{u'} = |u'_{r+p_i} - \delta_{r+p_i}|, \quad AE_{u''} = |u''_{r+p_i} - \gamma_{r+p_i}|. \quad (44)$$

We compute the absolute error estimate between two consecutive iterations using the following terms

$$AEE_u = |U_{r+p_i}^{s+1} - U_{r+p_i}^s|, \quad AEE_\delta = |\delta_{r+p_i}^{s+1} - \delta_{r+p_i}^s|, \quad AEE_\gamma = |\gamma_{r+p_i}^{s+1} - \gamma_{r+p_i}^s|. \quad (45)$$

where s represents the solution at the corresponding iteration. We enforce the following conditional stopping procedure in the SIM-BHM iterative method:

- If $AEE < Tol$, the current iteration solutions are deemed satisfactory, and the SIM-BHM proceeds to the next block.
- If $AEE > Tol$, the iteration count is incremented, and the SIM-BHM continues within the same block.

Here, Tol represents the user-defined tolerance. Within each block, the method iterates until the error AEE falls below Tol . Once the AEE converges to within an acceptable criteria, the procedure advances to the next block or concludes the computation. By applying this conditional structure, we ensure that the accuracy of our solutions is systematically improved within each block, thereby enhancing control and precision in our numerical computations.

Algorithm

To illustrate how to implement this SIM-BHM, we demonstrate an algorithm with the steps provided below:

1. Define function
 - Input: Initial value problem function $f(z, u, u', u'')$, interval $[a, b]$, number of blocks N , tolerance Tol , number of intra-step points m .
 - Output: Approximate solution for $u(z)$.
2. Linearization scheme
 - Linearize f by using Equation (8).

3. Discretization
 - Divide interval $[a, b]$ into N blocks and determine the step size h .
4. Collocation points
 - Generate a set of $m + 1$ fractions in Equation (10).
 - Calculate collocation points within each block using Equation (11).
5. Approximate solutions
 - Approximate the solutions within each block using a polynomial of degree $m + 3$ in Equations (12)–(14), which satisfies Equations (1) and (2).
6. Collocation equations
 - Set up collocation equations and the initial conditions within each block based on Equations (15)–(18).
 - Solve the system obtained from collocation Equations (15)–(18) to obtain coefficients $a_{r,k}$.
7. Initialize iteration
 - Initialize iteration $s = 1$.
8. Solve equations
 - Solve the system of equations obtained from collocation Equations (20)–(22) to obtain approximation solutions.
9. Iterate
 - Repeat Steps 7–10 until convergence or maximum iterations reached:
 - (a) Use Equation (28) to compute approximation solutions $u_{r+p_j}^{s+1}$, $\delta_{r+p_j}^{s+1}$, and $\gamma_{r+p_j}^{s+1}$ at collocation points.
 - (b) Compute absolute error estimate (AEE) between consecutive iterations using Equation (45).
 - (c) If $AEE_u, AEE_\delta, \text{ and } AEE_\gamma \leq Tol$, accept solutions and move to next block (Step 10).
 - (d) If $AEE_u, AEE_\delta, \text{ and } AEE_\gamma > Tol$, reject solutions and increase iteration s by 1.
10. Output
 - Output the approximate solutions u_{r+p_j} , δ_{r+p_j} , and γ_{r+p_j} at the block r .

5. Numerical Experimentation

In the next section, we test the method by implementing the SIM-BHM with some specific third-order IVPs from the literature.

Example 1. Consider the linear third-order IVP [13,17–19]

$$\begin{aligned} u''' - 3 \sin z &= 0, \quad z \in [0, b], \\ u''(0) &= -2, \quad u'(0) = 0, \quad u(0) = 1, \end{aligned}$$

This IVP has the exact solution: $u(z) = 3 \cos z + \frac{1}{2}z^2 - 2$.

Here, we selected $m = 6$, and to implement the SIM-BHM6 scheme, we defined

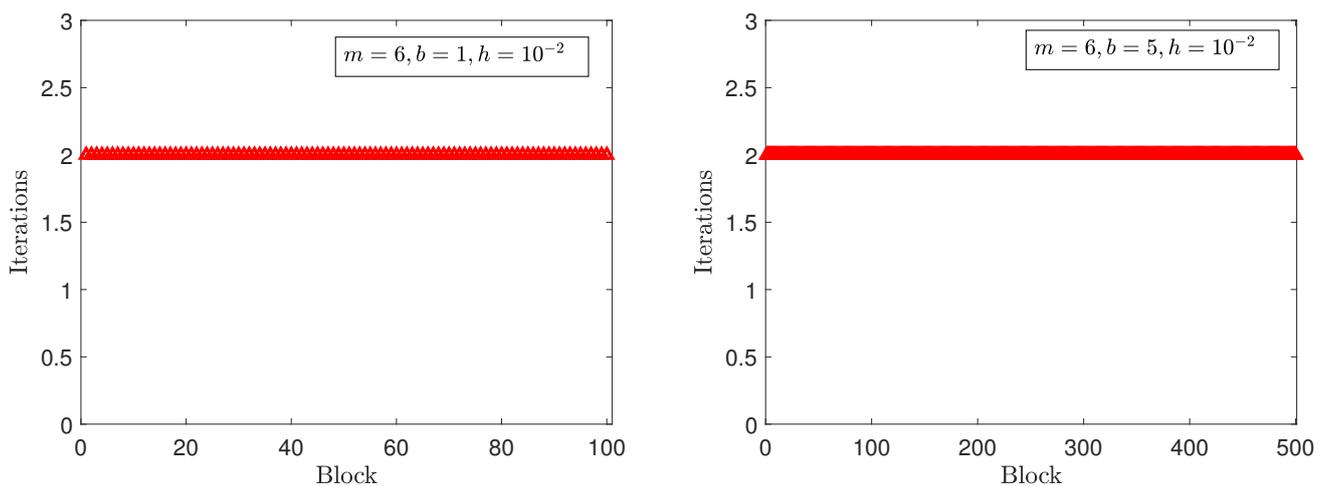
$$f = 3 \sin z, \quad \mathcal{L}_2 = 0, \quad \mathcal{L}_1 = 0, \quad \mathcal{L}_0 = 0, \quad \mathcal{N} = 3 \sin z.$$

In Table 2, we compare the maximum absolute errors of the SIM-BHM6 method with different variants of the hybrid block method [13,17–19]. Adesanya et al. [17] used the block method with six collocation points, Aro et al. [18] used the one-twelfth multi-step block method, Skwame et al. [19] used the equally spaced block method with five collocation

points, and Osa et al. [13] used the multi-step block method with fifth–fourth collocation points. It is evident that the SIM-BHM6 consistently outperforms the existing block methods in terms of reducing maximum absolute errors. Figure 4 showcases the number of iterations required in each block. The result provides strong evidence of the impressive convergence properties of SIM-BHM6, achieving good accuracy in just two iterations when $Tol = 10^{-18}$ across all blocks, as shown in Figure 4a,b. It is particularly noteworthy that the method attains a high level of accuracy while maintaining computational efficiency.

Table 2. Maximum absolute error comparison using SIM-BHM6.

	b	h	Adesanya et al. [17]	Areo and Omojola [18]	Skwame et al. [19]	Osa and Olaoluwa [13]	SIM-BHM6
Max AE	1	10^{-1}	2.2736×10^{-10}	5.8703×10^{-15}	3.5652×10^{-14}	1.3678×10^{-13}	4.4409×10^{-16}
CPU			-	-	-	-	0.024886
Max AE	5	-	-	-	-	-	1.0658×10^{-14}
CPU			-	-	-	-	0.029308
Max AE	1	10^{-2}	-	-	-	-	7.7716×10^{-16}
CPU			-	-	-	-	0.035961
Max AE	5	10^{-2}	-	-	-	-	1.2434×10^{-14}
CPU			-	-	-	-	0.096864



(a) $b = 1$ and $N = 100$.

(b) $b = 5$ and $N = 500$.

Figure 4. Iterations per block using the SIM-BHM6.

Example 2. Consider the linear third-order IVP [15,29,30]

$$u''' - 2u'' - 3u' + 10u + 34e^{-2z} + 16e^{-2z} + 10z^2 - 6z - 34 = 0, \quad z \in [0, b],$$

$$u''(0) = 0, \quad u'(0) = 0, \quad u(0) = 0,$$

The exact solution: $u(z) = z^2e^{-2z} - z^2 + 3$.

In this example, we used SIM-BHM6 with

$$f = 2u'' + 3u' - 10u - 34e^{-2z} - 16 \exp(-2z) - 10z^2 + 6z + 34,$$

$$\mathcal{L}_2 = 2, \quad \mathcal{L}_1 = 3, \quad \mathcal{L}_0 = 10, \quad \mathcal{N} = -34e^{-2z} - 16e^{-2z} - 10z^2 + 6z + 34.$$

Table 3 gives a comparison of the maximum absolute errors using the SIM-BHM6 method with different variants of the block method [15,29,30]. As shown in Table 3, Rufai and Ramos [15] used a variable step-size fourth-derivative block method, Awoyemi1

et al. [29] used a linear multi-step block method with five collocation points, and Allogmany and Ismail [30] used a fourth and fifth derivative block method with three implicit collocation points. We note that SIM-BHM6 outperformed the other block methods in terms of reducing maximum absolute errors. Figure 5 shows that the method achieves good accuracy in two iterations for $Tol = 10^{-10}$ across all blocks.

Table 3. Maximum absolute error comparison for SIM-BHM6 when $Tol = 10^{-10}$.

	b	h	Rufai and Ramos [15]	Awoyemi1 et al. [29]	Allogmany and Ismail [30]	SIM-BHM6
Max AE	1	10^{-1}	2.08069×10^{-11}	3.73600×10^{-5}	2.84944×10^{-9}	9.78773×10^{-13}
CPU			-	-	-	0.025860
Max AE	4	10^{-1}	4.50239×10^{-9}	-	-	3.5040×10^{-10}
CPU			-	-	-	0.031220
Max AE	4	10^{-2}	-	-	-	5.2776×10^{-12}
CPU			-	-	-	0.092856

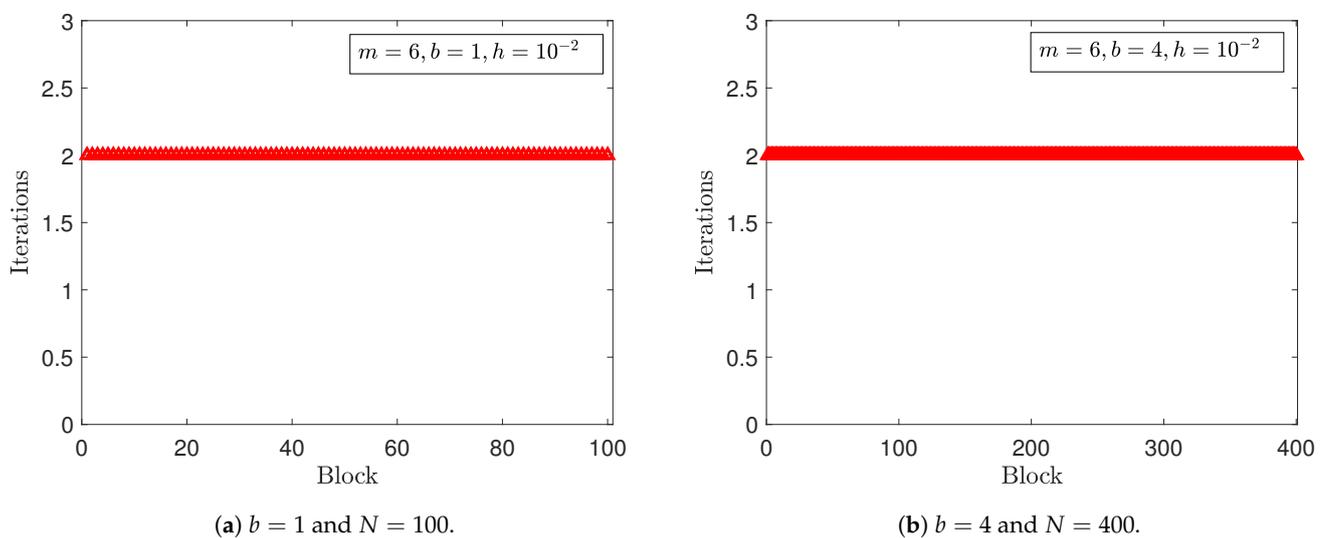


Figure 5. Iterations per block using SIM-BHM6.

Example 3. Consider the nonlinear third-order IVP [31]

$$u''' - zu'' + z^2u^2 - z \sin z + \cos z - z^2 \sin^2 z = 0, \quad z \in [0, b],$$

$$u''(0) = 0, \quad u'(0) = 1, \quad u(0) = 0,$$

The exact solution of Example 3 is $u(z) = \sin z$.

In this example, we selected $m = 4, m = 5,$ and $m = 6$. We define

$$f = zu'' - z^2u^2 + z \sin z - \cos z + z^2 \sin^2 z,$$

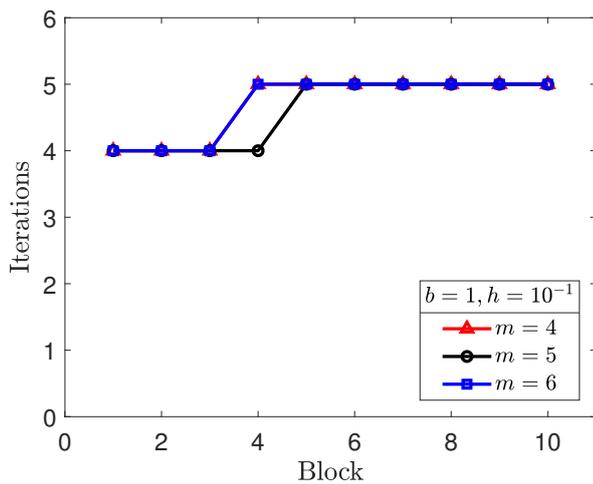
$$\mathcal{L}_2 = z, \quad \mathcal{L}_1 = 0, \quad \mathcal{L}_0 = -zu, \quad \mathcal{N} = z \sin z - \cos z + z^2 \sin^2 z.$$

In Table 4, we present a comparative analysis of the maximum absolute errors achieved using the SIM-BHM4, SIM-BHM5, and SIM-BHM6 methods with the work in [31]. Adeyeye and Omar [31] employed a block method with equally spaced collocation points. Notably, our findings show enhancement in accuracy when adopting SIM-BHM, particularly as we increase the number of intra-step points and reduce the step sizes. Figure 6a–c provide a visual representation of the number of iterations required across various blocks. In the case of shorter intervals, as in Figure 6a with 10 blocks, we observe that the maximum number of iterations required is five. As shown in Figure 6b, when 100 blocks are utilized,

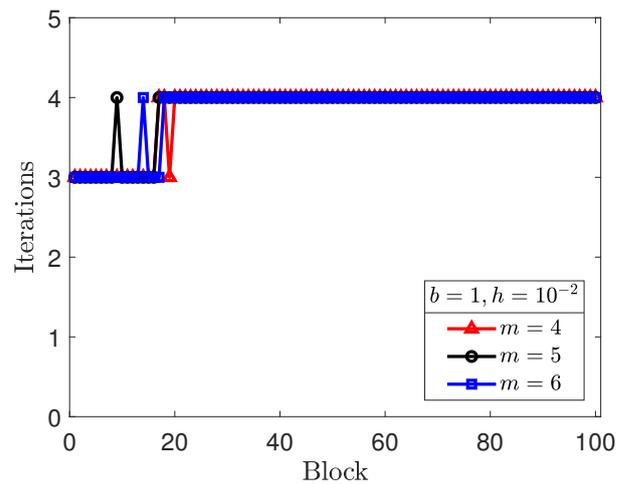
the maximum number of iterations reduces to four. These observations underscore the role of smaller step sizes in reducing the maximum iteration count within the blocks. For larger intervals, as illustrated in Figure 6a with 300 blocks, the maximum number of iterations remains at five. This suggests that, even in scenarios with a larger solution domain, the SIM-BHM maintains its efficient convergence characteristics. Figure 6d provides further evidence of SIM-BHM’s effectiveness, indicating that the method returns a maximum absolute error less than $O(10^{-13})$. This level of precision underscores the robustness and accuracy of the SIM-BHM approach. SIM-BHM6 generally outperforms both SIM-BHM4 and SIM-BHM5.

Table 4. Maximum absolute error comparison for SIM-BHM4, SIM-BHM5, and SIM-BHM6.

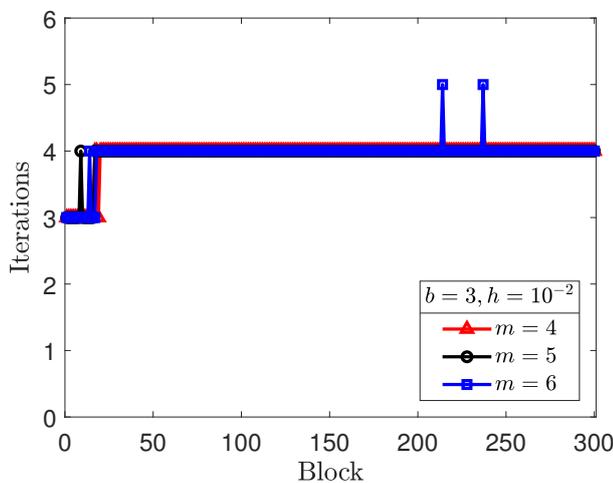
	b	h	Tol	Adeyeye and Omar [31]	SIM-BHM4	SIM-BHM5	SIM-BHM6
Max AE	1	10^{-1}	10^{-18}	4.644063×10^{-13}	4.640732×10^{-13}	5.173639×10^{-14}	5.551115×10^{-16}
CPU				-	0.013125	0.012183	0.013210
Max AE	1	10^{-2}	10^{-18}	-	1.4433×10^{-15}	1.7764×10^{-15}	9.9920×10^{-16}
CPU				-	0.028999	0.031710	0.031207
Max AE	3	10^{-2}	10^{-18}	-	4.1911×10^{-14}	2.1955×10^{-14}	1.8152×10^{-14}
CPU				-	0.083401	0.091006	0.094574



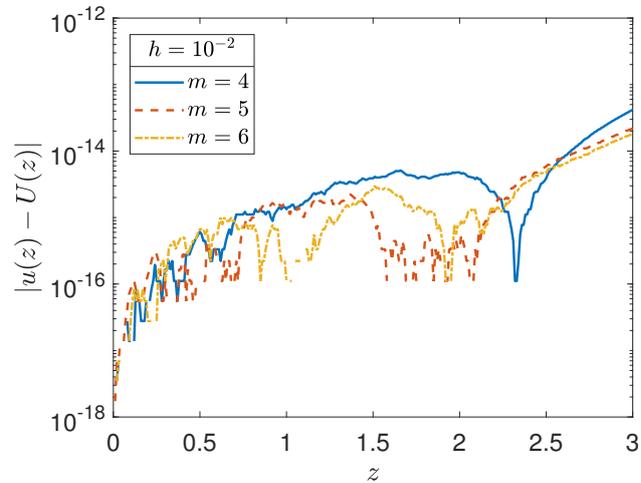
(a) $b = 1$, and $N = 10$.



(b) $b = 1$ and $N = 100$.



(c) $b = 3$ and $N = 300$.



(d) Absolute error for $b = 3$.

Figure 6. Iterations per block and absolute error for SIM-BHM when $m = 4$, $m = 5$, and $m = 6$.

Example 4. Consider a nonlinear system of third-order IVPs [15]

$$\begin{aligned}
 u''' &= \frac{1}{2}wv'e^{4z}, & u''(0) &= 1, & u'(0) &= -1, & u(0) &= 1, & z \in [0, b], \\
 v''' &= \frac{8}{3}uw'e^{2z}, & v''(0) &= 1, & v'(0) &= -2, & v(0) &= 4, \\
 w''' &= 27vu', & w''(0) &= 1, & w'(0) &= -3, & w(0) &= 9,
 \end{aligned}$$

with the exact solutions given as $u(z) = \exp(-z)$, $v(z) = \exp(-2z)$, and $w(z) = \exp(-3z)$. In the case of a system of equations, the SIM-BHM6 parameters are given by

$$\begin{aligned}
 f_1 &= \frac{1}{2}wv'e^{4z}, & \mathcal{L}_{21} &= 0, & \mathcal{L}_{11} &= 0, & \mathcal{L}_{01} &= 0, & \mathcal{N}_1 &= \frac{1}{2}wv'e^{4z}, \\
 f_2 &= \frac{8}{3}uw'e^{2z}, & \mathcal{L}_{22} &= 0, & \mathcal{L}_{12} &= 0, & \mathcal{L}_{02} &= 0, & \mathcal{N}_2 &= \frac{8}{3}uw'e^{2z}, \\
 f_3 &= 27wv', & \mathcal{L}_{23} &= 0, & \mathcal{L}_{13} &= 0, & \mathcal{L}_{03} &= 0, & \mathcal{N}_3 &= 27wv'.
 \end{aligned}$$

Table 5 shows a comparison of the maximum absolute errors achieved using SIM-BHM6 with the work of Rufai and Ramos [15]. Table 5 illustrates the superior accuracy and computational efficiency of SIM-BHM6 compared to the methodology proposed in [15]. SIM-BHM6 consistently yields more precise approximate solutions across a range of scenarios while maintaining efficient computational performance. Figure 7 provides a representation of the maximum iterations within the computational blocks. Notably, the reduction in user-defined tolerance led to an increase in the number of iterations between the blocks. We observed a maximum of six iterations. Figure 8 gives a comparison of the exact vs. numerical of the displacements in Figure 8a and phase portrait in Figure 8b.

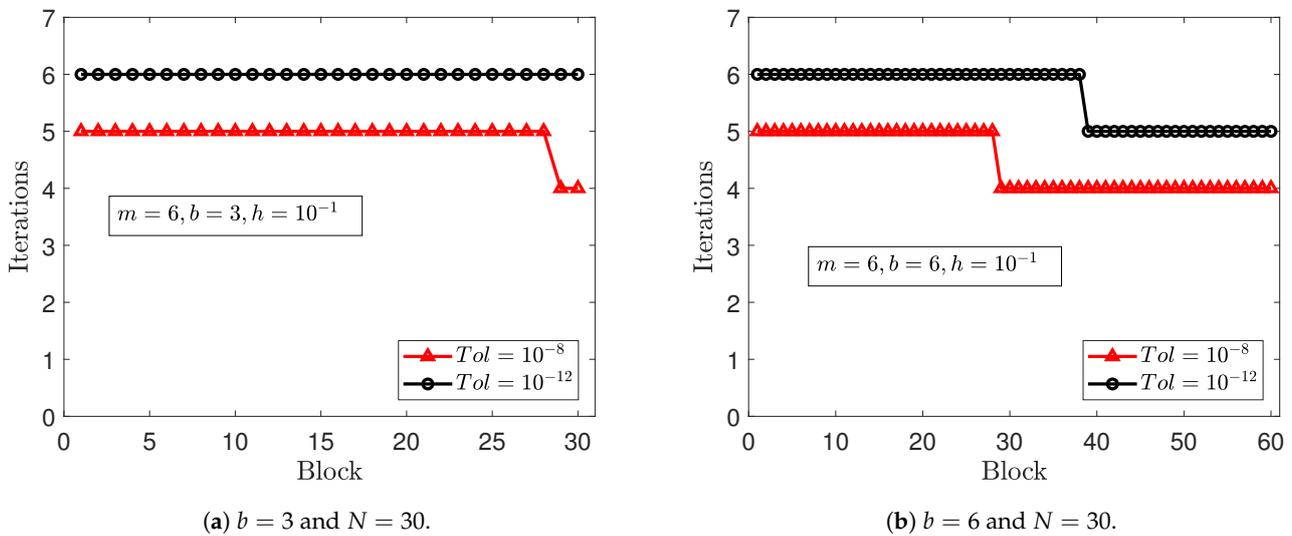


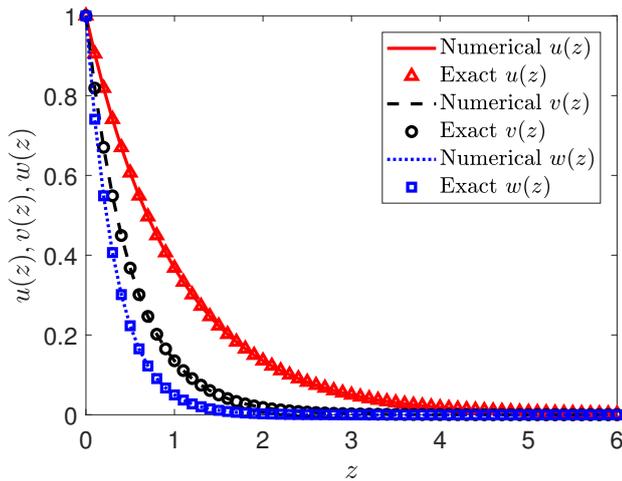
Figure 7. Iterations per block using SIM-BHM6 when $Tol = 10^{-8}$ and $Tol = 10^{-12}$.

Table 5. Maximum absolute error comparison using the SIM-BHM6.

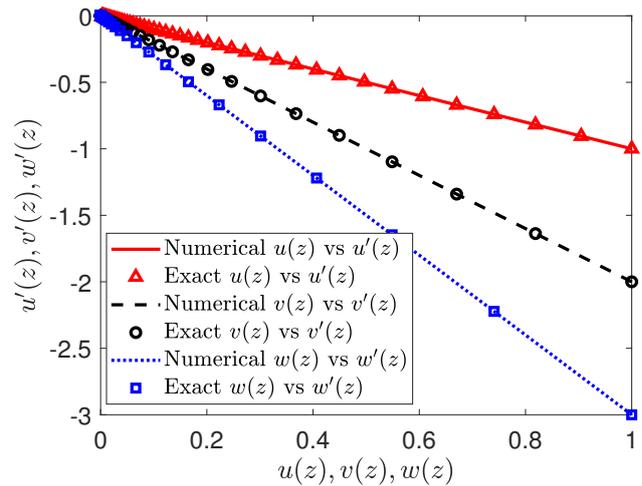
Method	b	h	Tol	Max AE_u	Max AE_v	Max AE_w	CPU
Rufai and Ramos [15]	3	10^{-1}	10^{-8}	4.36535×10^{-8}	3.08415×10^{-8}	2.43322×10^{-8}	0.172608
SIM-BHM6				4.4973×10^{-11}	4.6361×10^{-11}	1.6977×10^{-11}	0.094197
Rufai and Ramos [15]	6	10^{-1}	10^{-8}	8.81911×10^{-4}	2.95244×10^{-5}	2.71811×10^{-9}	0.172608
SIM-BHM6				2.5891×10^{-6}	1.1437×10^{-7}	7.1468×10^{-10}	0.122777

Table 5. Cont.

Method	b	h	Tol	Max AE_u	Max AE_v	Max AE_w	CPU
Rufai and Ramos [15] SIM-BHM6	3	10^{-1}	10^{-12}	2.38150×10^{-11} 2.1786×10^{-11}	1.50130×10^{-11} 3.0987×10^{-11}	2.66831×10^{-12} 1.2338×10^{-11}	0.395336 0.090950
Rufai and Ramos [15] SIM-BHM6	6	10^{-1}	10^{-12}	4.53908×10^{-7} 1.8995×10^{-6}	2.87265×10^{-8} 8.5346×10^{-8}	4.84118×10^{-10} 5.7199×10^{-10}	0.56434 0.126770



(a) Displacement profile.



(b) Phase portrait.

Figure 8. Numerical solution vs. exact solution using SIM-BHM6 when $Tol = 10^{-12}$, $h = 10^{-1}$, and $b = 6$.

Example 5. Consider a nonlinear IVP [13] given as

$$u''' - (2zu'' + u')u' = 0, \quad z \in [0, b],$$

$$u''(0) = 1, \quad u'(0) = \frac{1}{2}, \quad u(0) = 0,$$

with the exact solution:

$$u(z) = \frac{1}{2} + \log\left(\frac{2+z}{2-z}\right).$$

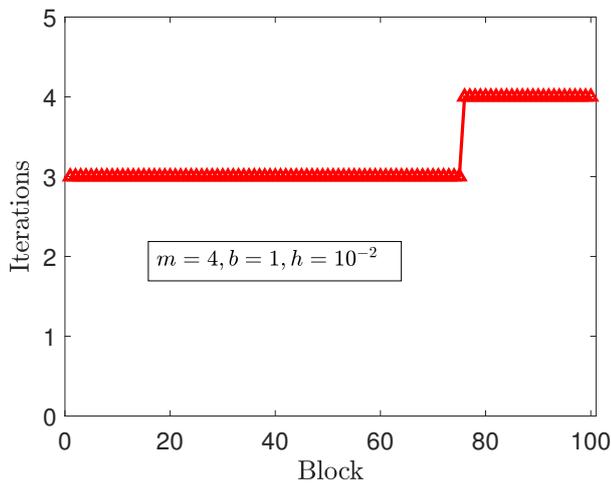
In this example, we use SIM-BHM4 with

$$f = (2zu'' + u')u', \quad \mathcal{L}_2 = 2zu', \quad \mathcal{L}_1 = u', \quad \mathcal{L}_0 = 0, \quad \mathcal{N}(z) = 0.$$

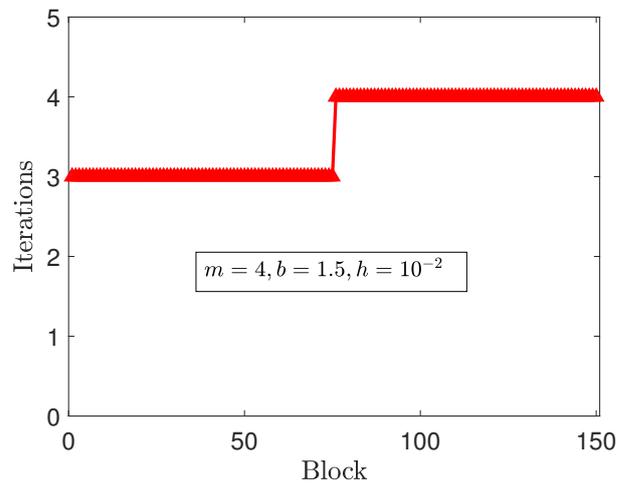
Table 6 provides a comprehensive comparison between the maximum absolute errors achieved using the SIM-BHM4 method and the results obtained by Osa and Olaoluwa [13]. Figure 9 illustrates the number of iterations within the computational blocks for different values of b for a tolerance $Tol = 10^{-10}$. The maximum number of iterations required was $s = 4$ for short and large intervals. The corresponding absolute error profiles for $u(z)$, $u'(z)$ and $u''(z)$ are plotted in Figure 10a. Figure 10b shows the exact vs. numerical solution for $u(z)$, $u'(z)$, and $u''(z)$.

Table 6. Maximum absolute errors comparison for SIM-BHM4 when $Tol = 10^{-10}$.

b	h	Osa and Olaoluwa [13] Max $ u(z) - U(z) $	SIM-BHM4 Max $ u(z) - U(z) $	Max $ u'(z) - \delta(z) $	Max $ u''(z) - \gamma(z) $	CPU
1	10^{-2}	2.6241×10^{-12}	1.7764×10^{-15}	3.7748×10^{-15}	8.5487×10^{-15}	0.042517
1.5	10^{-2}	-	1.5765×10^{-14}	1.7342×10^{-13}	3.3595×10^{-13}	0.051358

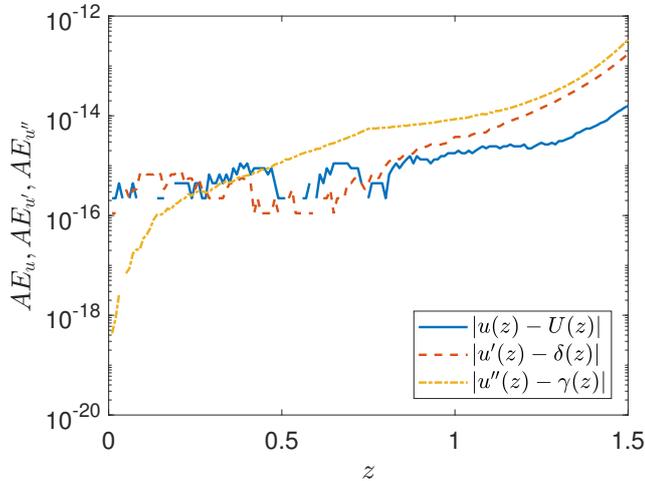


(a) $b = 1$ and $N = 100$.

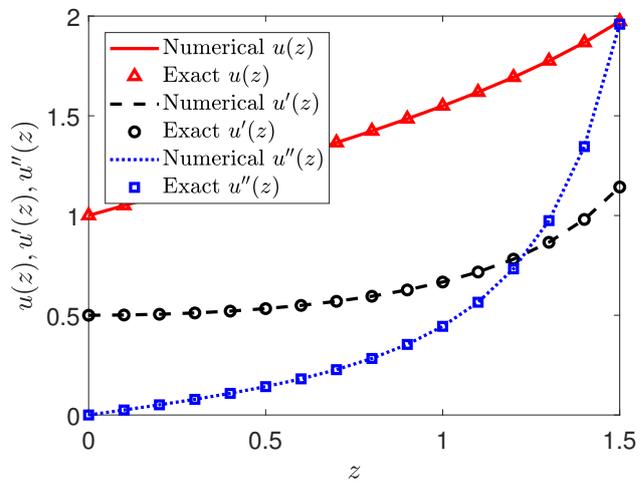


(b) $b = 1.5$ and $N = 150$.

Figure 9. Iterations per block using the SIM-BHM4 when $Tol = 10^{-10}$.



(a) Absolute error.



(b) Solution profile.

Figure 10. Absolute error and numerical solution vs. exact solution using the SIM-BHM4 when $Tol = 10^{-10}$, $h = 10^{-1}$, and $b = 1.5$.

6. On the Application of the SIM-BHM to Solve Nonlinear Jerk Equations

The concept of jerk is particularly relevant in the study of motion and control systems, especially in engineering and physics [32,33]. In physics, the jerk equation models particle motion under varying forces, providing valuable insights into dynamic systems [34]. The equation is used to model systems where the rate of change in acceleration is important, such as in vibration analysis, control systems, and motion planning. Additionally, jerk analysis finds application in signal processing and control theory, facilitating the filtering of noisy signals and the detection of anomalies, contributing to technological advancements.

Higher derivatives of motion, such as snap, are also important in motion control and can be experienced in everyday life, such as on trampolines and roller coasters [35].

We consider a class of nonlinear jerk equation IVP containing a third derivative of position with respect to time that describes the rate of change in acceleration, and it is given by

$$u''' + \beta_1 u' + \beta_2 (u')^3 + \beta_3 u^2 u' + \beta_4 u u' u'' + \beta_5 u' (u'')^2 = 0, \quad z \in [0, b], \quad (46)$$

$$u''(0) = 0, \quad u'(0) = \beta_0, \quad u(0) = 0, \quad (47)$$

where the parameters $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4,$ and β_5 are constant. Since there is no analytic solution to Equation (46), we compared the results of the solution profiles to approximated solutions that have been previously reported in the literature [36–39].

- **Case 1:** $\beta_1 = \beta_4 = \beta_5 = 0$ and $\beta_2 = \beta_3 = 1$. Then, the nonlinear jerk Equation (46) takes the form

$$u''' + u^2 u' + (u')^3 = 0, \quad z \in [0, b], \quad (48)$$

$$u''(0) = 0, \quad u'(0) = \beta_0, \quad u(0) = 0, \quad (49)$$

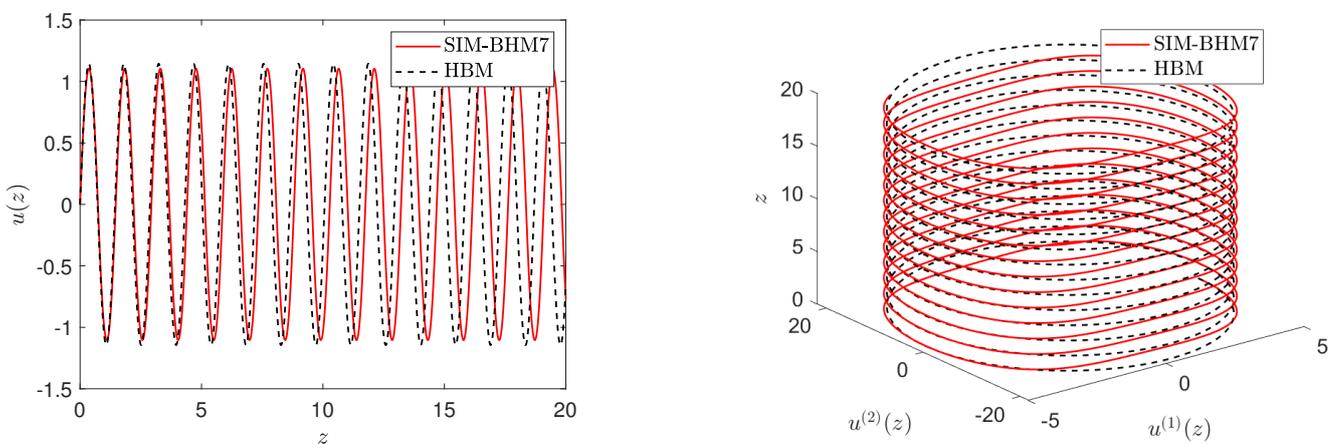
Gottlieb [38] employed the harmonic balance method (HBM) approach approximation solution to solve the nonlinear Equation (48). He found that HBM yields a good approximated solution of the period and displacement amplitude of oscillations for a range of values of initial velocity, given as

$$u(z) = \frac{\beta_0}{\Omega} \sin \Omega z, \quad \Omega = \frac{1}{2\sqrt{2}} \sqrt{3\beta_0^2 + \sqrt{9\beta_0^4 + 16\beta_0^2}}$$

where Ω is the angular frequency. To solve Equation (48) numerically, we use SIM-BHM7 with

$$f = -u^2 u' - (u')^3, \quad \mathcal{L}_2 = 0, \quad \mathcal{L}_1 = -u^2 - (u')^2, \quad \mathcal{L}_0 = 0, \quad \mathcal{N}(z) = 0.$$

Case 1 is solved using SIM-BHM7 with $h = 0.01$ and $Tol = 10^{-14}$ in the domain of integration $[0, 20]$. The displacement and phase trajectory are given in Figure 11a and 11b, respectively. A comparison of the approximated analytical vs. numerical solution is depicted by plotting the two results on the same graph. The corresponding iterations per block and the absolute error profiles for the displacement are plotted in Figure 12a and 12b, respectively, for different values of the initial velocity β_0 .



(a) Displacement $u(z)$.

(b) Phase trajectory.

Figure 11. Displacement and phase trajectory for using SIM-BHM7 when $Tol = 10^{-14}$, $b = 20$, $h = 0.01$, and $N = 2000$.

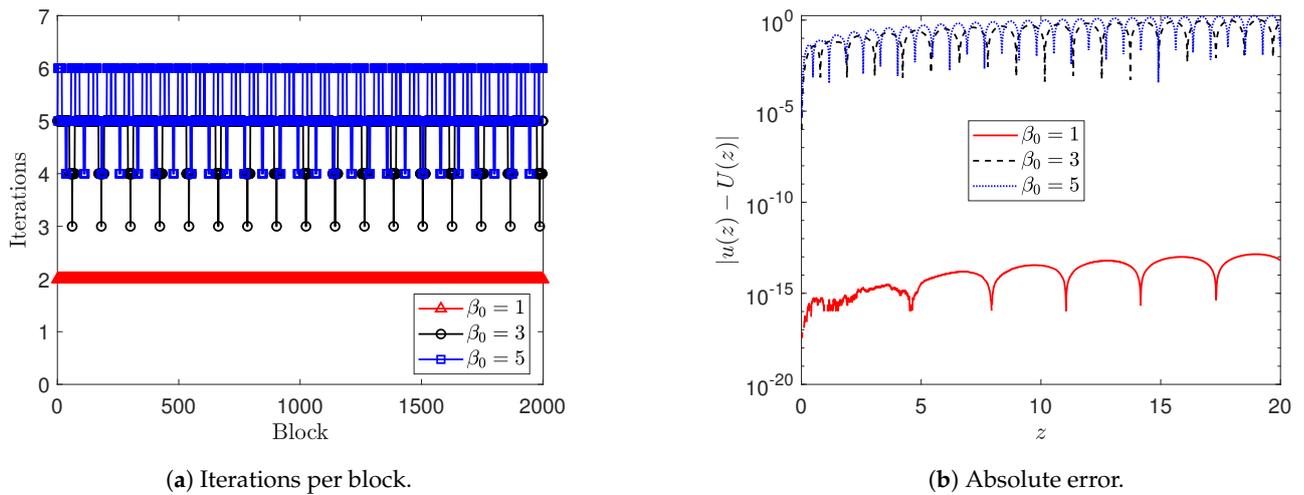


Figure 12. Iterations per block and absolute error for using the SIM-BHM7 when $\beta_0 = 5$, $Tol = 10^{-14}$, $b = 20$, $h = 0.01$, and $N = 2000$.

- **Case 2:** $\beta_2 = \beta_3 = \beta_5 = 0$, $\beta_1 = 1$ and $\beta_4 = -1$. For the **Case 2** values of the parameters, the nonlinear jerk Equation (46) is given as

$$u''' + u' - uu'u'' = 0, \quad z \in [0, b], \tag{50}$$

$$u''(0) = 0, \quad u'(0) = \beta_0, \quad u(0) = 0. \tag{51}$$

Mirzabeigy and Yildirim [39] employed the modified differential transform method (MDTM) to obtain approximate periodic solutions to Equation (50), given as

$$u(z) = \frac{\beta_0}{\Omega} \sin \Omega z + \frac{\beta_0}{96\Omega^3} [(48\Omega - 9\beta_0^2 - 48) \sin \Omega z - \beta_0^2 \sin 3\Omega z],$$

$$\Omega = \frac{1}{2} \sqrt{\beta_0^2 + 4}.$$

In **Case 2**, we employ SIM-BHM3 with

$$f = -u' + uu'u'', \quad \mathcal{L}_2 = uu', \quad \mathcal{L}_1 = -1, \quad \mathcal{L}_0 = 0, \quad \mathcal{N}(z) = 0.$$

To determine the accuracy of SIM-BHM3, the numerical results are compared with Kashkari and Alqarni [40]. In their work, they implemented a two-step hybrid block method (TSHBM) with a polynomial of degree 6. Table 7 gives a comparison of the SIM-BHM for $\beta_0 = 0.2$ and $\beta_0 = 0.3$. It can be observed that SIM-BHM3 gives similar results to MDTM and TSHBM within a fast CPU time.

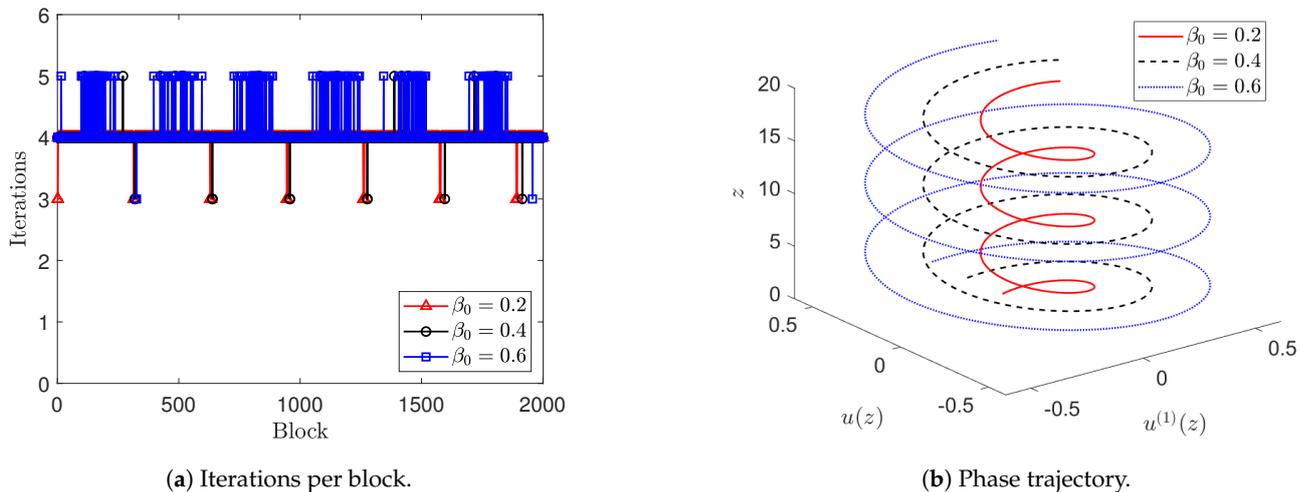
Table 7. Comparison of implementation the SIM-BHM3 when $h = 0.125$, $b = 2$, and $Tol = 10^{-18}$.

z	$\beta_0 = 0.2$			$\beta_0 = 0.3$		
	MDTM [39]	TSHBM [40]	SIM-BHM3	MDTM [39]	TSHBM [40]	SIM-BHM3
0.000	0	0	0	0	0	0
0.125	0.024934	0.024934	0.024934950723562	0.037402	0.037402	0.037402433678462
0.250	0.049480	0.049480	0.049480895840067	0.074220	0.074220	0.074221538790320
0.375	0.073253	0.073253	0.073255260278968	0.109878	0.109879	0.109884305379430
0.500	0.095881	0.095881	0.095888161918553	0.143814	0.143814	0.143837971511998
0.625	0.117008	0.117008	0.117028372165012	0.17549	0.17549	0.175559287806927
0.750	0.1363	0.136300	0.136348852382876	0.204397	0.204399	0.204562874433873
0.875	0.153452	0.153453	0.153551767255572	0.23007	0.230073	0.230408498532534
1.000	0.168191	0.168191	0.168372904835116	0.252088	0.252093	0.252707185491022
1.150	0.18028	0.180281	0.180585461680617	0.270088	0.270095	0.271126160102852

Table 7. Cont.

z	$\beta_0 = 0.2$			$\beta_0 = 0.3$		
	MDTM [39]	TSHBM [40]	SIM-BHM3	MDTM [39]	TSHBM [40]	SIM-BHM3
1.250	0.189525	0.189526	0.190003176962089	0.283769	0.283778	0.285392683210093
1.375	0.195777	0.195778	0.196482819208038	0.292902	0.292913	0.295296898220800
1.500	0.198935	0.198937	0.199926041919773	0.297332	0.297344	0.300693825250087
1.625	0.198948	0.198949	0.200280629084105	0.296984	0.296996	0.301504637583959
1.750	0.195815	0.195816	0.197541149338335	0.291864	0.291876	0.297717328253788
1.875	0.189587	0.189589	0.191749029829054	0.282059	0.282007	0.289386829338821
2.000	0.180365	0.180365	0.182992050066185	0.267734	0.267745	0.276634591040122
CPU (s)	-	-	0.020688	-	-	0.019756

Figure 13a depicts the number of iterations required per block for different values of the initial velocity $\beta_0 = 0.2, 0.4, 0.6$. It can be observed that SIM-BHM3 converges within five iterations. Figure 13b illustrates the phase portrait for different values of the initial velocity.



(a) Iterations per block.

(b) Phase trajectory.

Figure 13. Displacement and phase trajectory for using the SIM-BHM3 when $Tol = 10^{-18}$, $b = 20$, $h = 0.01$, and $N = 2000$.

- **Case 3:** $\beta_2 = \beta_3 = \beta_4 = 0$, $\beta_1 = 1$, and $\beta_5 = 1$.

$$u''' + u' + u'(u'')^2 = 0, \quad z \in [0, b], \tag{52}$$

$$u''(0) = 0, \quad u'(0) = \beta_0, \quad u(0) = 0. \tag{53}$$

The periodic solutions to Equation (52) are given as [39]

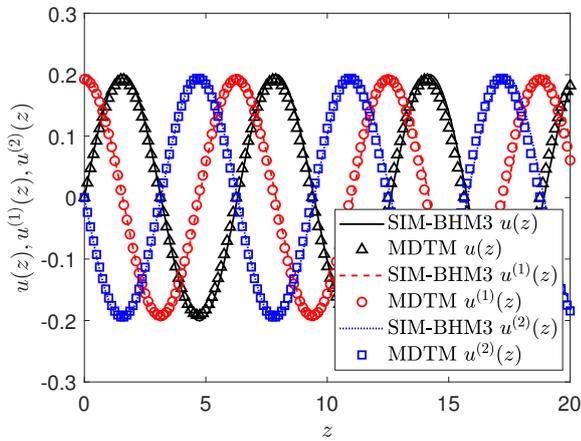
$$u(z) = \frac{\beta_0}{\Omega} \sin \Omega z + \frac{\beta_0}{96\Omega^3} \left[(48\Omega - 9\beta_0^2 - 48) \sin \Omega z - \beta_0^2 \sin 3\Omega z + (12\Omega^3 \beta_0^2 + 48\Omega - 48\Omega^3) z \cos \Omega z \right],$$

$$\Omega = 2\sqrt{\frac{1}{4 - \beta_0^2}}.$$

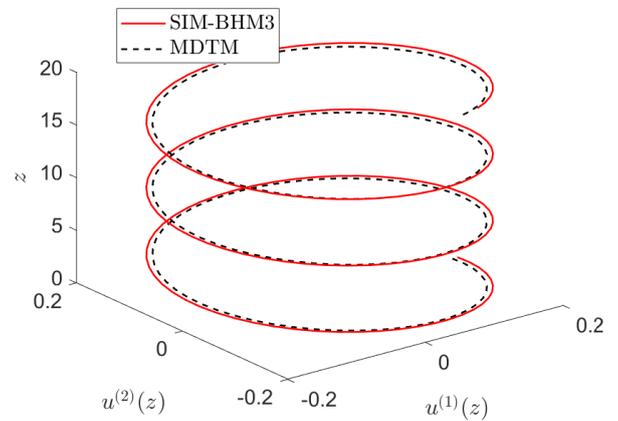
Here, we set

$$f = -u' - u'(u'')^2, \quad \mathcal{L}_2 = -u'u'', \quad \mathcal{L}_1 = -1, \quad \mathcal{L}_0 = 0, \quad \mathcal{N}(z) = 0.$$

The displacement, velocity, acceleration, and phase portrait trajectories are given in Figure 14. A comparison of the exact (MDTM) vs. numerical (SIM-BHM3) solutions is depicted by plotting the two results on the same graph. The corresponding phase portrait and iterations per block are plotted in Figure 15a and 15b, respectively, for different values of β_0 .

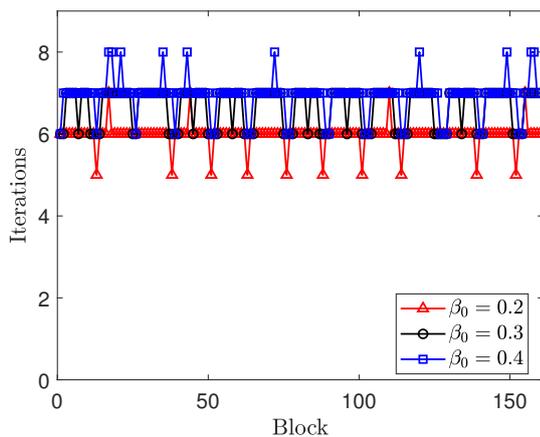


(a) Displacement $u(z)$, velocity $u^{(1)}(z)$, and acceleration $u^{(2)}(z)$.

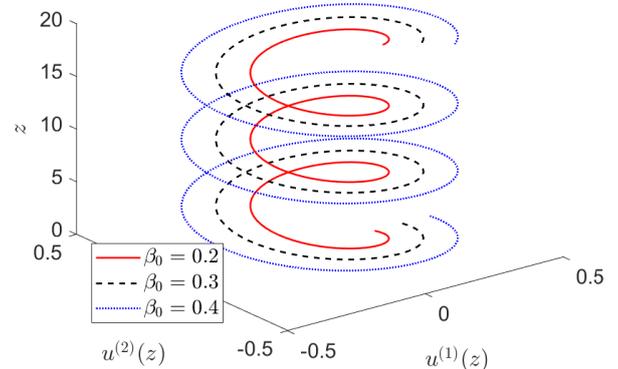


(b) Phase trajectory.

Figure 14. Displacement, velocity, acceleration, and phase portrait for using the SIM-BHM3 when $\beta_0 = 0.2, Tol = 10^{-18}, b = 20, h = 0.01,$ and $N = 200$.



(a) Iterations per block when $b = 2$ and $h = 0.125$.



(b) Phase portrait when $b = 20$ and $h = 0.01$.

Figure 15. Displacement and phase trajectory for using SIM-BHM3 when $Tol = 10^{-18}$.

- **Case 4:** $\beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 1$. For the **Case 4** values of the parameters, the nonlinear jerk Equation (46) is given as

$$u''' + u' + (u')^3 + u^2u' + uu'u'' + u'(u'')^2 = 0, \quad z \in [0, b], \tag{54}$$

$$u''(0) = 0, \quad u'(0) = \beta_0, \quad u(0) = 0. \tag{55}$$

With linearization, we obtain

$$f = -u' - (u')^3 - u^2u' - uu'u'' - u'(u'')^2, \quad \mathcal{L}_2 = -uu' - u'u'',$$

$$\mathcal{L}_1 = -1 - (u')^2 - u^2, \quad \mathcal{L}_0 = 0, \quad \mathcal{N}(z) = 0.$$

Equation (54) is solved using SIM-BHM7 with $h = 0.01$ in the domain of integration $[0, 40]$ for different values of the initial velocity β_0 . The displacement and iterations per block utilized to solve Equation (54) are plotted in Figure 16 for different values of the initial velocity.

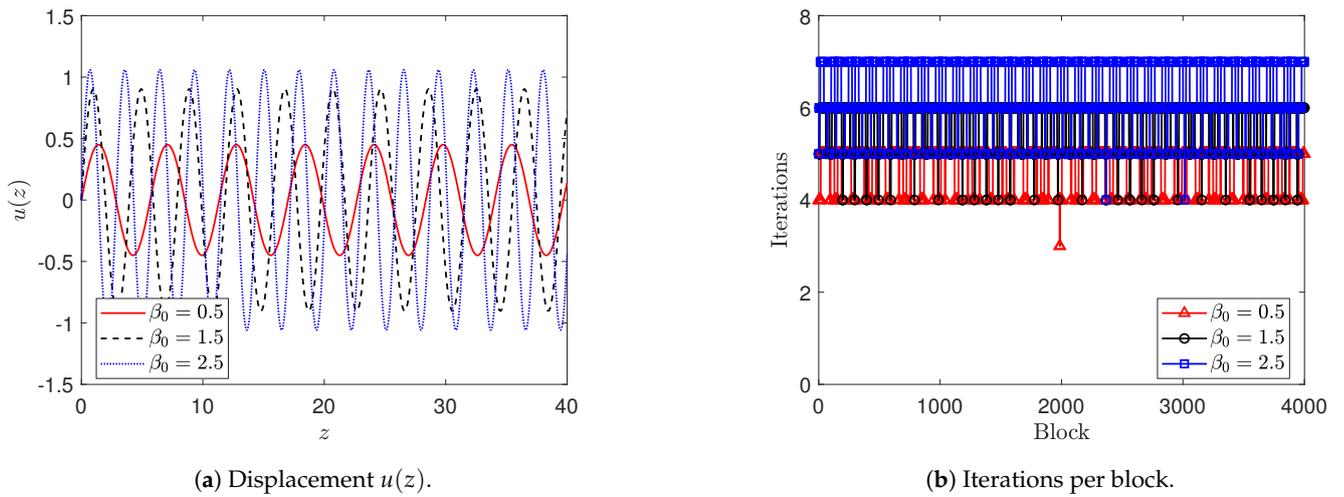


Figure 16. Displacement and iterations per block for using SIM-BHM7 when $Tol = 10^{-18}$, $b = 40$, and $N = 4000$.

Figure 17 depicts phase portraits for **Case 4** when varying β_0 .

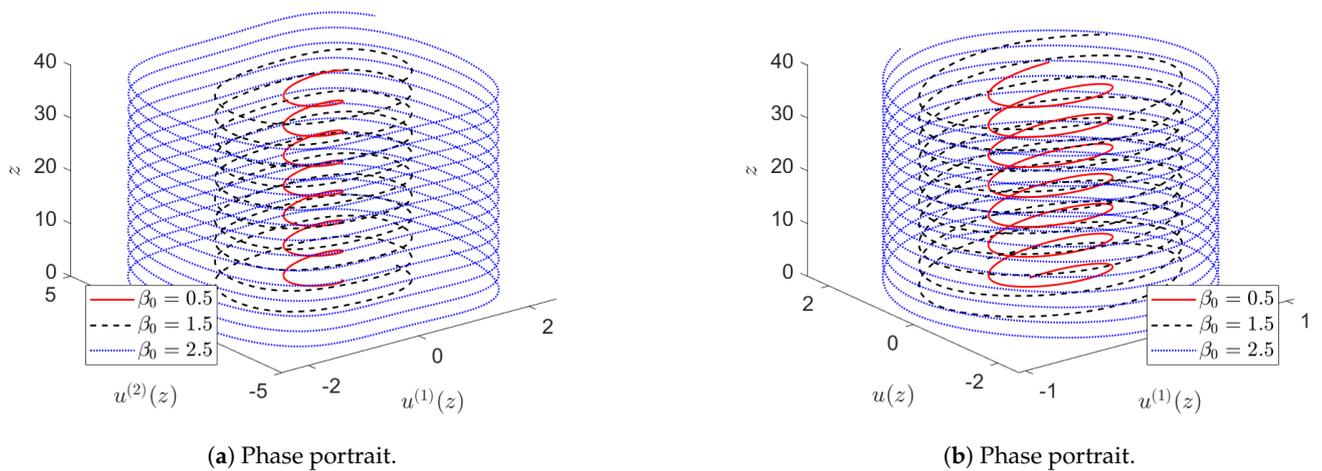


Figure 17. Phase trajectory for using the SIM-BHM7 when $Tol = 10^{-18}$, $b = 40$, and $N = 4000$.

- **Case 5:** $\beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 4$. For the **Case 5** values of the parameters, the nonlinear jerk Equation (46) is given as

$$u''' + 4[u' + (u')^3 + u^2u' + uu'u'' + u'(u'')^2] = 0, \quad z \in [0, b], \tag{56}$$

$$u''(0) = 0, \quad u'(0) = \beta_0, \quad u(0) = 0. \tag{57}$$

With linearization, we obtain

$$f = -4u' - 4(u')^3 - 4u^2u' - 4uu'u'' - 4u'(u'')^2, \quad \mathcal{L}_2 = -4uu' - 4u'u'',$$

$$\mathcal{L}_1 = -4 - 4(u')^2 - 4u^2, \quad \mathcal{L}_0 = 0, \quad \mathcal{N}(z) = 0.$$

Equation (56) is solved using SIM-BHM7 with $h = 0.01$ and $b = 40$ for different values of the initial velocity. Figure 18 illustrates the solution $u(z)$ and the iteration required. Figure 19 shows the phase portrait.

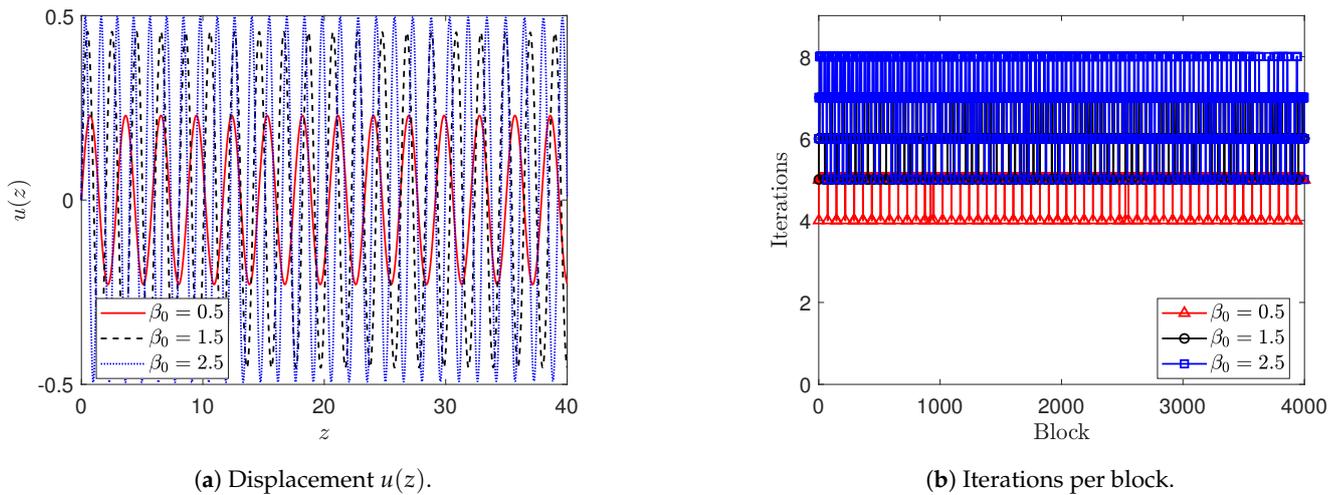


Figure 18. Displacement and iterations per block for using SIM-BHM7 when $Tol = 10^{-18}$, $b = 40$, and $N = 4000$.

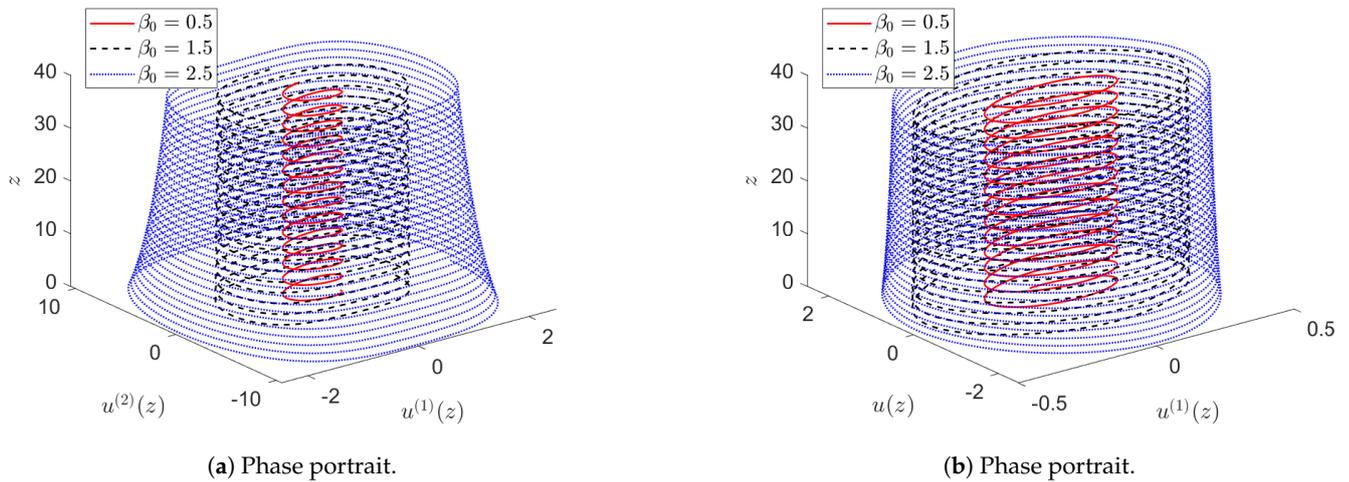


Figure 19. Phase portraits for using SIM-BHM7 when $Tol = 10^{-18}$, $b = 40$, and $N = 4000$.

We considered five cases of nonlinear jerk Equation (46) with different parameter combinations for Equations (48)–(56). This allowed us to test the SIM-BHM approach on various forms of the jerk equation. For **Cases 1–3**, known analytical solutions from previous works were available to validate our numerical results. The SIM-BHM-produced displacement profiles, velocities, accelerations, and phase portraits are in agreement with these established solutions. In **Cases 4 and 5**, there were no analytical solutions. The SIM-BHM generated physically realistic oscillatory behaviors as the initial velocity parameter β_0 was varied. This indicates that the method can reliably solve these types of jerk equations numerically. Across all test cases, the SIM-BHM converged rapidly, typically requiring only 2–8 iterations per block. Even for large solution domains and higher β_0 values producing larger oscillations, the method maintained its fast convergence. The displacement profiles, phase portraits, and plots of the iterations clearly illustrate the accuracy and consistency of the SIM-BHM approach. Comparisons with previous works validated the high precision of the numerical solutions obtained. In conclusion, the SIM-BHM proves to be an effective technique for solving an important class of nonlinear jerk equations. It generates solutions in close agreement with the known results. The method exhibits robust computational performance and accuracy across a wide range of problem scenarios.

The numerical experiments consistently showed the SIM-BHM achieving higher accuracy when a smaller step size h was used. For example, reducing h from 10^{-1} to 10^{-2} led to

errors 1–2 orders of magnitude smaller across test problems. This implies that the method exhibits the expected property of increased accuracy for smaller discretization. Larger solution domains were also handled effectively. **Cases 1–3** of the jerk equations covered intervals up to $b = 20$ with good precision. **Cases 4 and 5** were solved over the wider range of $b = 40$ while maintaining accuracy. The SIM-BHM modeled problems requiring integration over large physical spaces. Additional support comes from Example 5, where accuracy improved for the longer interval of $b = 1.5$ compared to $b = 1$. These examples provide strong evidence that SIM-BHM's accuracy increases predictably with smaller h and can be systematically extended to large b values through domain discretization.

7. Conclusions

We introduced a novel SIM-BHM iterative method for solving third-order IVPs with a fixed step size. The primary objective of the SIM-BHM is to obtain accurate solutions to the IVPs. To evaluate the convergence of the method, we evaluated the absolute error and the absolute error estimate between consecutive iterations. We applied the SIM-BHM to a variety of IVPs, both linear and nonlinear. We compared its performance with existing block methods. The results consistently demonstrated the superior performance of the SIM-BHM in terms of returning the lowest maximum absolute errors, highlighting its accuracy and efficiency. The SIM-BHM offers a versatile approach by employing a selective intra-step method and ensuring convergence. The method gives highly accurate solutions with good computational performance. The main findings with regard to this numerical method are as follows:

- Smaller intra-step sizes generally lead to smaller truncation errors.
- Increasing the number of collection points improves the accuracy of SIM-BHM.
- For large intervals, the SIM-BHM gave more accurate approximations than some existing BHMs, for example, those in [13,15,17–19,29–31].
- The SIM-BHM gives robust computational performance using only a few iterations.

The SIM-BHM method was applied to solve nonlinear jerk equations. Jerk equations are an important class of third-order problems with applications in fields like engineering and physics. Five cases of jerk equations with varying parameter combinations were tested. Comparisons to known approximate solutions from previous works showed excellent agreement for **Cases 1–3**. The method generated physically realistic oscillatory behaviors for **Cases 4 and 5**. Across all jerk equation problems, the SIM-BHM maintained its fast convergence, typically requiring only a few iterations per block even for large solution domains and higher oscillation amplitudes. The accuracy of the numerical solutions was validated through comparisons. This demonstrates the SIM-BHM provides an effective technique for reliably solving nonlinear jerk equations.

In future work, the following would allow for improving the performance of the SIM-BHM:

- Implementing adaptivity with variable step size would allow the method to better capture solutions with varying behaviors over the domain. This could enhance accuracy.
- Investigating the use of Legendre–Gauss–Lobatto and Gauss–Lobatto collocation points, which have been shown to improve accuracy.
- Applying optimally spaced intra-step points within each block, as studies have observed non-uniform point distributions lead to higher precision than uniform spacing [7].
- Applying the overlapping domain approach, where blocks intersect, as this has been demonstrated to reduce errors versus standard non-overlapping grids [4,41].
- Extending the SIM-BHM framework to other problem types, such as higher-order initial value problems, boundary value problems, singular differential equations, and fractional differential equations, to evaluate performance on broader classes of equations.

Author Contributions: Conceptualization, S.A.A.A.A.A.; Investigation, S.A.A.A.A.A. and U.O.R.; Writing—original draft, S.A.A.A.A.A.; Writing—review & editing, P.S., S.P.G., H.S.M. and O.A.I.N.; Supervision, P.S., S.P.G. and O.A.I.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the University of KwaZulu-Natal.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AE	Absolute error
AEE	Absolute error estimate
BHM	Block hybrid method
HBM	Harmonic balance method
MDTM	Modified differential transform method
ODE	Ordinary differential equation method
TSHBM	Two-step hybrid block method
Max AE	Largest absolute error
SIM	Simple iteration method

References

- Gragg, W.B.; Stetter, H.J. Generalized multistep predictor-corrector methods. *J. Assoc. Comput. Mach.* **1964**, *11*, 188–209. [[CrossRef](#)]
- Shampine, L.F.; Watts, H. Block implicit one-step methods. *Math. Comput.* **1969**, *23*, 731–740. [[CrossRef](#)]
- Gear, C.W. Hybrid methods for initial value problems in ordinary differential equations. *J. Soc. Ind. Appl. Math. Ser. B Numer. Anal.* **1965**, *2*, 69–86. [[CrossRef](#)]
- Motsa, S. Overlapping Grid-Based Optimized Single-Step Hybrid Block Method for Solving First-Order Initial Value Problems. *Algorithms* **2022**, *15*, 427. [[CrossRef](#)]
- Shateyi, S. On the Application of the Block Hybrid Methods to Solve Linear and Non-Linear First Order Differential Equations. *Axioms* **2023**, *12*, 189. [[CrossRef](#)]
- Burden, R.; Faires, J. *Numerical Analysis*, 9th ed.; Cengage Learning: Boston, MA, USA, 2011.
- Orakwelu, M.G. Generalised Implicit Block Hybrid Algorithms for Initial Value Problems. Ph.D. Thesis, University of KwaZulu-Natal, Pietermaritzburg, South Africa, 2019.
- El-Hawary, H.; Mahmoud, S. On some 4-point spline collocation methods for solving second-order initial value problems. *Appl. Numer. Math.* **2001**, *38*, 223–236. [[CrossRef](#)]
- Bellman, R.E.; Kalaba, R.E. *Quasilinearization and Nonlinear Boundary-Value Problems*; Elsevier: New York, NY, USA, 1965.
- Mandelzweig, V.; Tabakin, F. Quasilinearization approach to nonlinear problems in physics with application to nonlinear ODEs. *Comput. Phys. Commun.* **2001**, *141*, 268–281. [[CrossRef](#)]
- Kelley, C.T. *Iterative Methods for Linear and Nonlinear Equations*; SIAM: Philadelphia, PA, USA, 1995.
- Ramos, J.I. On the variational iteration method and other iterative techniques for nonlinear differential equations. *Appl. Math. Comput.* **2008**, *199*, 39–69. [[CrossRef](#)]
- Osa, A.L.; Olaoluwa, O.E. A fifth-fourth continuous block implicit hybrid method for the solution of third order initial value problems in ordinary differential equations. *Appl. Comput. Math* **2019**, *8*, 50–57.
- Butcher, J.C. *Numerical Methods for Ordinary Differential Equations*; John Wiley & Sons: Hoboken, NJ, USA, 2016.
- Rufai, M.A.; Ramos, H. A variable step-size fourth-derivative hybrid block strategy for integrating third-order IVPs, with applications. *Int. J. Comput. Math.* **2022**, *99*, 292–308. [[CrossRef](#)]
- Orakwelu, M.G.; Otegbeye, O.; Mambili-Mamoundou, H. A class of single-step hybrid block methods with equally spaced points for general third-order ordinary differential equations. *J. Niger. Soc. Phys. Sci.* **2023**, *5*, 1484. [[CrossRef](#)]
- Adesanya, A.O.; Udoh, D.M.; Ajileye, A. A new hybrid block method for the solution of general third order initial value problems of ordinary differential equations. *Int. J. Pure Appl. Math.* **2013**, *86*, 365–375. [[CrossRef](#)]
- Areo, E.; Omojola, M. One-Twelfth Step Continuous Block Method. *Int. J. Pure Appl. Math.* **2017**, *114*, 165–178. [[CrossRef](#)]
- Skwame, J.; Dalatu, P.; Sabo, J.; Mathew, M. Numerical application of third derivative hybrid block methods on third Order Initial Value Problem of ordinary differential equations. *Int. J. Stat. Appl. Math.* **2019**, *4*, 90–100.
- Otegbeye, O.; Ansari, M.S. A finite difference based simple iteration method for solving boundary layer flow problems. *AIP Conf. Proc.* **2022**, *2435*, 020055.
- Parsaeitabar, Z.; Nazemi, A. A Third-degree B-spline Collocation Scheme for Solving a Class of the Nonlinear Lane—Emden Type Equations. *Iran. J. Math. Sci. Inform.* **2017**, *12*, 15–34.

22. Motsa, S.S. A new spectral local linearization method for nonlinear boundary layer flow problems. *J. Appl. Math.* **2013**, *2013*, 423628. [[CrossRef](#)]
23. Keller, H.B.; Cebeci, T. Accurate numerical methods for boundary-layer flows. II: Two dimensional turbulent flows. *AIAA J.* **1972**, *10*, 1193–1199. [[CrossRef](#)]
24. Zubik-Kowal, B.; Vandewalle, S. Waveform relaxation for functional-differential equations. *SIAM J. Sci. Comput.* **1999**, *21*, 207–226. [[CrossRef](#)]
25. Burden, A.M.; Burden, R.L.; Faires, J.D. *Numerical Analysis*, 10th ed.; Cengage Learning: Boston, MA, USA, 2016. [[CrossRef](#)]
26. Butcher, J.C. Implicit Runge–Kutta processes. *Math. Comput.* **1964**, *18*, 50–64. [[CrossRef](#)]
27. Lambert, J.D. *Computational Methods in Ordinary Differential Equations*; Wiley: London, UK, 1973.
28. Dahlquist, G. Convergence and stability in the numerical integration of ordinary differential equations. *Math. Scand.* **1956**, *4*, 33–53. [[CrossRef](#)]
29. Awoyemi, D.; Kayode, S.; Adoghe, L. A five-step P-stable method for the numerical integration of third order ordinary differential equations. *Am. J. Comput. Math.* **2014**, *4*, 119–126. [[CrossRef](#)]
30. Allogmany, R.; Ismail, F. Implicit three-point block numerical algorithm for solving third order initial value problem directly with applications. *Mathematics* **2020**, *8*, 1771. [[CrossRef](#)]
31. Adeyeye, O.; Omar, Z. Solving third order ordinary differential equations using one-step block method with four equidistant generalized hybrid points. *Int. J. Appl. Math.* **2019**, *49*, 1–9.
32. Kyriakopoulos, K.J.; Saridis, G.N. Minimum jerk path generation. In Proceedings of the 1988 IEEE International Conference on Robotics and Automation, Philadelphia, PA, USA, 24–29 April 1988; pp. 364–369.
33. Hayati, H.; Eager, D.; Pendrill, A.M.; Alberg, H. Jerk within the context of science and engineering—A systematic review. *Vibration* **2020**, *3*, 371–409. [[CrossRef](#)]
34. Eichhorn, R.; Linz, S.J.; Hänggi, P. Transformations of nonlinear dynamical systems to jerky motion and its application to minimal chaotic flows. *Phys. Rev. E* **1998**, *58*, 7151. [[CrossRef](#)]
35. Eager, D.; Pendrill, A.M.; Reistad, N. Beyond velocity and acceleration: Jerk, snap and higher derivatives. *Eur. J. Phys.* **2016**, *37*, 065008. [[CrossRef](#)]
36. Gottlieb, H. Simple nonlinear jerk functions with periodic solutions. *Am. J. Phys.* **1998**, *66*, 903–906. [[CrossRef](#)]
37. Liu, C.S.; Chang, J.R. The periods and periodic solutions of nonlinear jerk equations solved by an iterative algorithm based on a shape function method. *Appl. Math. Lett.* **2020**, *102*, 106151. [[CrossRef](#)]
38. Gottlieb, H. Harmonic balance approach to periodic solutions of non-linear jerk equations. *J. Sound Vib.* **2004**, *271*, 671–683. [[CrossRef](#)]
39. Mirzabeigy, A.; Yildirim, A. Approximate periodic solution for nonlinear jerk equation as a third-order nonlinear equation via modified differential transform method. *Eng. Comput.* **2014**, *31*, 622–633. [[CrossRef](#)]
40. Kashkari, B.S.; Alqarni, S. Two-step hybrid block method for solving nonlinear jerk equations. *J. Appl. Math. Phys.* **2019**, *7*, 1893. [[CrossRef](#)]
41. Mkhathshwa, M.P. Overlapping Grid Spectral Collocation Methods for Nonlinear Differential Equations Modelling Fluid Flow Problems. Ph.D. Thesis, University of KwaZulu-Natal, Pietermaritzburg, South Africa, 2020.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.