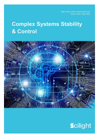
Scilight

Complex Systems Stability & Control
https://www.sciltp.com/journals/cssc



Article

An Optimized Hybrid Two-Step Block Method for Second-Order IVPs

Adeniyi S. Onanaye [1], Adebayo O. Adeniran [2,*] and Bamidele E. Adegbite [3]

[1] Department of Mathematics and Statistics, Redeermer's University, Ede 232101, Nigeria
[2] Department of General Studies (Mathematics Unit), Federal Polytechnic, Ile-Oluji 351110, Nigeria
[3] Department of General Studies Education, Federal College of Education, Iwo 230222, Nigeria
* Correspondence: adeadeniran@fedpolel.edu.ng

How To Cite: Onanaye, A.S.; Adeniran, A.O.; Adegbite, B.E. An Optimized Hybrid Two-Step Block Method for Second-Order IVPs. *Complex Systems Stability & Control* **2026**, *2*(2), 3. https://doi.org/10.53941/cssc.2026.100006

Received: 17 February 2026
Revised: 16 April 2026
Accepted: 17 April 2026
Published: 29 April 2026

Abstract: This article presents the development and analysis of an optimized two-step hybrid block method designed for the direct numerical integration of second-order Initial Value Problems (IVPs) of the form $y'' = f(x, y, y')$. The proposed scheme utilizes Euler's polynomials as basis functions for the approximation, employing interpolation and collocation techniques to derive a continuous linear multistep method. To enhance the accuracy and stability of the block method, an optimization strategy is implemented to determine the optimal placement of two off-step points within the two-step framework. Theoretical analysis confirms that the resulting block method is consistent, zero stable, and convergent, achieving an order of accuracy of seven. The method is implemented as a simultaneous integrator, eliminating the need for predictors or separate starting values. Numerical experiments on moderately and highly stiff problems demonstrate the superior performance of the proposed algorithm, comparisons with existing methods in the literature reveal a significant reduction in absolute error. The results establish the optimized hybrid block method as a robust, efficient, and highly accurate tool for solving complex second-order mathematical models in science and engineering.

Keywords: second-order IVPs; hybrid block method; Euler polynomials; optimization strategy; off-step points; numerical stability

Mathematics Subject Classification (2020): 65L05; 65L20

1. Introduction

Second-order Initial Value Problems (IVPs) are foundational to science and engineering, modeling critical phenomena such as celestial mechanics, circuit oscillations (RLC), and structural vibrations. While some basic models allow for exact mathematical solutions, most real-world applications involve non-linear complexities that require numerical approximation rather than closed-form expressions [1].

Historically, these problems were solved by converting a single second-order equation into a system of two first-order Ordinary Differential Equations (ODEs). While this allows the use of standard tools like Runge-Kutta methods, it often introduces:

Computational Overhead: Increased complexity in the calculation process [2].

Structural Loss: Reduced ability to preserve the physical properties of the original problem, particularly in "stiff" or highly oscillatory systems.

To bypass these hurdles, researchers have shifted toward direct numerical methods. By solving the second-order IVP in its original form, these methods—particularly linear multistep schemes—offer superior efficiency, stability, and accuracy [3–11].

Modern research focuses on two primary enhancements [10,12–17]:

Hybrid Methods: These introduce ”off-step” points between traditional grid marks. This strategic sampling of intermediate data significantly boosts accuracy and stability without a massive increase in computational cost [18].

Optimization Strategies [1,19,20]: Scheme Developers now use algorithms to fine-tune method coefficients. The goal is to minimize truncation errors, maximize the ”region of absolute stability,” and eliminate phase lags in oscillatory data.

The field has seen a surge in adaptive and block-based hybrid methods designed to improve robustness: Gurjinder et al. [19] developed adaptive block methods that adjust step sizes dynamically to balance speed and precision. Rani and Kumar [21] utilized Hermite interpolation within a hybrid block framework to enhance solution reliability. Singh et al. [19] and Kalogiratou & Monovasilis [13] focused on integrating optimization algorithms to find the ideal parameters for step-size control.

This study contributes to the field by designing optimized hybrid two-step solvers. By combining Euler's polynomial functions, interpolation, and collocation with advanced optimization algorithms, the proposed scheme aims to increase accuracy and efficiency for complex engineering models, provide a robust alternative to established solvers and ensure high performance across a diverse array of scientific disciplines. Ultimately, this work seeks to provide more reliable numerical tools for the mathematical models that define our physical world.

The Euler's polynomial, $E_n(x)$, are a sequence of polynomial defined by the generating function:

$$\frac{2e^{xt}}{e^t+1} = \sum_{n=0}^{\infty} E_n(x)\frac{t^n}{n!} \tag{1}$$

This sequence begins with the initial condition:

$$E_0 = 1$$

The primary focus of this numerical analysis is the Second-Order Initial Value Problem(IVP). This type of equation models systems where the acceleration (second derivative) depends on the independent variable, the position, and the velocity. It is expressed as:

$$y''(x) = f(x,\ y(x),\ y'(x)) \tag{2}$$

To obtain a unique solution over the interval $a < x < b$, the system requires two specific initial conditions at the starting point a:

Initial Position: $y(a) = y_a$

Initial Velocity: $y'(a) = y'_a$

2. Derivation of the Scheme

This section details the derivation of a fixed-step two-step hybrid block method. To begin, we transition from the continuous domain $[a, b]$ of the original problem to a discrete set of interest points, defined as:

$$a = x_0 < x_1 < x_2 < x_3 < \cdots < x_n = b.$$

The discretization uses a constant step-size, h, determined by the relation:

$$h = x_{n+1} - x_n, \quad n = 0, 1, 2, \ldots, N-1.$$

To construct the numerical scheme, we approximate the exact solution over the interval $[a, b]$ using the Euler's polynomial defined previously in Equation (1). This approximation serves as the basis for the interpolation and collocation techniques used to develop the block method. Degree of polynomial $=$ number of interpolation point+ number of collocation points-1.

$$y(x) \approx P(x) = \sum_{n=0}^{6} a_n E_n(x) \tag{3}$$

Differentiating Equation (3) twice, we have

$$y''(x) \approx f(x, y, y') = \sum_{n=0}^{6} a_n \frac{d^2\left(E_n(x)\right)}{dx^2} \tag{4}$$

where $a_n \in \Re$ are real unknown coefficients that will be uniquely determined. In this article, we shall consider two off grid points $x_{n+d} = x_n + dh$, where $0 < d < 1$ and $x_{n+c} = x_n + ch$, $1 < c < 2$.

In order obtain the value of the unknown coefficients of the exact solution, the following conditions were imposed:

Interpolation:

$$P(x_n) = y_n, \quad P(x_{n+1}) = y_{n+1} \tag{5}$$

Collocation:

$$P''(x_n) = f_n, \quad P''(x_{n+d}) = f_{n+d}, \quad P''(x_{n+1}) = f_{n+1}, \quad P''(x_{n+c}) = f_{n+c} \quad P''(x_{n+2}) = f_{n+2} \tag{6}$$

Combining Equations (5) and (6), we arrive at system of seven equations with seven unknowns a_n, $n = 0, 1, \cdots, 6$, this equations are solved using computer aided system Maple 18. The values of $a_n's$ are cumbersome and are not included in the articles. Inserting this obtained coefficient into (3), we have:

$$y(x) = \alpha_0 y_n + \alpha_1 y_{n+1} + h^2\left[\beta_0 f_n + \beta_d f_{n+d} + \beta_1 f_{n+1} + \beta_c f_{n+c} + \beta_2 f_{n+2}\right], \tag{7}$$

where

$$\alpha_0 = -x + 1$$

$$\alpha_1 = x$$

$$\beta_0 = \frac{(x-1)\left(5\,dcx^2 - 3\,dx^3 - 3\,cx^3 + 2\,x^4 - 25\,dcx + 12\,dx^2 + 12\,cx^2 - 7\,x^3 + 35\,dc - 8\,dx - 8\,cx + 3\,x^2 - 8\,d - 8\,c + 3\,x + 3\right)x}{120\,dc}$$

$$\beta_d = \frac{2\,x^6 + (-3\,c - 9)\,x^5 + (15\,c + 10)\,x^4 - 20\,cx^3 + (8\,c - 3)\,x}{(60\,d - 60\,c)\,(d-1)\,(d-2)\,a}$$

$$\beta_1 = \frac{-2\,x^6 + (3\,d + 3\,c + 6)\,x^5 + ((-5\,c - 10)\,d - 10\,c)\,x^4 + 20\,dcx^3 + ((-15\,c + 7)\,d + 7\,c - 4)\,x}{(60\,c - 60)\,(d-1)}$$

$$\beta_c = \frac{\left(3\,x^5 - 15\,x^4 + 20\,x^3 - 8\,x\right)d - 2\,x^6 + 9\,x^5 - 10\,x^4 + 3\,x}{(60\,c - 60)\,(c-2)\,(d-c)\,c}$$

$$\beta_2 = \frac{2\,x^6 + (-3\,d - 3\,c - 3)\,x^5 + ((5\,c + 5)\,d + 5\,c)\,x^4 - 10\,dcx^3 + ((5\,c - 2)\,d - 2\,c + 1)\,x}{(120\,c - 240)\,(d-2)}$$

Evaluating (7) at $x = x_{n+2}$ (at $x = 2$) and simplifying yields

$$y_{n+2} = -y_n + 2y_{n+1} + h^2[\frac{1}{60}\left(\frac{5dc-3}{dc}\right)f_n - \left(\frac{1}{10(d-1)(d-2)(d-2)d}\right)f_{n+d} + \frac{1}{30}\left(\frac{25dc-25d-25c+28}{(d-1)(d-c)}\right)f_{n+1} +$$

$$[\frac{1}{10}\left(\frac{1}{c(d-c)(c-1)(c-2)}\right)f_{n+c} + \frac{1}{10}\left(\frac{5dc-10d-10c+17}{(d-2)(c-2)}\right)f_{n+2}] \tag{8}$$

Equation (8) depend on parameters d and c, which is related to the off-step point x_{n+d} and x_{n+c}. The goal is to obtain appropriate value for the parameter d and c, for its purpose, we adopt the following optimization strategy (Ramos et al. [1]).

Rearranging (8), we have;

$$y_{n+2} + y_n - 2y_{n+1} - h^2\left[\frac{1}{60}\left(\frac{5dc-3}{dc}\right)f_n - \left(\frac{1}{10(d-1)(d-2)(d-2)d}\right)f_{n+d} + \frac{1}{30}\left(\frac{25dc-25d-25c+28}{(d-1)(d-c)}\right)f_{n+1} - \right]$$

$$h^2\left[\frac{1}{10}\left(\frac{1}{c(d-c)(c-1)(c-2)}\right)f_{n+c} + \frac{1}{10}\left(\frac{5dc-10d-10c+17}{(d-2)(c-2)}\right)f_{n+2}\right] = 0 \tag{9}$$

Expanding each term of Equation (9) using the Taylor series expansion gives;

$$y_{n+2} = y_n + 2hy'_n + \frac{(2h)^2}{2!}y''_n + \frac{(2h)^3}{3!}y'''_n + \frac{(2h)^4}{4!}y''''_n + \cdots$$

$$-y_n = -[y_n]$$

$$-2y_{n+1} = -2\left[y_n + hy'_n + \frac{(h)^2}{2!}y''_n + \frac{(h)^3}{3!}y'''_n + \frac{(h)^4}{4!}y''''_n + \cdots\right]$$

$$-h^2\frac{1}{60}\left(\frac{5dc-3}{dc}\right)f_n = --h^2\frac{1}{60}\left(\frac{5dc-3}{dc}\right)y''_n$$

$$-h^2\left(\frac{1}{10(d-1)(d-2)(d-2)d}\right)f_{n+d} = -\left(\frac{1}{10(d-1)(d-2)(d-2)d}\right)\left[y''_n + dhy'''_n + \frac{(dh)^2}{2!}y''''_n + \frac{(dh)^3}{3!}y'''''_n + \cdots\right]$$

$$h^2\frac{1}{30}\left(\frac{25dc-25d-25c+28}{(d-1)(d-c)}\right)f_{n+1}=h^2\frac{1}{30}\left(\frac{25dc-25d-25c+28}{(d-1)(d-c)}\right)\left[y_n''+hy_n'''+\frac{(h)^2}{2!}y_n''''+\frac{(h)^3}{3!}y_n'''''+\cdots\right]$$

$$h^2\frac{1}{10}\left(\frac{1}{c(d-c)(c-1)(c-2)}\right)f_{n+c}=h^2\frac{1}{10}\left(\frac{1}{c(d-c)(c-1)(c-2)}\right)f_{n+c}\left[y_n''+chy_n'''+\frac{(ch)^2}{2!}y_n''''+\frac{(ch)^3}{3!}y_n'''''+\cdots\right]$$

$$h^2\frac{1}{10}\left(\frac{5dc-10d-10c+17}{(d-2)(c-2)}\right)f_{n+2}=h^2\frac{1}{10}\left(\frac{5dc-10d-10c+17}{(d-2)(c-2)}\right)\left[y_n''+2hy_n'''+\frac{(2h)^2}{2!}y_n''''+\frac{(2h)^3}{3!}y_n'''''+\cdots\right]$$

Comparing each of the coefficient above , we have;

$$h^0y_n:1-1=0$$

$$h^1y_n':2-2=0$$

$$h^2y_n'':\frac{2^2}{2!}-2(\frac{1^2}{2!})-\frac{1}{60}\left(\frac{5dc-3}{dc}\right)-\left(\frac{1}{10(d-1)(d-2)(d-2)d}\right)+\frac{1}{30}\left(\frac{25dc-25d-25c+28}{(d-1)(d-c)}\right)+$$
$$\frac{1}{10}\left(\frac{1}{c(d-c)(c-1)(c-2)}\right)+\frac{1}{10}\left(\frac{5dc-10d-10c+17}{(d-2)(c-2)}\right)=0$$

$$h^3y_n''':\frac{2^3}{3!}-2(\frac{1^3}{3!})-\frac{1}{60}\left(\frac{5dc-3}{dc}\right)(0)-\left(\frac{1}{10(d-1)(d-2)(d-2)d}\right)(d)+\frac{1}{30}\left(\frac{25dc-25d-25c+28}{(d-1)(d-c)}\right)(1)+$$
$$\frac{1}{10}\left(\frac{1}{c(d-c)(c-1)(c-2)}\right)(c)+\frac{1}{10}\left(\frac{5dc-10d-10c+17}{(d-2)(c-2)}\right)(2)=0$$

$$h^4y_n'''':\frac{2^4}{4!}-2(\frac{1^4}{4!})-\left(\frac{1}{10(d-1)(d-2)(d-2)d}\right)(\frac{d^2}{2!})+\frac{1}{30}\left(\frac{25dc-25d-25c+28}{(d-1)(d-c)}\right)(\frac{1^2}{2!})+$$
$$\frac{1}{10}\left(\frac{1}{c(d-c)(c-1)(c-2)}\right)(\frac{c^2}{2!})+\frac{1}{10}\left(\frac{5dc-10d-10c+17}{(d-2)(c-2)}\right)(\frac{2^2}{2!})=0$$

$$h^5y_n''''':\frac{2^5}{5!}-2(\frac{1^5}{5!})-\left(\frac{1}{10(d-1)(d-2)(d-2)d}\right)(\frac{d^3}{3!})+\frac{1}{30}\left(\frac{25dc-25d-25c+28}{(d-1)(d-c)}\right)(\frac{1^3}{3!})+$$
$$\frac{1}{10}\left(\frac{1}{c(d-c)(c-1)(c-2)}\right)(\frac{c^3}{3!})+\frac{1}{10}\left(\frac{5dc-10d-10c+17}{(d-2)(c-2)}\right)(\frac{2^3}{3!})=0$$

$$h^6y_n'''''':\frac{2^6}{6!}-2(\frac{1^6}{6!})-\left(\frac{1}{10(d-1)(d-2)(d-2)d}\right)(\frac{d^4}{4!})+\frac{1}{30}\left(\frac{25dc-25d-25c+28}{(d-1)(d-c)}\right)(\frac{1^4}{4!})+$$
$$\frac{1}{10}\left(\frac{1}{c(d-c)(c-1)(c-2)}\right)(\frac{c^4}{4!})+\frac{1}{10}\left(\frac{5dc-10d-10c+17}{(d-2)(c-2)}\right)(\frac{2^4}{4!})=0$$

$$h^7y_n''''''':\frac{2^7}{7!}-2(\frac{1^7}{7!})-\left(\frac{1}{10(d-1)(d-2)(d-2)d}\right)(\frac{d^5}{5!})+\frac{1}{30}\left(\frac{25dc-25d-25c+28}{(d-1)(d-c)}\right)(\frac{1^5}{5!})+$$
$$\frac{1}{10}\left(\frac{1}{c(d-c)(c-1)(c-2)}\right)(\frac{c^5}{5!})+\frac{1}{10}\left(\frac{5dc-10d-10c+17}{(d-2)(c-2)}\right)(\frac{2^5}{5!})\neq 0$$

Thus

$$C_0=C_1=C_2=C_3=C_4=C_5=C_6=0,\ \ C_7\neq 0$$

$$L\left[y(x_{n+2}),h\right]=\left(\frac{d}{1200}+\frac{c}{1200}-\frac{1}{600}\right)y_n'''''''h^7+O(h^8) \tag{10}$$

To include the initial condition of Equation (2), we take the derivative of the continuous method (7) with respect to x

$$y'(x) = \alpha'_0 y_n + \alpha'_1 y_{n+1} + h^2 \left[\beta'_0 f_n + \beta'_d f_{n+d} + \beta'_1 f_{n+1} + \beta'_c f_{n+c} + \beta'_2 f_{n+2}\right], \tag{11}$$

Evaluate (11) at $x = x_{n+2}$ (at $x = 2$),

$$y'_{n+2} = -y_n + y_{n+1} + h^2[\frac{1}{20}\left(\frac{5dc + 8d + 8c - 19}{dc}\right) f_n - \left(\frac{9c - 19}{60(c-1)(c-2)(d-c)d}\right) f_{n+d} +$$

$$\frac{1}{60}\left(\frac{65dc - 78d - 73c + 92}{(c-1)(d-1)}\right) f_{n+1} - \left(\frac{8d - 19}{60(c-1)(d-2)(c-d)c}\right) f_{n+c} + \frac{1}{120}\left(\frac{45dc - 82d - 82c + 145}{(d-2)(c-2)}\right)] \tag{12}$$

In a similar fashion in which Equation (10) was obtained from Equation (9), we obtain the local truncation error for Equation (12) as

$$L\left[y(x_{n+1}), h\right] = \left[\frac{1}{50400}(-56d + 133)c + \frac{19}{7200}d - \frac{13}{2800}\right] y_n h^7 + O(h^8) \tag{13}$$

By forcing the leading terms in Equations (10) and (13) to equal zero, we obtain the following system of nonlinear equations.

$$\frac{d}{1200} + \frac{c}{1200} - \frac{1}{600} = 0$$

$$\frac{1}{50400}(-56d + 133)c + \frac{19}{7200}d - \frac{13}{2800} = 0$$

Because the implicit system in above equations describes a curve symmetrical around the diagonal, we can expect a unique solution for a given set of constraints. $0 < d < 1$, $1 < c < 2$

$$d = 1 - \frac{1}{7}\sqrt{21}$$

$$c = 1 + \frac{1}{7}\sqrt{21}$$

Inserting d and c into (8) gives;

$$y_{n+2} = -y_n + 2y_{n+1} + h^2\left[-\frac{1}{240}f_n + \frac{49}{240}f_{n+1-\frac{1}{7}\sqrt{21}} + \frac{3}{5}f_{n+1} + \frac{49}{240}f_{n+1+\frac{1}{7}\sqrt{21}} - \frac{1}{240}f_{n+2}\right] \tag{14}$$

Also evaluating (7) at $x = x_n + 1 - \frac{1}{7}\sqrt{21}$ $(i.e.,\ x = 1 - \frac{1}{7}\sqrt{21})$ and $x = x_n + 1 + \frac{1}{7}\sqrt{21}(\ x = 1 + \frac{1}{7}\sqrt{21})$

$$y_{n+1-\frac{1}{7}\sqrt{21}} = -15\,\frac{\left(-168\sqrt{21} - 1176\right)\sqrt{21}}{123480 + 17640\sqrt{21}} y_n + \frac{70560}{123480 + 17640\sqrt{21}} y_{n+1} - h^2 \frac{15(\sqrt{21} + \frac{8}{5})\sqrt{21}}{123480 + 17640\sqrt{21}} f_n +$$

$$h^2\left(-\frac{15(\frac{77}{3}\sqrt{21} + \frac{728}{5})\sqrt{21}}{123480 + 17640\sqrt{21}} f_{n+1-\frac{1}{7}\sqrt{21}} - \frac{-15(16\sqrt{21} + \frac{768}{5})\sqrt{21}}{123480 + 17640\sqrt{21}} f_{n+1} - \frac{15(-\frac{35}{3}\sqrt{21} + \frac{1688}{5})\sqrt{21}}{123480 + 17640\sqrt{21}} f_{n+1+\frac{1}{7}\sqrt{21}} - \right)$$

$$h^2 \frac{-15(\sqrt{21} + \frac{8}{5})\sqrt{21}}{123480 + 17640\sqrt{21}} f_{n+2} \tag{15}$$

$$y_{n+1+\frac{1}{7}\sqrt{21}} = 15\,\frac{\left(-168\sqrt{21} - 1176\right)\sqrt{21}}{-123480 + 17640\sqrt{21}} y_n - \frac{70560}{-123480 + 17640\sqrt{21}} y_{n+1} + h^2 \frac{15(\sqrt{21} - \frac{8}{5})\sqrt{21}}{-123480 + 17640\sqrt{21}} f_n +$$

$$h^2\left(-\frac{15(-\frac{35}{3}\sqrt{21} - \frac{168}{5})\sqrt{21}}{-123480 + 17640\sqrt{21}} f_{n+1-\frac{1}{7}\sqrt{21}} + \frac{-15(-16\sqrt{21} + \frac{768}{5})\sqrt{21}}{-123480 + 17640\sqrt{21}} f_{n+1} + \frac{15(-\frac{77}{3}\sqrt{21} - \frac{728}{5})\sqrt{21}}{-123480 + 17640\sqrt{21}} f_{n+1+\frac{1}{7}\sqrt{21}} + \right)$$

$$h^2 \frac{15(\sqrt{21} - \frac{8}{5})\sqrt{21}}{-123480 + 17640\sqrt{21}} f_{n+2} \tag{16}$$

Inserting the value of d and c into Equation (12) gives

$$hy'_{n+2} = -y_n + y_{n+1} + h^2\left[-\frac{1}{480}f_n + \left(-\frac{7}{180}\sqrt{21} + \frac{539}{1440}\right)f_{n+1-\frac{1}{7}\sqrt{21}} + \frac{59}{90}f_{n+1} + \left(\frac{7}{180}\sqrt{21} + \frac{539}{1440}\right)f_{n+1+\frac{1}{7}\sqrt{21}} + \right.$$
$$\left. \frac{47}{480}f_{n+2}\right] \qquad (17)$$

Also evaluating (11) at $x = x_n(i.e., \ at \ x = 0)$, $x = x_n + 1 - \frac{1}{7}\sqrt{21}$ $(i.e., \ x = 1 - \frac{1}{7}\sqrt{21})$ and $x = x_n + 1 + \frac{1}{7}\sqrt{21}(x = 1 + \frac{1}{7}\sqrt{21})$

$$hy'_n = -y_n + y_{n+1} + h^2\left[-\frac{49}{480}f_n + \left(-\frac{7}{180}\sqrt{21} - \frac{49}{288}\right)f_{n+1-\frac{1}{7}\sqrt{21}} - \frac{1}{18}f_{n+1} + \left(\frac{7}{180}\sqrt{21} - \frac{49}{288}\right)f_{n+1+\frac{1}{7}\sqrt{21}} - \right.$$
$$\left. \frac{1}{480}f_{n+2}\right] \qquad (18)$$

$$hy'_{n++1-\frac{1}{7}\sqrt{21}} = -y_n + y_{n+1} + h^2\left[\left(\frac{3}{980} + \frac{13}{672}\right)f_n + \left(-\frac{29}{630}\sqrt{21} + \frac{49}{480}\right)f_{n+1-\frac{1}{7}\sqrt{21}} + \left(-\frac{64}{735}\sqrt{21} + \frac{3}{10}\right)f_{n+1} + \right]$$
$$h^2\left[\left(-\frac{1}{63}\sqrt{21} + \frac{49}{480}\right)f_{n+1+\frac{1}{7}\sqrt{21}} + \left(\frac{3}{980}\sqrt{21} - \frac{79}{3360}\right)f_{n+2}\right] \qquad (19)$$

$$hy'_{n+1} = -y_n + y_{n+1} + h^2\left[-\frac{1}{480}f_n + \left(\frac{1}{2073600}\sqrt{21} + \frac{49}{480}\right)f_{n+1-\frac{1}{7}\sqrt{21}} + \frac{3}{10}f_{n+1} + \right]$$
$$h^2\left[\left(-\frac{1}{2073600}\sqrt{21} + \frac{49}{480}\right)f_{n+1+\frac{1}{7}\sqrt{21}} + \frac{1}{60}f_{n+2}\right] \qquad (20)$$

$$hy'_{n++1+\frac{1}{7}\sqrt{21}} = -y_n + y_{n+1} + h^2\left[\left(-\frac{3}{980} + \frac{13}{672}\right)f_n + \left(\frac{1}{61}\sqrt{21} + \frac{49}{480}\right)f_{n+1-\frac{1}{7}\sqrt{21}} + \left(\frac{64}{735}\sqrt{21} + \frac{3}{10}\right)f_{n+1} + \right]$$
$$h^2\left[\left(\frac{29}{630}\sqrt{21} + \frac{49}{480}\right)f_{n+1+\frac{1}{7}\sqrt{21}} + \left(-\frac{3}{980}\sqrt{21} - \frac{79}{3360}\right)f_{n+2}\right] \qquad (21)$$

Equations (14)–(21) will be combined as a block, translate into Computer code and used for solving numerical problems in Section 5.

3. Characteristics of the Scheme

3.1. Order and Error Constant

Let $y(t)$ be a sufficiently differentiable function. Consider the following difference operator associated with the hybrid block method given in

$$L[y(x):h] = \sum_{j=0}^{k}\left[\alpha_j y(x_{n+j}) - h^2\beta_j y''(x_{n+j})\right] \qquad (22)$$

where $\alpha_k = 1$, α_0 and $\beta - 0$ are not both zero. $y(x)$ is an arbitrary function which is continuously differentiable on the interval $[a, b]$. In a similar way as order and error constant of (8) was obtained, the order and error constant of our method ((14)–(21)) were obtained and are presented in the Table 1.

Table 1. Order and Error Constants of the methods.

Method Eqn. No.	Order	Error Constant
(14)	7	$\frac{1}{3590\sqrt{2}}$
(15)	7	$\frac{1}{7200\sqrt{2}}$
(16)	7	$\frac{2}{152100}$
(17)	7	$\frac{11}{7651000}$
(18)	7	$\frac{1}{10200}$
(19)	7	$\frac{13}{5217000}$
(20)	7	$-\frac{7}{144000}$
(21)	7	$\frac{1}{2520000}$

3.2. Consistency of the Method

Definition 1. *According to Adeniran et al. [6] and Ramos et al. [13]. A block method is said to be consistent if it has an order of convergence, say* $\rho \geq 1$.

The block method is consistent as our method are $\rho \geq 1$

3.3. Zero Stability

Definition 2. *According to Lambert [22], a linear multistep method is zero-stable if no root of its first characteristic polynomial,* $\rho(\lambda)$*, has a modulus greater than one, and every root with a modulus of one is simple (has a multiplicity of one).*

To analyze the zero stability of our block method, we consider the limit as $h \to 0$. The first characteristic polynomial is defined by:

$$\rho(\lambda) = \det\left(\lambda A^{(0)} - A^{(1)}\right), \tag{23}$$

where the matrices $A^{(0)}$ and $A^{(1)}$ are given by:

$$A^{(0)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad A^{(1)} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Substituting these into (23), we obtain:

$$\rho(\lambda) = \det\begin{pmatrix} \lambda & 0 & -1 \\ 0 & \lambda & -1 \\ 0 & 0 & \lambda - 1 \end{pmatrix} = \lambda(\lambda^2 - \lambda)$$

Setting $\rho(\lambda) = 0$ yields the roots $\lambda = 0$, $\lambda = 1$ and $\lambda = 1$. Since $|\lambda| \leq 1$ and the root with modulus one is simple, the method is zero-stable.

3.4. Convergence

Theorem 1 (Fundamental Theorem of Dahlquist Henrici [23]). *The necessary and sufficient condition for a multistep method to be convergent is for it to be consistent and zero stable.*

Thus the block method is convergent, since zero stability + consistency $\equiv$ Convergence.

3.5. Region of Absolute Stability

In this section, we analyze the region of absolute stability of the proposed Optimized Hybrid Two-Step Block Method defined by Equations (14)–(21). Since the scheme is formulated as a block hybrid with off-step points $x_{n+1-\frac{1}{7}}\sqrt{21}$ and $x_{n+1+\frac{1}{7}}\sqrt{21}$, its stability behavior is examined using the standard linear test equation (see [7,19,24–26] for details).

Consider the second order linear test problem:

$$y'' = \lambda y, \quad \lambda \in \mathbb{C} \tag{24}$$

Let

$$z = \lambda h^2$$

Substituting $f = \lambda y$ into Equations (14)–(21), the full block scheme can be written in matrix form as:

$$Y_{n+1} = M(z) Y_n \tag{25}$$

where

$$Y_n = \begin{pmatrix} y_n \\ y_{n-1} \end{pmatrix}, \quad M(z) \in \mathbb{C}^{2\times 2}$$

is the amplification matrix obtained by collecting coefficient of y_n and y_{n-1} after eliminating intermediate stages.

3.5.1. Stability Matrix Construction

From Equations (14)–(21), the main step relation(14) provides:

$$y_{n+2} = \alpha_1(z)y_{n+1} + \alpha_2(z)y_n \tag{26}$$

where the coefficients $\alpha_1(z)$ and $\alpha_2(z)$ are rational functions of z, incorporating contributions from all hybrid evaluations, thus the amplification method can be expressed as:

$$\begin{pmatrix} y_{n+2} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} \alpha_1(z) & \alpha_2(z) \\ 1 & 0 \end{pmatrix} \begin{pmatrix} y_{n+1} \\ y_n \end{pmatrix}.$$

Hence the amplification matrix is

$$M(z) = \begin{pmatrix} \alpha_1(z) & \alpha_2(z) \\ 1 & 0 \end{pmatrix}$$

3.5.2. Stability Polynomial (Periodic Solution)

The corresponding stability(Characteristic) polynomial is

$$\zeta^2 - \alpha_1(z)\zeta - \alpha_2(z) = 0 \tag{27}$$

Let the roots be $\zeta_1(z)$, $\zeta_2(z)$, these are the amplification factors of the method.

Definition 3. *The region of absolute stability is defined as:*

$$\Re = \{z \in \mathbb{C} : |\zeta_1(z)| \leq 1, \;\; |\zeta_2(z)| \leq 1, \}$$

with the additional condition that for multiple roots:

$$|\zeta(z)| < 1,$$

this ensure boundedness of the numerical solution as $n \to \infty$.

3.5.3. Boundary Locus Method

To compute the boundary of $\Re$, we apply the boundary locus by setting

$$\zeta = e^{i\theta}, \quad \theta \in [0, \; 2\pi] \, ,$$

substituting into the stability polynomial

$$e^{2i\theta} - \alpha_1(z)e^{i\theta} - \alpha_2(z)e^{i\theta} = 0 \tag{28}$$

Solving for z in (28) gives

$$z(\theta) = \frac{\rho(e^{i\theta})}{\sigma(e^{i\theta})},$$

where the numerator and denominator arises from the structure of the Equations (14)–(21).

3.5.4. Stability Region Plot

The stability region is obtained numerically by evaluating $z(\theta)$ over $\theta \in [0, 2\pi]$.

The region of absolute stability of the proposed method derived from Equations (14)–(21) is displayed in Figure 1. The method exhibit a wide stability interval along the negative real axis, indicating suitability for moderately stiff problems.

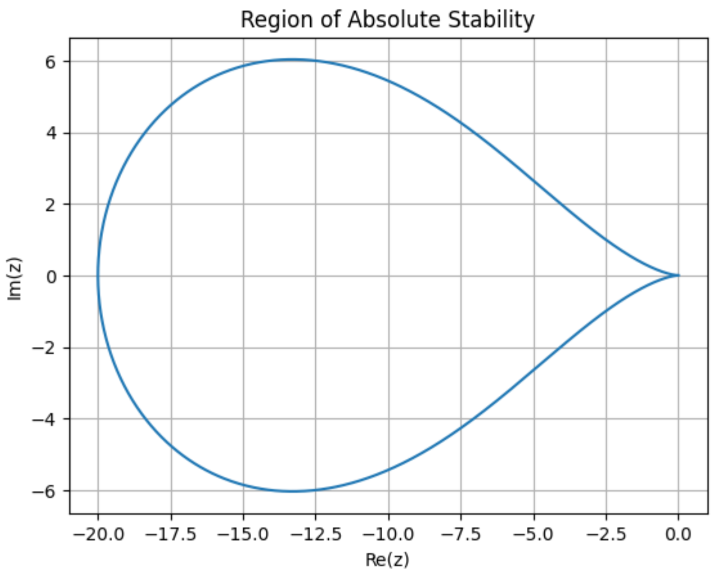


Figure 1. Region of absolute stability.

4. Implementation of the Method

The derived method is efficiently implemented by combining Equations (14)–(21) into a simultaneous integrator for Initial Value Problems (IVPs) of second order, eliminating the need for starting values or predictors. A computer program was developed in Maple 18 to solve selected test problems. To evaluate the method's accuracy, step sizes consistent with those found in the literature were used, and the results were compared against existing methods. The performance is assessed using the absolute error, defined as:

$$\text{Error(err)} = |y_{exact} - y_{computed}|$$

where y_{exact} is the analytical solution and y_{computed} is the numerical solution obtained using the proposed method (Equations (14)–(21)).

5. Numerical Examples

In order to study the efficiency of the developed method, we present numerical experiments with the following examples.

Example 1. *We consider a moderately stiff problem*

$$y''(t) = y', \quad y(0) = 0, \quad y'(0) = -1$$

Exact solution: $y(t) = 1 - e^t$.
Table 2 showing the exact,computed value for Example 1 with $h = 0.1$.

Table 2. Computation data for Example 1.

x	y_{exact}	$y_{computed}$	Err	Yah Err [27]
0.1	-0.1051709180756476248	$-0.1051709180753619036\,3$	$2.8572117 \times 10^{-13}$	8.7×10^{-05}
0.2	-0.2214027581601698339	$-0.2214027581601451862\,8$	2.464762×10^{-14}	3.2×10^{-04}
0.3	-0.3498588075760031040	$-0.3498588075756268834\,9$	$3.7622051 \times 10^{-13}$	2.2×10^{-03}
0.4	-0.4918246976412703178	$-0.4918246976412101084\,4$	6.020936×10^{-14}	4.9×10^{-03}
0.5	-0.6487212707001281468	$-0.6487212706996353592\,8$	$4.9278752 \times 10^{-13}$	9.1×10^{-03}
0.6	-0.8221188003905089749	$-0.8221188003903986650\,2$	$1.1030988 \times 10^{-13}$	1.4×10^{-02}
0.7	-1.0137527074704765216	$-1.0137527074698339924\,0$	6.425292×10^{-13}	2.1×10^{-02}
0.8	-1.2255409284924676046	$-1.2255409284922879609\,0$	1.796437×10^{-13}	2.9×10^{-02}
0.9	-1.4596031111569496638	$-1.4596031111561152427\,0$	8.344211×10^{-13}	4.0×10^{-02}
1.0	-1.7182818284590452354	$-1.7182818284587709638\,0$	2.742716×10^{-13}	5.2×10^{-02}

The results presented in Table 2 provide a comparative analysis of the proposed numerical method against the exact solution and the existing literature by Yahaya and Badmus ([27]) (Yah). The computed values ($y_{computed}$) show a remarkably high degree of agreement with the exact solution (y_{exact}) across the entire interval $x \in [0.1, 1.0]$. With a step size of $h = 0.1$, the proposed method maintains an absolute error (err) consistently within the range of 10^{-13} to 10^{-14} for the majority of the steps.

The primary observation from the data is the significant improvement in accuracy achieved by the current method. While the exact solution values (y_{exact}) and the computed values ($y_{computed}$) align closely across the entire interval $x \in [0.1, 1.0]$, the error analysis reveals a stark contrast. The computed error (err) for the proposed method consistently stays within the range of 10^{-13} to 10^{-14}. In comparison, the error recorded by Yahaya and Badmus [27] ranges from 10^{-5} to 10^{-2}. Our method exhibits a highly stable error profile. For instance, at $x = 1.0$, the proposed method maintains an error of 2.742716×10^{-13}, whereas the Yahaya and Badmus error grows significantly to 5.2×10^{-2}.

A notable characteristic of the Yahaya and Badmus [27] results is the rapid accumulation of error as x increases, moving from 10^{-5} at the start to 10^{-2} at the boundary. In contrast, the proposed method does not show a strictly increasing error trend; instead, it fluctuates slightly within a very tight tolerance. This suggests that the proposed scheme possesses superior numerical stability and is less susceptible to the accumulation of truncation or rounding errors over successive steps.

Example 2. *We consider a highly stiff initial value problem*

$$y''(t) = -1001y' - 1000y, \;\; y(0) = 0, \;\; y'(0) = -1$$

Exact solution: $y(t) = e^{-t}$
Table 3 showing the exact,computed value for Example 2 with $h = 0.1$.

Table 3. Computation data for Example 2.

t	y_{exact}	$y_{computed}$	**Err**	**Ade1 Err [2]**
0.1	0.90483741803595957316	0.90483741803596033597	7.6281×10^{-16}	2.90×10^{-09}
0.2	0.81873075307798185867	0.81873075307798306425	1.20558×10^{-15}	1.87×10^{-08}
0.3	0.74081822068171786607	0.74081822068171930403	1.43796×10^{-15}	9.97×10^{-08}
0.4	0.67032004603563930074	0.67032004603564083467	1.53393×10^{-15}	5.25×10^{-07}
0.5	0.60653065971263342360	0.60653065971263496679	1.54319×10^{-15}	2.75×10^{-07}
0.6	0.54881163609402643263	0.54881163609402793164	1.49901×10^{-15}	1.44×10^{-06}
0.7	0.49658530379140951470	0.49658530379141093823	1.42353×10^{-15}	7.50×10^{-06}
0.8	0.44932896411722159143	0.44932896411722292270	1.33127×10^{-15}	3.92×10^{-05}
0.9	0.40656965974059911188	0.40656965974060034360	1.23172×10^{-15}	2.04×10^{-04}
1.0	0.36787944117144232160	0.36787944117144345249	1.13089×10^{-15}	1.06×10^{-03}

The numerical results presented in Table 3 provide a comparative analysis between the exact solution, our proposed computational method, and the results obtained by Adeniran et al. [2] (Ade1). The evaluation is conducted over the interval $t \in [0.1, 1.0]$ *with a fixed step size of* $h = 0.1$.

Our computed values ($y_{computed}$) show an exceptionally high degree of agreement with the exact solution (y_{exact}). The absolute errors produced by our method remain consistently within the magnitude of 10^{-15} to 10^{-16}. A direct comparison between our computed error (err) and the error reported in the literature (Ade1 err) reveals that our method significantly outperforms the existing approach. While Ade1 errors range from 10^{-9} to 10^{-3}, our method maintains a much tighter error margin.

Our method's error remains relatively stable as t increases, peaking slightly at $t = 0.5$ (1.54319×10^{-15}) before gradually decreasing.In contrast, the error in the Adeniran et al. [2] model grows exponentially as t approaches 1.0, reaching a maximum of 1.06×10^{-03}.

The results demonstrate that the proposed method is not only convergent but also highly efficient in minimizing truncation and round-off errors compared to existing literature. The improvement in precision—roughly six to twelve orders of magnitude better than Ade1—validates the superior performance of our algorithm for Example 2.

Example 3. *We consider the initial value problem*

$$y''(t) = -100t + 99sin(t), \;\; y(0) = 1, \;\; y'(0) = 11$$

Exact solution: $y(t) = \cos(10t) + \sin(10t) + \sin(t)$
Table 4 showing the exact, computed value for Example 3 with $h = \frac{1}{320}$.

Table 4. Computation data for Example 3.

t	y_{exact}	$y_{computed}$	Err	Ade Err [2]
$\frac{1}{320}$	1.0338816673842019107	1.0338816673842019181	7.4×10^{-18}	2.00×10^{-14}
$\frac{2}{320}$	1.0667567878524546543	1.0667567878524546538	5.0×10^{-19}	6.00×10^{-14}
$\frac{3}{320}$	1.0985962803650165721	1.0985962803650165794	7.3×10^{-18}	1.30×10^{-13}
$\frac{4}{320}$	1.1293720750962665318	1.1293720750962665308	1.0×10^{-18}	2.10×10^{-13}
$\frac{5}{320}$	1.1590571408149113575	1.1590571408149113649	7.4×10^{-18}	3.60×10^{-13}
$\frac{6}{320}$	1.1876255112500243853	1.1876255112500243838	1.5×10^{-18}	5.40×10^{-13}
$\frac{7}{320}$	1.2150523104171694481	1.2150523104171694554	7.3×10^{-18}	7.40×10^{-13}
$\frac{8}{320}$	1.2413137768798800444	1.2413137768798800425	1.9×10^{-18}	9.60×10^{-13}
$\frac{9}{320}$	1.2663872869228030575	1.2663872869228030647	1.8×10^{-18}	1.24×10^{-12}
$\frac{10}{320}$	1.2902513766138791315	1.2902513766138791291	2.4×10^{-18}	1.55×10^{-12}

The numerical results presented in Table 4 provide a comparative analysis between the exact solution, our proposed computational method ($y_{computed}$*), and the existing results established by Adeniran et al. [2] for Example 3 using a step size of* $h = \frac{1}{320}$.

The primary observation from the data is the high level of precision achieved by our computed values. Throughout the interval $t \in [\frac{1}{320}, \frac{10}{320}]$*, the absolute errors (err) remain exceptionally low, fluctuating within the range of* 10^{-18} *to* 10^{-19}*. The maximum recorded error for our method is approximately* 7.4×10^{-18}*.The minimum recorded error reaches an impressive* 5.0×10^{-19} *at* $t = \frac{2}{320}$*. These values demonstrate that our method captures the exact solution with near-machine precision, indicating strong numerical stability and convergence at this step size.*

While the error in the literature (Ade err) grows steadily as t *increases moving from* 10^{-14} *to* 10^{-12}*, our method maintains a consistently lower error profile, even at* $t = \frac{10}{320}$*, our error remains in the* 10^{-18} *magnitude, whereas the reference error has increased by two orders of magnitude.*

The results confirm that our proposed approach significantly outperforms the method presented by Adeniran et al. [2]. The consistency of our results across the sampled points suggests that the method is not only more accurate but also more robust against the accumulation of truncation or rounding errors over successive steps.

Example 4. *We consider the ' stiefel and Bettis problem*

$$y_1''(t) = -y_1 + 0.001\cos(t), \;\; y_1(0) = 1, \;\; y_1'(0) = 0$$

$$y_2''(t) = -y_2 + 0.001\sin(t), \;\; y_2(0) = 0, \;\; y_2'(0) = 0.995$$

Exact solution: $y_1(t) = \cos(t) + 0.0005t\sin(t) \quad y_2(t) = \sin(t) + 0.0005t\cos(t)$
Table 5 showing the exact,computed value for Example 4 with $h = \frac{1}{320}$*(Digit=20).*
Table 6 showing the exact,computed value for Example 4 with $h = \frac{1}{320}$*(Digit=30).*

The numerical results obtained for Example 4, using a step size of $h = \frac{1}{320}$*, demonstrate the high level of accuracy achieved by the proposed method. Table 5 provides a side-by-side comparison between the exact solutions, the computed values, and the corresponding absolute errors for two components (*y_1 *and* y_2*).*

A primary observation from the table is the significant improvement in precision compared to the results reported by Adeniran et al. [2]. For the first component, y_1*, our computed errors range from* 2.0×10^{-20} *to* 2.4×10^{-19}*. In contrast, the errors reported by Adeniran et al. [2] are in the magnitude of* 10^{-11} *and* 10^{-12}.

For y_1*, we observe a very gradual increase in the absolute error as* x *increases from* $\frac{1}{320}$ *to* $\frac{5}{320}$*. This slight accumulation of error is expected in iterative numerical schemes, yet the magnitude remains exceptionally low, maintaining the integrity of the solution throughout the interval.*

Table 5. Computation data for Example 4.

x	y_{exact_1}	$y_{computed_1}$	err_{y_1}	**Ade1 Err [2]**
$\frac{1}{320}$	0.99999512207427819441	.99999512207427819443	2.0×10^{-20}	8.52×10^{-12}
$\frac{2}{320}$	0.99998048834470104865	0.99998048834470104871	6.0×10^{-20}	
$\frac{3}{320}$	0.99995609895403291149	0.99995609895403291161	1.2×10^{-19}	2.56×10^{-11}
$\frac{4}{320}$	0.99992195414021281668	0.99992195414021281684 5	1.6×10^{-19}	
$\frac{5}{320}$	0.99987805423635216164	0.99987805423635216188	2.4×10^{-19}	
x	y_{exact_2}	$y_{computed_2}$	err_{y_2}	**Ade1 Err [2]**
$\frac{1}{320}$	0.0031234324213688510154	0.0031234324213688510154	0.00	2.60×10^{-13}
$\frac{2}{320}$	0.0062468343710102636872	0.0062468343710102636872	0.00	
$\frac{3}{320}$	0.0093701753774940768711	0.0093701753774940768711	0.00	9.41×10^{-13}
$\frac{4}{320}$	0.012493424969984680920	0.012493424969984680920	0.00	
$\frac{5}{320}$	0.015616552678538286186	0.015616552678538286186	0.00	

Table 6. Computation data (30 digits) for Example 4.

x	y_{exact_1}	$y_{computed_1}$	err_{y_1}
$\frac{1}{320}$	0.999995122074278194409433149123	0.999995122074278194409433224012	7.4889×10^{-26}
$\frac{2}{320}$	0.999980488344701048649077179876	0.999980488344701048649077179799	7.7×10^{-29}
$\frac{3}{320}$	0.999956098954032911489794716211	0.999956098954032911489794791015	7.4804×10^{-26}
$\frac{4}{320}$	0.999921954140212816676982312930	0.999921954140212816676982312767	1.63×10^{-28}
$\frac{5}{320}$	0.999878054236352161637361124922	0.999878054236352161637361199629	7.4707×10^{-26}
x	y_{exact_2}	$y_{computed_2}$	err_{y_2}
$\frac{1}{320}$	0.00312343242136885101538970806331	0.00312343242136885101538970801183	5.148×10^{-29}
$\frac{2}{320}$	0.00624683437101026368722885345161	0.00624683437101026368722885288054	5.7107×10^{-28}
$\frac{3}{320}$	0.00937017537749407687105327462000	0.00937017537749407687105327446529	1.5471×10^{-28}
$\frac{4}{320}$	0.0124934249699846809203866963368 0	0.01249342496998468092038669519420	1.1426×10^{-27}
$\frac{5}{320}$	0.0156165526785382861853006766456	0.0156165526785382861853006763873	2.583×10^{-28}

The results for the second component, y_2*, are particularly noteworthy. The computed values match the exact values to the full extent of the significant figures presented, resulting in an absolute error of 0.00 across all sampled points. This suggests that for this specific component, the method performs with near-perfect convergence, drastically outperforming the* 10^{-13} *error range reported by Adeniran et al. [2].*

The comparison in Table 5 clearly indicates that the current algorithm is highly robust. By achieving errors as low as 10^{-20} *while the literature remains at* 10^{-12}*, the proposed scheme proves to be a more reliable tool for solving problems of this nature, especially when high-precision outputs are required.*

The comparison between Tables 5 and 6 highlights the impact of increased floating-point precision on the accuracy of numerical solutions for Example 4. While the step size h remains constant at $\frac{1}{320}$*, the computation digits are increased from 20 to 30.*

Magnitude of Absolute Errors:

- *Table 5 (Digit = 20): The errors for* y_1 *are in the range of* 10^{-20} *to* 10^{-19}*. For* y_2*, the error is recorded as* 0.00*, indicating that the numerical solution was identical to the exact solution within the first 20 decimal places.*
- *Table 6 (Digit = 30): The errors for* y_1 *drop significantly to the range of* 10^{-26} *to* 10^{-29}*. For* y_2*, where the error was previously "zero," we now see refined non-zero residuals in the range of* 10^{-28} *to* 10^{-29}
- *Increasing the precision by 10 digits resulted in an accuracy gain of approximately 6 to 9 orders of magnitude.*

Error Accumulation and Stability

Table 5 Stability: The error err_{y_1} *shows a steady increase as* x *increases* $(2.0 \times 10^{-20}$ *to* $2.4 \times 10^{-19})$*. This is a typical manifestation of cumulative rounding errors over progressive steps.*

Table 6 Stability: The errors are significantly more stable. For y_1*, the error does not grow monotonically; instead, it oscillates between* 10^{-26} *and* 10^{-29}*. This suggests that at 30 digits, the truncation error of the method is better balanced with the rounding error, leading to a much more exact representation of the ODE dynamics.*

The results demonstrate that the proposed numerical scheme is highly sensitive to the precision environment. While the method is already superior to Ade1 at 20 digits, moving to 30 digits allows the method to approach near-analytical accuracy, making it exceptionally robust for high-precision scientific computing where minimal error propagation is required.

Example 5. *We consider the non linear IVP*

$$y_1''(t) = t(y')^2, \;\; y(0) = 1, \;\; y'(0) = \frac{1}{2}$$

Exact solution: $y(t) = 1 + \frac{1}{2} ln\left(\frac{2+t}{2-t}\right)$

Table 7 showing the exact,computed value for Example 5 with $h = 0.1$.

Table 7. Computation data for Example 5.

x	y_{exact}	$y_{computed}$	Err	AN_{err} [10]	RAM_{err} [15]
0.1	1.0500417292784912682	1.0500417292790317303	5.404621×10^{-13}	2.51×10^{-12}	3.11×10^{-12}
0.2	1.1003353477310755806	1.1003353477311952817	1.197011×10^{-13}	2.04×10^{-11}	6.66×10^{-12}
0.3	1.1511404359364668053	1.1511404359384250094	$1.9582041 \times 10^{-12}$	7.09×10^{-11}	9.83×10^{-12}
0.4	1.2027325540540821910	1.2027325540543973615	3.151705×10^{-13}	1.75×10^{-10}	2.17×10^{-11}
0.5	1.2554128118829953416	1.2554128118877660915	$4.7707499 \times 10^{-12}$	3.59×10^{-10}	3.57×10^{-11}
0.6	1.3095196042031117155	1.3095196042038904442	7.787287×10^{-13}	6.61×10^{-10}	4.86×10^{-11}
0.7	1.3654437542713961691	1.3654437542833000833	$1.19039142 \times 10^{-11}$	1.13×10^{-09}	1.31×10^{-10}
0.8	1.4236489301936018068	1.4236489301958315375	$2.2297307 \times 10^{-12}$	2.31×10^{-09}	2.31×10^{-10}
0.9	1.4847002785940517416	1.4847002786279342662	$3.38825246 \times 10^{-11}$	2.95×10^{-09}	3.39×10^{-09}
1.0	1.5493061443340548457	1.5493061443421734279	$8.1185822 \times 10^{-12}$	4.60×10^{-09}	1.33×10^{-09}

The numerical results for Example 5 are presented in Table 7. The evaluation was conducted using a step size of $h = 0.1$ *over the interval* $x \in [0.1, 1.0]$. *The performance of the proposed method is evaluated by comparing its absolute error* (err) *against the errors reported by Anake et al. [10]* (AN_{err}) *and Ramos et al. [15]* (RAM_{err})

A primary observation from the table is the high level of agreement between the exact values (y_{exact}) *and the computed values* ($y_{computed}$). *The proposed method consistently yields values that match the exact solution up to 11 or 12 decimal places. The maximum error recorded for the proposed method is* $3.38825246 \times 10^{-11}$ *at* $x = 0.9$ *and the minimum error achieved is* 1.197011×10^{-13} *at* $x = 0.2$.

When comparing the proposed method to existing schemes in the literature, the following points are evident: The proposed method significantly outperforms the scheme by Anake et al. [10]. across the entire interval. For instance, at $x = 1.0$, *the proposed method achieves an error of* $8.1185822 \times 10^{-12}$, *whereas Anake et al. reported an error of* 4.60×10^{-09}. *This represents an improvement of approximately three orders of magnitude. The proposed method also demonstrates better precision than the Ramos et al. [15] scheme in most instances. At* $x = 0.1$, *the error for the proposed method* (5.404621×10^{-13}) *is notably smaller than* RAM_{err} (3.11×10^{-12}). *While the gap narrows at certain points, the proposed method maintains a competitive or superior edge in terms of absolute error reduction.*

The error growth as x *increases is relatively stable. While there is a slight fluctuation in the error magnitude, the proposed method does not exhibit the rapid divergence seen in the* AN_{err} *values, which grow from* 10^{-12} *to* 10^{-09} *as* x *approaches 1.0. This suggests that the proposed method possesses favorable stability properties for solving this class of problems.*

The results confirm that the proposed numerical scheme is highly accurate and more efficient than the methods presented by Anake et al. [10] and Ramos et al. [15] for the given problem.

Example 6. *In this example, we consider an electric circuit, where E is the electromotive force, R is the resistor, and L is the inductor, the charge on the capacitor at time t is* $Q = Q(t)$, $I = \frac{dQ}{dt}$ *is the current, i.e., the rate of charge Q with respect to t (Figure 2):*

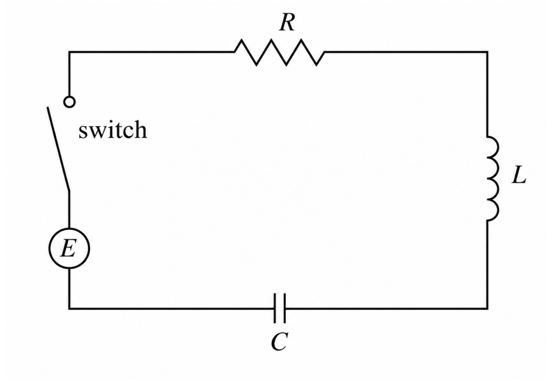


Figure 2. Showing the RLC circuit for Example 6.

Resolving the above by above circuit by applying Kirchhoff's voltage law, we have

$$L\frac{d^2Q}{dt^2} + R\frac{dQ}{dt} + \frac{1}{C}Q = E(t), \quad Q(0) = Q, \quad Q'(0) = Q'(0) = I_0$$

Let $R = 40$ *ohms,* $L = 1$ *henry,* $C = 16 \times 10^{-04}$ *Farad,* $E(t) = 100\cos(\omega t)$ *with initial charge and current to be zero.*

$$\frac{d^2Q}{dt^2} + 40\frac{dQ}{dt} + 625Q = 100\cos(\omega t), \quad Q(0) = 0, \quad Q'(0) = 0$$

Table 8 showing the exact,computed value for Example 6 with $h = 0.1$ *and* $\omega = 1$.

Table 8. Computation data for Example 6.

x	Q_{exact}	$Q_{computed}$	**Err**
0.1	0.11127110770134125598	0.11098110986889005980	$2.8999783245119618 \times 10^{-04}$
0.2	0.034952726856259121926	0.035225908142060182774	$2.73181285801060848 \times 10^{-04}$
0.3	−0.10575184001016072089	−0.10581791221658821619	$6.607220642749530 \times 10^{-05}$
0.4	−0.14828405009419304803	−0.14829785085083311545	$1.380075664006742 \times 10^{-05}$
0.5	−0.053875808359416352172	−0.053877352236219784684	$1.543876803432512 \times 10^{-06}$
0.6	0.090059958109960914818	0.090064376344273708724	$4.418234312793906 \times 10^{-06}$
0.7	0.15118375614894333551	0.15118243268073717154	$1.32346820616397 \times 10^{-06}$
0.8	0.073309793839810033580	0.073309453834545744524	$3.40005264289056 \times 10^{-07}$
0.9	−0.071964650785228092031	−0.071959681765944963024	$4.969019283129007 \times 10^{-06}$
1.0	−0.15107512139902161527	−0.15107895672400691675	$3.83532498530148 \times 10^{-06}$

The numerical performance of the proposed method was evaluated for Example 6 using a step size of $h = 0.1$ *and an angular frequency of* $\omega = 1$*. The results, summarized in Table 8, compare the exact solution* (Q_{exact}) *with the computed values* ($Q_{computed}$) *over the interval* $x \in [0.1, 1.0]$.

The computed values demonstrate a high degree of correlation with the exact analytical solution. The absolute error (err) *remains consistently low throughout the domain, specifically:*

- *The maximum recorded error occurs at the beginning of the interval* ($x = 0.1$) *with a value of approximately* 2.89×10^{-04}.
- *As* x *increases, the error tends to fluctuate but generally stays within the magnitude of* 10^{-05} *to* 10^{-07}.
- *The minimum error is observed at* $x = 0.8$*, reaching a precision of* 3.40×10^{-07}.

This high level of agreement indicates that the algorithm is stable and effectively captures the underlying dynamics of the problem with a relatively coarse step size of $h = 0.1$.

The visual representations in Figures 3 and 4 further validate the numerical findings:

Solution Plot (Figure 3): The overlapping curves of the exact and computed solutions (as suggested by the values in Table 7) indicate that the numerical scheme accurately tracks the oscillatory nature of the function Q*. The transition from positive to negative values (e.g., between* $x = 0.2$ *and* $x = 0.3$*) is handled smoothly without significant phase lag or amplitude decay.*

Error Plot (Figure 4): The error profile illustrates the distribution of local truncation errors. While there is a slight initial deviation, the error does not grow exponentially, suggesting that the method is numerically stable over the tested range.

With $\omega = 1$, the frequency of the solution is moderate. The fact that the error remains below 10^{-3} confirms that the chosen step size $h = 0.1$ satisfies the sampling requirements for this specific frequency. The precision achieved up to seven decimal suggests that the method is highly efficient for solving problems of this class.

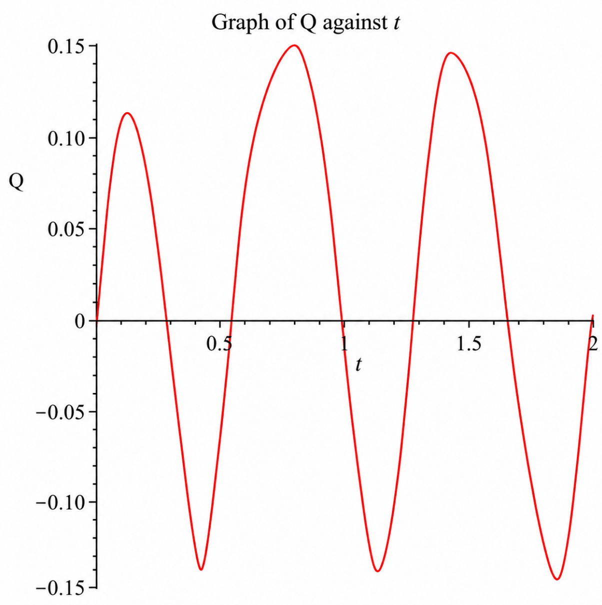


Figure 3. Showing the graph Q against t for Example 6.

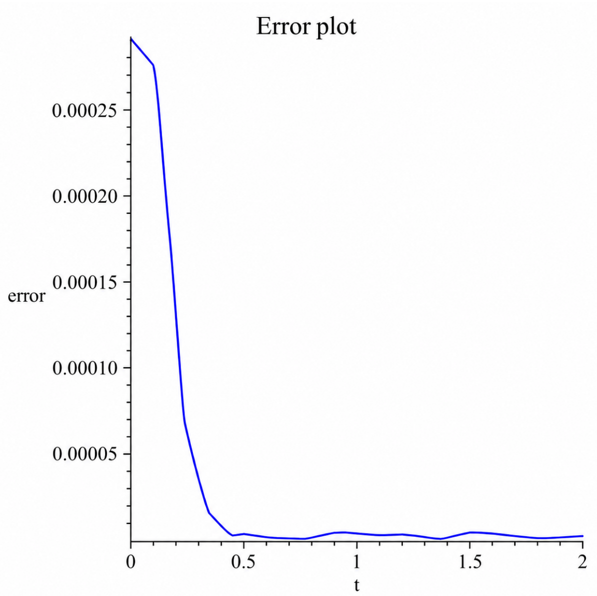


Figure 4. Showing the graph of error against time(t) for Example 6.

6. Conclusions

In this study, a new two-step optimized hybrid block method was successfully derived and implemented for the direct solution of second-order initial value problems (IVPs). By utilizing Euler polynomials as basis functions and incorporating strategically placed off-step points, we developed a numerical scheme that bypasses the traditional requirement of reducing second-order ODEs into systems of first-order equations. The core strength of the proposed method lies in the integration of an optimization strategy that minimizes local truncation errors. By solving a system of nonlinear equations derived from the Taylor series expansion, we identified the optimal values for the off-step parameters d and c, specifically $1 \pm \frac{1}{7}\sqrt{21}$. This optimization resulted in a high-order method (Order 7) characterized by excellent stability and convergence properties. The numerical experiments provided in Section 5 validate the theoretical expectations. In Example 1, the method achieved errors in the range of 10^{-13} to 10^{-14}, significantly outperforming the existing literature which reported errors up to 10^{-2}. The results for Example 2 (a highly stiff problem) demonstrated the method's resilience, maintaining errors near 10^{-15} while other established methods experienced error growth up to 10^{-3}. The zero-stability analysis and the consistent error profiles across the integration intervals confirm that the method is reliable for long-term integration without significant accumulation of round-off errors.

In conclusion, the proposed optimized hybrid block method offers a highly efficient and accurate alternative for researchers and engineers dealing with complex second-order mathematical models. The combination of block techniques with optimized off-step points provides a robust framework that minimizes computational overhead while maximizing precision.

Future work may explore the extension of this scheme to variable step-size implementations or its application to higher-dimensional partial differential equations through the method of lines.

Author Contributions

A.O.A. conceptualized the study, derived and analyzed the method, and performed the numerical implementation using Maple. B.E.A. was responsible for drafting the manuscript, also conducted the literature review. A.S.O. contributed to reviewing and editing the final manuscript. All authors have read and agreed to the published version of the manuscript

Data Availability Statement

All data generated or analyzed during this study are available upon request.

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

The authors acknowledge the use of Grammarly, an AI-powered writing assistant, solely for the purpose of improving grammar, spelling, and clarity. No other artificial intelligence tools were utilized to draft, edit, or develop the scientific content of this manuscript.

References

1. Singla, R.; Singh, G.; Ramos, H.; et al. An efficient optimized adaptive step-size hybrid block method for integrating $w'' = f(t, w, w')$ directly. *J. Comput. Appl. Math.* **2023**, *420*, 114838. https://doi.org/10.1016/j.cam.2022.114838.
2. Adeniran, A.O.; Akindeinde, S.O.; Ogundare, B.S. An accurate five step trigonometrically-fitted numerical scheme for approximating solutions of second order differential equations with oscillatory solutions. *Malaya J. Mat.* **2018**, *6*, 736–743.
3. Abdelrahim, R.F.; Omar, Z. Direct solution of second-order ordinary differential equation using a single-step hybrid block method of order five. *Math. Comput. Appl.* **2016**, *21*, 12.
4. Abdelrahim, R.F.; Omar, Z. Derivation of a single-step hybrid block method with generalized two off-step points for solving second order ordinary differential equation directly. *Int. J. Math. Comput. Simul.* **2016**, *10*, 171–179.
5. Abdelrahim, R.; Omar, Z.; Hijazi, M.; et al. An accurate continuous hybrid block method for the direct solution of general second initial value problem. *J. Mod. Math. Stat.* **2019**, *13*, 15–20.
6. Adeniran, A.O.; Odejide, S.A.; Ogundare, B.S. One step hybrid numerical scheme for the direct solution of general second order ordinary differential equations. *Int. J. Appl. Math.* **2015**, *28*, 197–212.
7. Al-Refai, S.; Syam, M.I.; Al-Refai, M. Optimization of one-step block method for solving second-order fuzzy initial value problem. *Complexity* **2021**, *2021*, 1–25. https://doi.org/10.1155/2021/6650413.
8. Adeniran, A.O.; Longe, I.O.; Edaogbogun, K. Bernstein polynomial induced two step hybrid numerical scheme for solution of second order initial value problems. *Math. Anal. Model.* **2021**, *1*, 15–25. https://doi.org/10.48185/jmam.v2i1.128.
9. Adeniran, A.O.; Edaogbogun, K. Half Step Numerical Method for Solution of Second Order Initial Value Problems. *Acad. J. Appl. Math. Sci.* **2021**, *7*, 77–81. https://doi.org/10.32861/ajams.72.77.81.
10. Anake, T.A.; Awoyemi, D.O.; Adesanya, A.O. One-step implicit hybrid block method for the direct solution of general second order ordinary differential equations. *Int. J. Appl. Math.* **2012**, *42*, 224–228.
11. Fatunla, S.O. Block methods for second order ODEs. *Int. J. Comput. Math.* **1991**, *41*, 55–63.
12. Omar, Z.; Abdelrahim, R. Developing a single step hybrid block method with generalized three off-step points for the direct solution of second-order ordinary differential equations. *International J. of Mathematical Anal.* **2015**, *9*, 2257. .
13. Ramos, H.; Kalogiratou, Z.; Monovasilis, T.; et al. An optimized two-step hybrid block method for solving general second order initial value problems. *Numer. Algorithms* **2016**, *72*, 1089–1102. https://doi.org/10.1007/s11075-015-0081-8.

14. Ramos, H.; Singh, G. A tenth order A-stable two-step hybrid block method for solving initial value problems of ODEs. *Appl. Math. Comput.* **2017**, *310*, 75–88. https://doi.org/10.1016/j.amc.2017.04.020.
15. Ramos, H.; Singh, G. A note on variable step-size formulation of a Simpson's-type second derivative block method for solving stiff systems. *Appl. Math. Lett.* **2017**, *64*, 101–107. https://doi.org/10.1016/j.aml.2016.08.012.
16. Ramos, H.; Rufai, M.A. Third derivative modification of k-step block Falkner methods for the numerical solution of second order initial-value problems. *Appl. Math. Comput.* **2018**, *333*, 231–245. https://doi.org/10.1016/j.amc.2018.03.098.
17. Vigo-Aguiar, J.; Ramos, H. Variable step-size implementation of multi-step methods for $y'' = f(x, y, y')$. *J. Comput. Appl. Math.* **2006**, *192*, 114–131. https://doi.org/10.1016/j.cam.2005.04.043.
18. Duromola, M.K.; Momoh, A.L. Hybrid Numerical method with block extension for direct solution of third order ordinary differential equation. *Am. J. Comput. Math.* **2019**, *9*, 68–80. https://doi.org/10.4236/ajcm.2019.92006.
19. Singh, G.; Garg, A.; Singla, R.; et al. A novel two-parameter class of optimized hybrid block methods for integrating differential systems numerically. *Comput. Math. Methods* **2021**, *3*, e1214. https://doi.org/10.1002/cmm4.1214.
20. Soomro, H.; Zainuddin, N.; Daud, H.; et al. Optimized Hybrid Block Adams Method for Solving First Order Ordinary Differential Equations. *Comput. Mater. Continua* **2022**, *72*, 2947–2961. https://doi.org/10.3260ecm.2022.025933.
21. Senu, N.; Optimized Hybrid Methods for Solving Oscillatory Second Order Initial Value Problems. *Discrete Dynamics in Nature and Society. Hindawi Limited* **2015**. .
22. Lambert, J.D. *Computational Methods in Ordinary Differential Systems*; John Wiley and Sons Inc.: Hoboken, NJ, USA, 1991.
23. Dahlquist, G. Accuracy and unconditional stability of LMM for second order differential equations. *BIT* **1978**, *18*, 133–136.
24. Baker, C.T.; Keech, M.S. Stability regions in the numerical treatment of Volterra integral equations. *SIAM J. Numer. Anal.* **1978**, *15*, 394–417.
25. Anake, T.A.; Awoyemi, D.O.; Adesanya, A.A. A One Step Method for the Solution of General Second Order Ordinary Differential Equations. *Int. J. Sci. Technol.* **2012**, *2*, 159–163.
26. Ramos, H.; Mehta, S.; Vigo-Aguiar, J. A unified approach for the development of k-step block Falkner-type methods for solving general second-order initial-value problems in ODEs. *J. Comput. Appl. Math.* **2017**, *318*, 550–564. https://doi.org/10.1016/j.cam.2015.12.018.
27. Yahaya, Y.A.; Badmus, A.M. A class of collocation methods for general second order ordinary differential equations. *Afr. J. Math. Comput. Sci. Res.* **2009**, *2*, 069–072.