Armenian Journal of Mathematics
Volume 18, Number 9, 2026, 1–29
https://doi.org/10.52737/18291163-2026.18.09-1-29

Adaptive Block Hybrid Method with Quasilinearization for the Accurate Solution of General Third-Order IVPs

S.D. Yakubu, P. Sibanda, S.P. Goqo and S.O. Akindeinde

Abstract. This paper introduces a novel variable-step block hybrid method for the direct solution of general third-order initial value problems. The proposed approach combines quasilinearization with an embedded procedure to enhance accuracy and computational efficiency. Rigorous analysis establishes the convergence and A-stability of the proposed method, providing a reliable numerical framework for a broad class of third-order initial value problems. Numerical experiments on a range of benchmark test problems demonstrate that the proposed method achieves high accuracy and competitive computational efficiency compared with several existing approaches. These results highlight the effectiveness and potential of the proposed method as a robust tool for the direct numerical solution of third-order differential equations arising in engineering and related applications.

Key Words: Block Hybrid Method, Quasilinearization, Embedded Procedure, Lagrange Interpolation
Mathematics Subject Classification 2020: 65L05, 65L06, 65L20

Introduction

Physical problems such as electromagnetic wave propagation, rocket motion, gravity-driven flows, and thin-film flows in physics and engineering are often modeled using third-order ordinary differential equations (ODEs) see, for example, [16]. Despite their importance, most of these problems lack closed-form analytical solutions, necessitating the use of numerical approximations. Consequently, recent studies have focused on developing direct numerical techniques for solving third-order initial value problems (IVPs).

In this study, we aim to develop an efficient numerical technique with strong stability properties capable of handling both stiff and non-stiff third-order IVPs, which often pose significant numerical challenges. The proposed method directly addresses the general third-order problem

$$y^{'''}(t) = f(t, y, y', y''),\ y(t_0) = y_0,\ y'(t_0) = \delta_0,\ y''(t_0) = \sigma_0,\ t \in [t_0, t_N], \quad (1)$$

where $f : \mathbb{R} \times \mathbb{R}^{3m} \to \mathbb{R}^m$ is continuous and satisfies a Lipschitz condition in its dependent variables.

For numerical methods based on linear multistep or block formulations, strong stability is commonly associated with the root condition. In particular, a method is said to be strongly stable if all the roots of its first characteristic polynomial lie within or on the unit circle, with every root on the unit circle being simple. This condition ensures that perturbations introduced during the numerical integration do not grow without bound and is fundamental for establishing the stability and convergence of the numerical method.

Third-order ODEs have therefore attracted considerable attention from researchers due to their wide range of applications and the challenges associated with their numerical solution. Numerous theoretical and numerical studies addressing this class of problems have been reported in the literature (see, for example, [3, 4, 9, 11, 15]). To improve the accuracy of existing methods for directly solving both special and general third-order IVPs, Lawal et al. [16] constructed a four-step block method based on a hybrid linear multistep approach for special third-order IVPs. The method was derived using collocation and interpolation techniques with power-series basis functions. Jator et al. [10] developed a new class of implicit continuous linear multistep methods and applied them as boundary value methods for the numerical integration of general third-order initial and boundary value problems. Similarly, Olagunju and Adeyefa [20] proposed a sixth-order single-step hybrid block method for the direct integration of first-, second-, and third-order IVPs. Appadu and Peer [6] constructed an optimized third-order weighted essentially non-oscillatory scheme for hyperbolic conservation laws, while Appadu [5] proposed an optimized low-dispersion and low-dissipation Runge–Kutta algorithm for computational aeroacoustics.

More recently, Rufai and Ramos [24] derived a fourth-derivative hybrid block strategy for the integration of third-order IVPs, while Osa and Olaoluwa [23] developed a fifth-fourth continuous block implicit hybrid method for solving third-order ODEs. An optimized implicit single-step fourth-derivative block hybrid method specifically designed for the direct solution of general third-order IVPs was introduced by Yakubu and Sibanda [27], incorporating advanced optimization techniques to enhance accuracy and computational efficiency. However, most existing methods do not efficiently address the stability and accuracy requirements associated with stiff versions of equation (1). The novelty of the proposed method lies in its adaptive implementation, which combines the quasilinearization technique with a block hybrid method to effectively handle stiff third-order ODEs, thereby achieving high accuracy and strong stability properties.

1 Derivation of the method

We develop a single-step high-order Block Hybrid Method (BHM) for the numerical solution of third-order IVPs of the form (1). The method uses Lagrange interpolation and collocation to achieve high-order accuracy and an efficient error estimator.

Let $\{t_{n+c_j}\}_{j=0}^{4}$ denote the collocation nodes in the interval $[t_n, t_{n+1}]$:

$$c_0 = 0, \quad c_1 = \frac{1}{4}, \quad c_2 = \frac{1}{2}, \quad c_3 = \frac{3}{4}, \quad c_4 = 1.$$

We construct the interpolating polynomial

$$Z(t) = \sum_{j=0}^{4} Y_j \, \ell_j(t),$$

where $\ell_j(t)$ are the Lagrange basis functions satisfying $\ell_j(t_{n+c_k}) = \delta_{jk}$. Its third derivative is approximated by

$$Z'''(t) = \sum_{j=0}^{4} Y_j \, \ell_j'''(t).$$

The interpolation conditions at t_n are

$$Z(t_n) = y_n, \quad Z'(t_n) = \delta_n, \quad Z''(t_n) = \sigma_n,$$

while collocation is imposed at the nodes t_{n+c_j}:

$$Z'''(t_{n+c_j}) = f(t_{n+c_j}, Z(t_{n+c_j}), Z'(t_{n+c_j}), Z''(t_{n+c_j})), \quad j = 0, \ldots, 4.$$

Solving the resulting system yields the BHM in block form:

$$\begin{aligned}
y_{n+c_i} &= y_n + hc_i \, \delta_n + \tfrac{1}{2} h^2 c_i^2 \, \sigma_n + h^3 \sum_{j=0}^{4} \epsilon_{ij} f_{n+c_j}, \\
\delta_{n+c_i} &= \delta_n + hc_i \, \sigma_n + h^2 \sum_{j=0}^{4} \chi_{ij} f_{n+c_j}, \\
\sigma_{n+c_i} &= \sigma_n + h \sum_{j=0}^{4} \upsilon_{ij} f_{n+c_j}, \qquad i = 0, \ldots, 4,
\end{aligned}$$

where the coefficients are given by

$$\epsilon_{ij} = \int_0^{c_i}\int_0^t\int_0^s \ell_j(u)\,du\,ds\,dt,\quad \chi_{ij} = \int_0^{c_i}\int_0^t \ell_j(s)\,ds\,dt,\quad \upsilon_{ij} = \int_0^{c_i} \ell_j(t)\,dt.$$

To enhance efficiency, the method can be implemented in adaptive mode using an embedded lower-order scheme constructed over a reduced node set $\{0, c_1, c_3, 1\}$, sharing the same function evaluations as the high-order method.

2 Analysis of the method

We present the theoretical analysis of the proposed third-order BHM. The method can be written in linear matrix form as

$$\Re_1 Y_{n+1} = \Re_0 Y_n + hD_0\Delta_n + h^2 L_0\Sigma_n + h^3(\Gamma_0 F_n + \Gamma_1 F_{n+1}), \tag{2}$$

where $\Re_0$, $\Re_1$, D_0, L_0, Γ_0, and Γ_1 are $m\times m$ matrices derived from the block hybrid formulation.

2.1 Local truncation error

Let $y(t)$ be sufficiently smooth. The linear differential operator associated with the BHM is

$$\mathcal{L}[y(t_n);h] = \sum_j \big[\xi_j y(t_n+jh) - h\upsilon_j y'(t_n+jh) - h^2\chi_j y''(t_n+jh) - h^3\epsilon_j y'''(t_n+jh)\big].$$

Expanding in a Taylor series about t_n gives the leading local truncation errors at the final node:

high-order:

$$y_{n+1} = \frac{y^8[t_n]h^8}{645120} + \mathcal{O}(h^9),$$

low-order:

$$y_{n+1} = \frac{y^7[t_n]h^7}{322560} + \mathcal{O}(h^8).$$

Hence, the high-order scheme achieves order $p = 5$ and the embedded low-order scheme achieves order $q = 4$.

2.2 Zero-stability

In the limit $h \to 0$, equation (2) reduces to

$$\Re_1 Y_{n+1} = \Re_0 Y_n,$$

and the corresponding characteristic polynomial is

$$g(\lambda) = \det[\lambda\Re_1 - \Re_0]. \tag{3}$$

For the proposed BHM, evaluation of (3) gives the characteristic roots

$$\lambda = 0, 0, 0, 1.$$

Thus, all roots lie within or on the unit circle, and the only root on the unit circle, $\lambda = 1$, is simple. Hence, the method satisfies the root condition and is zero-stable (strongly stable) [25]. Since the method is also consistent, it follows that the proposed BHM is convergent of order $p = 5$.

2.3 Linear stability

To investigate absolute stability, consider the scalar linear third-order test problem

$$y'''(t) = \mu y(t), \tag{4}$$

where $\mu \in \mathbb{C}$. Introducing the scaled stability variable

$$z = h\mu^3,$$

and applying the block hybrid scheme to (4) gives the amplification matrix

$$R(z) = \left(\Re_1 - z^3\Gamma_1\right)^{-1}\left(\Re_0 + zD_0 + z^2L_0 + z^3\Gamma_0\right).$$

The absolute stability region of the method is defined by

$$\mathcal{S} = \{z \in \mathbb{C} : \rho\big(R(z)\big) \leq 1\}$$

where $\rho(\cdot)$ denotes the spectral radius. A method is said to be A-stable if its absolute stability region contains the entire left half-plane, that is,

$$z \in \mathbb{C} : \mathrm{Re}(z) \leq 0 \subseteq \mathcal{S}.$$

This property is particularly important for stiff problems because it ensures that all modes associated with eigenvalues having non-positive real parts remain stable under the numerical discretization, irrespective of the step size.

For the proposed BHM, eigenvalue analysis of the amplification matrix $R(z)$ confirms that the entire left half-plane belongs to the absolute stability region. The corresponding stability region is illustrated in Figure 1. Therefore, the proposed BHM is A-stable (see Yakubu et al. [26]).

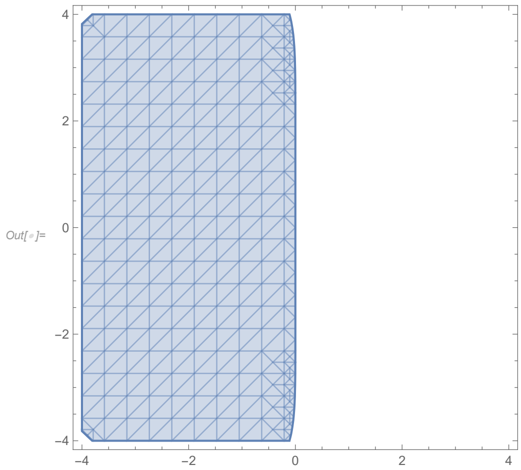


Figure 1: Stability region of the proposed BHM.

2.4 Estimator reliability and quasilinearization convergence

We examine the reliability of the embedded error estimator used in the proposed block hybrid method and the convergence of the quasilinearization (QLM) iteration for solving the resulting nonlinear block system. The embedded estimator is constructed from the converged stage evaluations of the right-hand side and exploits the reduced node set $0, c_1, c_3, 1$. Its consistency is established through moment conditions and a Taylor expansion, while the convergence of the QLM iteration is examined separately.

(A) Estimator analysis

Let the embedded approximation be denoted by $\widehat{y}_{n+1}$ and define the local error estimator by

$$\mathrm{EST}_y = y_{n+1} - \widehat{y}_{n+1}.$$

For the proposed third-order block hybrid formulation, the difference between the principal and embedded approximations can be written as

$$\mathrm{EST}_y = h^3 \sum_{j=0}^{m} w_j f_{n+c_j},$$

where the weights w_j are determined by the difference between the coefficients of the principal and embedded formulas. The corresponding stage values are evaluated at the converged block solution.

Lemma 1 (Moment conditions for the embedded estimator) *Let*

$$\mathrm{EST}_y = h^3 \sum_{j=0}^{m} w_j f_{n+c_j}$$

be the embedded error estimator associated with the proposed block hybrid method. If the estimator weights satisfy

$$\sum_{j=0}^{m} w_j c_j^k = 0, \qquad k = 0, 1, \ldots, s-1,$$

and

$$\sum_{j=0}^{m} w_j c_j^s \neq 0,$$

then the first non-vanishing term in the Taylor expansion of EST_y *occurs at order* h^{s+3}.

Proof. Expanding the stage evaluations about t_n gives

$$f_{n+c_j} = \sum_{k=0}^{\infty} \frac{(c_j h)^k}{k!} f^{(k)}(t_n).$$

Substitution into the estimator gives

$$\mathrm{EST}_y = h^3 \sum_{j=0}^{m} w_j \sum_{k=0}^{\infty} \frac{(c_j h)^k}{k!} f^{(k)}(t_n).$$

Interchanging the order of summation yields

$$\mathrm{EST}_y = h^3 \sum_{k=0}^{\infty} \frac{h^k}{k!} \left(\sum_{j=0}^{m} w_j c_j^k \right) f^{(k)}(t_n).$$

The assumed moment conditions imply

$$\sum_{j=0}^{m} w_j c_j^k = 0, \qquad k = 0, 1, \ldots, s-1.$$

Hence, all terms of degree less than s vanish and the first non-zero contribution is

$$\mathrm{EST}_y = \frac{h^{s+3}}{s!} \left(\sum_{j=0}^{m} w_j c_j^s \right) f^{(s)}(t_n) + \mathcal{O}(h^{s+4}).$$

Therefore,

$$\mathrm{EST}_y = \mathcal{O}(h^{s+3}),$$

which proves the result. □

Theorem 1 (Reliability of the embedded estimator) *Assume that the right-hand side is sufficiently differentiable and that the principal and embedded block hybrid schemes have orders $p = 5$ and $q = 4$, respectively. Then the local truncation errors at the endpoint satisfy*

$$\mathrm{LTE}_p = \frac{y^{(8)}(t_n)}{645120} h^8 + \mathcal{O}(h^9)$$

and

$$\mathrm{LTE}_q = \frac{y^{(7)}(t_n)}{322560} h^7 + \mathcal{O}(h^8).$$

Consequently, the difference between the principal and embedded approximations provides an asymptotically consistent local error estimator of the form

$$\mathrm{EST}_y = Ch^7 + \mathcal{O}(h^8),$$

where C is a bounded method-dependent coefficient. In particular,

$$|\mathrm{EST}_y| = \mathcal{O}(h^7).$$

Proof. The stated local truncation-error expansions follow from the Taylor expansion of the principal and embedded schemes. Since the principal scheme is of order 5 and the embedded scheme is of order 4, their leading local truncation errors are of orders h^8 and h^7, respectively. Therefore,

$$y_{n+1} - y^*_{n+1} = \frac{y^{(7)}(t_n)}{322560} h^7 + \mathcal{O}(h^8),$$

because the h^7 contribution from the embedded scheme is not cancelled by the higher-order principal scheme. Hence,

$$\mathrm{EST}_y = Ch^7 + \mathcal{O}(h^8),$$

with

$$C = \frac{y^{(7)}(t_n)}{322560}$$

for the leading local truncation-error contribution. Thus,

$$|\mathrm{EST}_y| = \mathcal{O}(h^7),$$

which establishes the asymptotic consistency of the embedded estimator. □

Remark 1 *The result shows that the embedded estimator has the same leading order as the local truncation error of the fourth-order embedded approximation. Hence, the estimator can be used to monitor the local discretization error and to guide adaptive step-size selection. The use of an embedded approximation for local error estimation and adaptive step-size control is standard in adaptive numerical integration; see, for example, Lambert [14] and Butcher [8].*

(B) QLM local convergence

Let the block collocation residual be

$$\mathcal{F}_h(\mathbf{U}) = 0,$$

where $\mathbf{U}$ contains the stage unknowns over one step of size h. The implemented QLM corresponds to Newton's method applied to $\mathcal{F}_h$.

Theorem 2 (Newton–Kantorovich convergence for the block residual) *Assume that $\mathcal{F}_h$ is continuously differentiable in a neighborhood of the initial iterate $\mathbf{U}^{(0)}$, that $D\mathcal{F}_h(\mathbf{U}^{(0)})$ is nonsingular, and that*

$$\|D\mathcal{F}_h(\mathbf{U}) - D\mathcal{F}_h(\mathbf{V})\| \leq L\|\mathbf{U} - \mathbf{V}\|$$

for some $L > 0$. Define

$$B = \left\|D\mathcal{F}_h(\mathbf{U}^{(0)})^{-1}\right\|, \qquad \eta = \left\|D\mathcal{F}_h(\mathbf{U}^{(0)})^{-1}\mathcal{F}_h(\mathbf{U}^{(0)})\right\|.$$

If

$$\gamma = BL\eta \leq \frac{1}{2},$$

then the Newton iterates

$$\mathbf{U}^{(k+1)} = \mathbf{U}^{(k)} - D\mathcal{F}_h(\mathbf{U}^{(k)})^{-1}\mathcal{F}_h(\mathbf{U}^{(k)})$$

are well-defined and converge to a unique root $\mathbf{U}^*$ in the corresponding Kantorovich neighborhood. Moreover, once the iterates enter the local Newton region, the convergence is quadratic.

Proof. The result follows directly from the Newton–Kantorovich theorem under the stated differentiability, nonsingularity, Lipschitz, and Kantorovich conditions. Hence, the QLM iteration converges locally to the solution of the nonlinear block system, with quadratic convergence in the local Newton region. □

Remark 2 *The Newton–Kantorovich conditions provide sufficient conditions for local convergence of the implemented QLM iteration. The theoretical result is consistent with the classical Newton–Kantorovich framework; see Kantorovich [12, 13].*

3 Implementation of the method

The proposed BHM is implemented in both constant and adaptive step-size modes. The nonlinear block collocation equations are solved iteratively using QLM. For the adaptive implementation, an embedded lower-order formula is used to construct an error estimate from the same function evaluations as the principal scheme. The resulting error indicator is then used to accept or reject a step and to adjust the step size. The constant step-size implementation uses the same QLM-based BHM without error-based step-size adaptation.

3.1 Embedded lower-order scheme and error estimation

The principal BHM provides the higher-order approximation y_{n+1} at the endpoint

$$t_{n+1} = t_n + h.$$

To estimate the local discretization error, an embedded lower-order formula is constructed using the same stage function evaluations required by the principal scheme. The embedded approximation at the endpoint is given by

$$y^*_{n+1} = y_n + h\delta_n + \frac{h^2}{2}\sigma_n + h^3\left(\frac{13}{360}f_n + \frac{1}{9}f_{n+c_1} + \frac{1}{45}f_{n+c_3} - \frac{1}{360}f_{n+1}\right), \tag{5}$$

where

$$f_n = f(t_n, y_n, \delta_n, \sigma_n),$$

and

$$f_{n+c_j} = f(t_{n+c_j}, y_{n+c_j}, \delta_{n+c_j}, \sigma_{n+c_j}), \qquad c_1 = \frac{1}{4}, \quad c_2 = \frac{1}{2}, \quad c_3 = \frac{3}{4}.$$

The notation y^*_{n+1} denotes the embedded lower-order approximation, whereas y_{n+1} denotes the approximation obtained from the principal BHM.

The embedded error estimate is defined as the difference between the principal and embedded endpoint approximations,

$$\mathrm{EST} = y_{n+1} - y^*_{n+1}. \tag{6}$$

On substituting the endpoint formula of the principal BHM and the embedded formula (5) into (6), the common terms cancel, and the resulting error estimator can be written in the compact form

$$\mathrm{EST} = \frac{h^3}{1260}\left(f_n - 4f_{n+c_1} + 6f_{n+c_2} - 4f_{n+c_3} + f_{n+1}\right). \tag{7}$$

For a system of third-order IVPs, the vector error estimate in (7) is converted into a scalar error indicator using

$$\text{err\_ratio} = \frac{|\text{EST}|}{\max\{1, |y_{n+1}|\}}, \tag{8}$$

where $|\cdot|$ denotes the Euclidean norm. In the adaptive implementation, a step is accepted when

$$\text{err\_ratio} < \text{TOL},$$

and rejected otherwise. The prescribed tolerance is denoted as TOL.

3.2 Quasilinearization iteration

The nonlinear block collocation equations are solved by quasilinearization. Let

$$\left(Y_{n+c_j}^{(r)}, D_{n+c_j}^{(r)}, S_{n+c_j}^{(r)}\right)$$

denote the stage approximations at the rth QLM iteration. The nonlinear function is linearized about the current iterate according to

$$\begin{aligned} f(t_{n+c_j}, Y_{n+c_j}, D_{n+c_j}, S_{n+c_j}) &\approx f_{n+c_j}^{(r)} + L_{1,n+c_j}^{(r)}\left(Y_{n+c_j} - Y_{n+c_j}^{(r)}\right) \\ &+ L_{2,n+c_j}^{(r)}\left(D_{n+c_j} - D_{n+c_j}^{(r)}\right) + L_{3,n+c_j}^{(r)}\left(S_{n+c_j} - S_{n+c_j}^{(r)}\right), \end{aligned}$$

where

$$f_{n+c_j}^{(r)} = f(t_{n+c_j}, Y_{n+c_j}^{(r)}, D_{n+c_j}^{(r)}, S_{n+c_j}^{(r)}),$$

and

$$L_1 = \frac{\partial f}{\partial y}, \qquad L_2 = \frac{\partial f}{\partial y'}, \qquad L_3 = \frac{\partial f}{\partial y''}.$$

Using the collocation coefficients derived for the proposed BHM, the linearized equations for the stage variables can be represented compactly as

$$\begin{aligned} Y_{n+c_i}^{(r+1)} &= y_n + hc_i\delta_n + \frac{h^2c_i^2}{2}\sigma_n + h^3\sum_{j=0}^{4}\epsilon_{ij}\widehat{f}_{n+c_j}^{(r)}, \\ D_{n+c_i}^{(r+1)} &= \delta_n + hc_i\sigma_n + h^2\sum_{j=0}^{4}\chi_{ij}\widehat{f}_{n+c_j}^{(r)}, \\ S_{n+c_i}^{(r+1)} &= \sigma_n + h\sum_{j=0}^{4}\upsilon_{ij}\widehat{f}_{n+c_j}^{(r)}, \qquad i = 1, \dots, 4, \end{aligned}$$

where $\widehat{f}_{n+c_j}^{(r)}$ denotes the quasilinearized approximation of the nonlinear function at the rth iteration. The coefficients ϵ_{ij}, χ_{ij}, and υ_{ij} are the collocation coefficients obtained in the derivation of the proposed method.

For a vector-valued problem, the Jacobian quantities L_1, L_2, and L_3 are the corresponding square matrices of partial derivatives with respect to y, y', and y'', respectively. The resulting coupled linear block system is assembled and solved simultaneously for all stage components.

The QLM iteration is terminated when the maximum change between two successive stage iterates satisfies

$$\max\left\{|Y^{(r+1)} - Y^{(r)}|_\infty, |D^{(r+1)} - D^{(r)}|_\infty, |S^{(r+1)} - S^{(r)}|_\infty\right\} < \mathrm{tol}_{\mathrm{nl}}, \qquad (9)$$

where $\mathrm{tol}_{\mathrm{nl}}$ is the prescribed nonlinear iteration tolerance. A maximum number of QLM iterations is also imposed. If convergence is not achieved within the prescribed maximum number of iterations, the adaptive implementation reduces the current step size and repeats the computation on the same interval.

3.3 Adaptive step-size procedure

After convergence of the QLM iteration, the endpoint values

$$(y_{n+1}, \delta_{n+1}, \sigma_{n+1})$$

are obtained from the final collocation stage. The implementation then evaluates the error estimator (7), followed by the normalized error indicator (8).

Since the principal BHM has algebraic order $p = 5$, the new step size is determined by

$$h_{\mathrm{new}} = \min\left\{h_{\max},\ \zeta h\left(\frac{\mathrm{TOL}}{\max(\mathrm{err\_ratio}, \varepsilon)}\right)^{1/(p+1)}\right\}, \qquad (10)$$

where $h_{\min}$ and $h_{\max}$ are the prescribed lower and upper step-size bounds, $\zeta \in (0, 1)$ is a safety factor, and ε is a small positive number used to avoid division by zero.

The step size is subsequently restricted by $h_{\min}$ and $h_{\max}$.

If

$$\mathrm{err\_ratio} < \mathrm{TOL},$$

the step is accepted, the numerical solution is advanced to t_{n+1}, and a new step size is selected. If

$$\mathrm{err\_ratio} \geq \mathrm{TOL},$$

the step is rejected, the step size is reduced, and the same interval is recomputed.

The final step is adjusted when necessary so that the integration terminates exactly at t_f.

3.4 Pseudo-code for the adaptive embedded procedure

The adaptive implementation of the proposed BHM with QLM is summarized as follows.

Step 1. Initialization.

Specify the third-order IVP (1), the initial values $(t_0, y_0, \delta_0, \sigma_0)$, the final integration point t_f, the initial step size h, the tolerance TOL, the step-size bounds $h_{\min}$ and $h_{\max}$, the nonlinear tolerance $\mathrm{tol}_{\mathrm{nl}}$, and the maximum number of QLM iterations.

Step 2. Stage construction.

For the current step, construct

$$t_{n+c_j} = t_n + hc_j, \qquad c_j \in \left\{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\right\}.$$

Initialize the stage predictors using the current endpoint values.

Step 3. QLM iteration.

Evaluate f and the Jacobian matrices L_1, L_2, L_3 at the current stage approximations. Assemble and solve the linearized block system to obtain updated stage values. Repeat until the convergence criterion (9) is satisfied or the maximum number of iterations is reached.

Step 4. Embedded error estimation.

After QLM convergence, obtain the principal endpoint approximation y_{n+1}. Construct the embedded approximation y^*_{n+1} according to (5). The implementation evaluates the equivalent difference directly through (7). Compute the normalized error indicator using (8).

Step 5. Step acceptance and update.

If err_ratio $<$ TOL, accept the step, advance the solution, and update h according to (10). Otherwise, reject the step, reduce h, and repeat the computation on the same interval.

Step 6. Termination.

Repeat Steps 2–5 until $t = t_f$, with the final step shortened when necessary to reach t_f exactly.

The complete adaptive procedure is summarized in Algorithm 1.

Algorithm 1 Adaptive Block Hybrid Method with QLM

Input: $t_0, y_0, \delta_0, \sigma_0, t_f, \mathrm{TOL}, h, h_{\min}, h_{\max}, \mathrm{tol}_{\mathrm{nl}}$, and maximum QLM iterations.
Initialize: $t \leftarrow t_0$, $y \leftarrow y_0$, $\delta \leftarrow \delta_0$, $\sigma \leftarrow \sigma_0$.
while $t < t_f$ **do**
    if $t + h > t_f$ **then**
        $h \leftarrow t_f - t$.
    end if
    if $h < h_{\min}$ **then**
        Stop because the step size has fallen below $h_{\min}$.
    end if
    Set $h \leftarrow \min(h, h_{\max})$.
    Construct the stage points $t_{n+c_j} = t + hc_j$.
    Initialize the stage predictors $(Y^{(0)}, D^{(0)}, S^{(0)})$.
    Evaluate $f_n = f(t, y, \delta, \sigma)$.
    Set $r \leftarrow 0$.
    repeat
        Evaluate f, L_1, L_2, and L_3 at the current stage iterates.
        Assemble the linearized block system using the collocation coefficient matrices $(\epsilon, \chi, \upsilon)$.
        Solve the block system for $(Y^{(r+1)}, D^{(r+1)}, S^{(r+1)})$.
        $r \leftarrow r + 1$.
    until the QLM convergence criterion is satisfied or r reaches the maximum number of iterations
    if QLM did not converge **then**
        Reduce h and retry the current step.
        continue.
    end if
    Evaluate the embedded error estimator EST from (7).
    Compute err_ratio using (8).
    if err_ratio $<$ TOL **then**
        Accept the step and advance the solution.
        Compute h using (10).
    else
        Reject the step and reduce h.
    end if
end while
Return the numerical solution.

3.5 Constant step-size implementation

The proposed BHM is also implemented with a constant step size. The prescribed step size h is used throughout the integration interval, except for the final step, which is shortened when necessary to terminate exactly at t_f. At each step, the collocation points are defined by

$$t_{n+c_j} = t_n + hc_j, \qquad c_j \in \left\{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\right\}.$$

The stage values are initialized from the current endpoint solution, and the nonlinear collocation equations are solved using QLM. After convergence, the endpoint values

$$(y_{n+1}, \delta_{n+1}, \sigma_{n+1})$$

are accepted as the numerical solution at the new step.

The constant step-size implementation proceeds as follows:

Step 1. Initialize $t_0, y_0, \delta_0, \sigma_0$, the prescribed step size h, the final point t_f, the nonlinear tolerance $\mathrm{tol}_{\mathrm{nl}}$, and the maximum number of QLM iterations.

Step 2. At the current point t_n, shorten the step if necessary so that the next step terminates at t_f.

Step 3. Construct the stage points

$$t_{n+c_j} = t_n + hc_j.$$

Initialize the stage predictors using the current endpoint values.

Step 4. Evaluate f and the Jacobian matrices L_1, L_2, L_3 at the current stage approximations and perform QLM iterations until the prescribed nonlinear convergence criterion is satisfied or the maximum number of iterations is reached.

Step 5. Accept the endpoint values

$$(y_{n+1}, \delta_{n+1}, \sigma_{n+1})$$

obtained from the final collocation stage as the updated numerical solution.

Step 6. Advance the independent variable to

$$t_{n+1} = t_n + h.$$

Step 7. Repeat Steps 2–6 until $t = t_f$.

The constant step-size implementation is summarized in Algorithm 2.

Algorithm 2 Block Hybrid Method with QLM (Constant Step Size)

Input: $t_0, y_0, \delta_0, \sigma_0, h, t_f, \mathrm{tol}_{\mathrm{nl}}$, and maximum QLM iterations.
Initialize: $t \leftarrow t_0$, $y \leftarrow y_0$, $\delta \leftarrow \delta_0$, $\sigma \leftarrow \sigma_0$.
while $t < t_f$ **do**
  if $t + h > t_f$ **then**
    $h \leftarrow t_f - t$.
  end if
  Construct the stage points $t_{n+c_j} = t + hc_j$.
  Initialize $(Y^{(0)}, D^{(0)}, S^{(0)})$ from (y, δ, σ).
  Evaluate $f_n = f(t, y, \delta, \sigma)$.
  Set $r \leftarrow 0$.
  repeat
    Evaluate f, L_1, L_2, and L_3 at the current stage approximations.
    Assemble the linearized block system using $(\epsilon, \chi, \upsilon)$.
    Solve for $(Y^{(r+1)}, D^{(r+1)}, S^{(r+1)})$.
    $r \leftarrow r + 1$.
  until the QLM convergence criterion is satisfied or r reaches the maximum number of iterations
  if QLM did not converge **then**
    Report failure of QLM convergence.
    Stop.
  end if
  Accept

$$(y, \delta, \sigma) \leftarrow (Y_4^{(r)}, D_4^{(r)}, S_4^{(r)}).$$

  Advance $t \leftarrow t + h$.
end while
Return the numerical solution.

4 Numerical simulations

In this section, we present numerical experiments to evaluate the performance of the proposed block hybrid method with quasilinearization (BHM-QLM) on several test problems of the form (1). The method is implemented in two variants: a constant step-size version (CBHM-QLM) and a variable step-size version (VBHM-QLM).

The accuracy of the method is measured using the maximum absolute error (MAE):

$$\mathrm{MAE} = \max_{0 \le j \le N} \|y(t_j) - y_j\|,$$

where $y(t_j)$ is the exact solution and y_j is the numerical approximation.

The rate of convergence (ROC) between two successive runs with N and

$2N$ steps is computed as

$$\mathrm{ROC} = \log_2\left(\frac{E(N)}{E(2N)}\right),$$

where $E(N)$ is the error obtained using N steps.

The following abbreviations are used in the subsequent tables and figures: NS–number of accepted steps; NFEVAL–number of function evaluations; NJAC–number of Jacobian evaluations; NITERS–number of nonlinear iterations; CPU Time–elapsed runtime; Tol–tolerance; MaxErr–maximum absolute error. For comparison, we include results from several existing methods: THMD [11], BVM_5 [10], ADHBM [1], ZOK [22], ZOA [21], MRKM [18], AWYM [7], ZARH [17], SMLMM [19], and ADGE [2].

The numerical results are reported in Tables 1–7 and Figures 2–7.

Problem 1
Nonlinear IVP

$$y''' + 2e^{-3y} - 4(1+t)^{-3} = 0,\ y(0) = 0,\ y'(0) = 1,\ y''(0) = -1,\ t \in [0, 1],$$

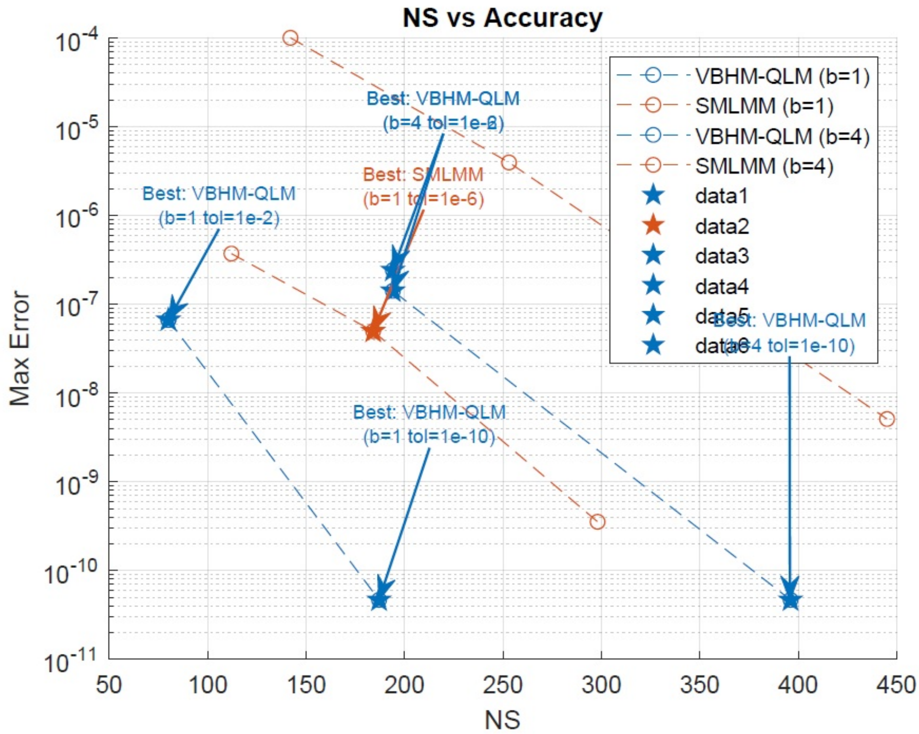
with exact solution

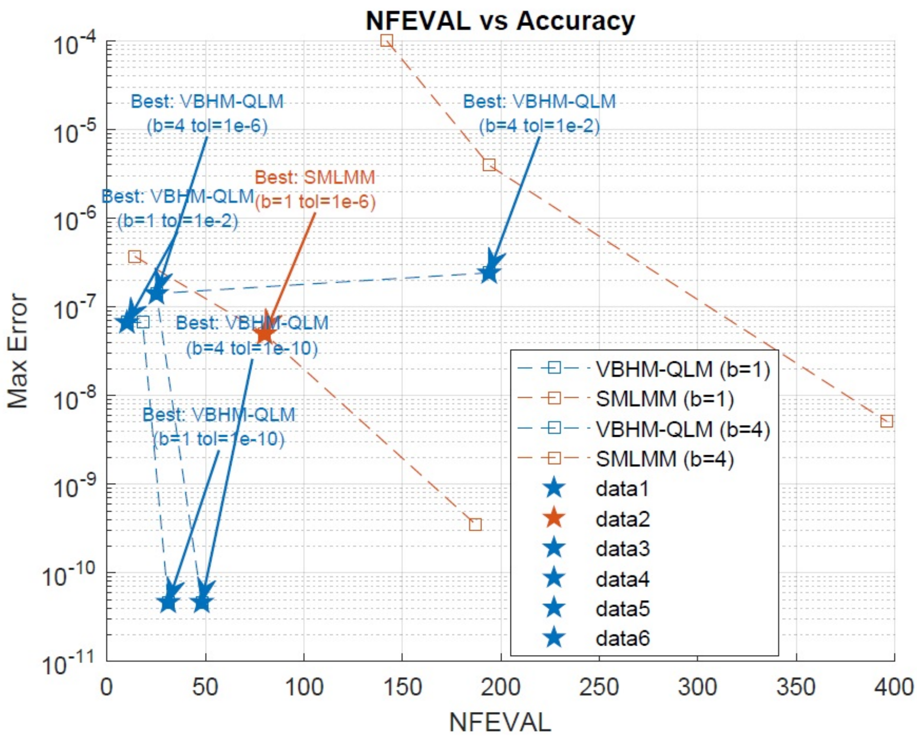
$$y(t) = \ln(1+t)$$

from [19].

Table 1: Performance comparison for Problem 1.

b	h	Tol	Method	NS	NFEVAL	NJAC	NITERS	Time_s	MaxErr_y
1	10^{-1}	10^{-2}	VBHM-QLM	4	80	60	15	0.029377	$6.6470e^{-8}$
			SMLMM	10	112	–	–	–	$3.7100e^{-7}$
			CBHM-QLM	11	179	124	31	0.030313	$1.5950e^{-10}$
4	10^{-1}	10^{-2}	VBHM-QLM	10	194	144	36	0.032138	$2.4170e^{-7}$
			SMLMM	14	142	–	–	–	$9.990e^{-5}$
			CBHM-QLM	40	680	480	120	0.043065	$1.5950e^{-10}$
1	10^{-1}	10^{-6}	VBHM-QLM	4	80	60	15	0.025065	$6.6470e^{-8}$
			SMLMM	18	184	–	–	–	$4.9600e^{-8}$
			CBHM-QLM	100	1700	1200	300	0.047854	$2.1560e^{-17}$
4	10^{-1}	10^{-6}	VBHM-QLM	10	194	144	36	0.030563	$1.4110e^{-7}$
			SMLMM	25	253	–	–	–	$3.9400e^{-6}$
			CBHM-QLM	401	6809	4804	1201	0.111207	$2.1560e^{-17}$
1	10^{-1}	10^{-10}	VBHM-QLM	10	187	132	33	0.031916	$3.5400e^{-11}$
			SMLMM	31	298	–	–	–	$3.5400e^{-10}$
4	10^{-1}	10^{-10}	VBHM-QLM	21	374	264	66	0.037752	$4.6730e^{-11}$
			SMLMM	48	445	–	–	–	$5.0900e^{-9}$


NS vs Accuracy
Max Error
NS
VBHM-QLM (b=1)
SMLMM (b=1)
VBHM-QLM (b=4)
SMLMM (b=4)
data1
data2
data3
data4
data5
data6
Best: VBHM-QLM (b=1 tol=1e-2)
Best: SMLMM (b=1 tol=1e-6)
Best: VBHM-QLM (b=1 tol=1e-10)
Best: VBHM-QLM (b=4 tol=1e-10)


NFEVAL vs Accuracy
Max Error
NFEVAL
VBHM-QLM (b=1)
SMLMM (b=1)
VBHM-QLM (b=4)
SMLMM (b=4)
data1
data2
data3
data4
data5
data6
Best: VBHM-QLM (b=4 tol=1e-6)
Best: VBHM-QLM (b=4 tol=1e-2)
Best: SMLMM (b=1 tol=1e-6)
Best: VBHM-QLM (b=1 tol=1e-2)
Best: VBHM-QLM (b=4 tol=1e-10)
Best: VBHM-QLM (b=1 tol=1e-10)

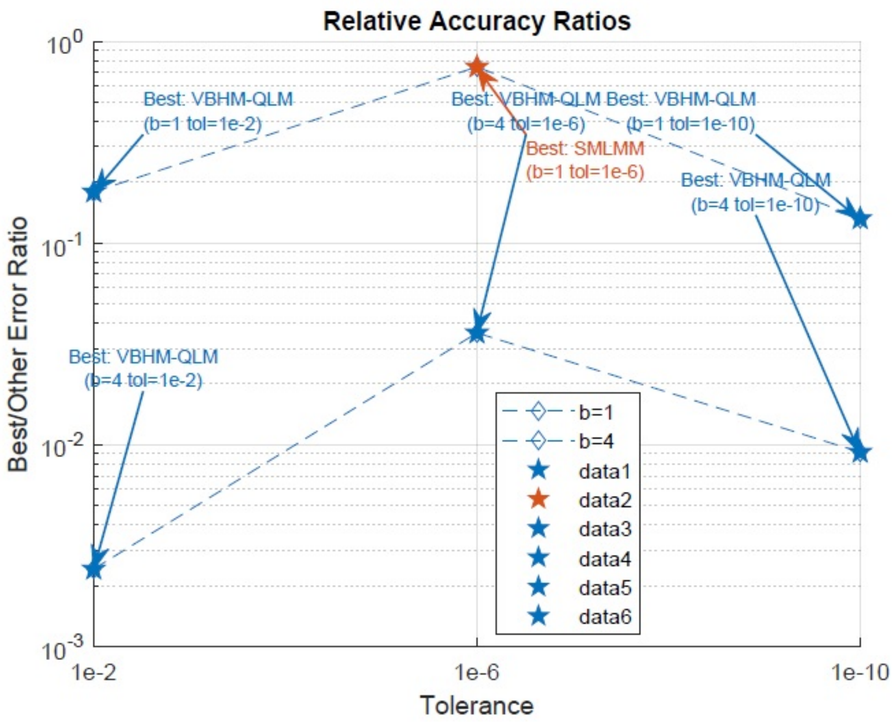


Figure 2: Accuracy comparison for Problem 1 at different tolerance levels.

Table 1 and Figure 2 present a comparative performance analysis of VBHM-QLM, CBHM-QLM, and SMLMM for different step sizes h and tolerance levels. The results show that VBHM-QLM consistently achieves higher accuracy with fewer steps than SMLMM, which is a variable-step block multistep method of order $p = 7$. Both methods were implemented in adaptive modes.

Problem 2
Linear IVP

$$y''' + 4y' - t = 0, \quad y(0) = 0, \quad y'(0) = 0, \quad y''(0) = 1, \quad t \in [0, 15],$$

with exact solution

$$y(t) = \frac{3}{10}(1 - \cos 2t) - \frac{1}{8}t^2$$

from [1].

Table 2: Performance comparison for Problem 2.

b	h	Method	NS	NFEVAL	NJAC	NITERS	Time_s	MaxErr_y
5	0.025	CBHM-QLM	200	2600	1600	400	0.058622	$1.1620e^{-12}$
		AWYM	200	–	–	–	–	$3.9400e^{-6}$
		ZARH	56	–	–	–	–	$9.1400e^{-8}$
		ADHBM	56	–	–	–	–	$3.6900e^{-11}$
10	0.025	CBHM-QLM	400	5200	3200	800	0.087373	$1.1620e^{-12}$
		AWYM	400	–	–	–	–	$3.8000e^{-6}$
		ZARH	136	–	–	–	–	$1.5200e^{-10}$
		ADHBM	136	–	–	–	–	$4.5300e^{-11}$
15	0.025	CBHM-QLM	600	7800	4800	1200	0.112579	$1.1620e^{-12}$
		AWYM	600	–	–	–	–	$2.2900e^{-6}$
		ZARH	180	–	–	–	–	$1.5400e^{-10}$
		ADHBM	180	–	–	–	–	$1.5200e^{-10}$

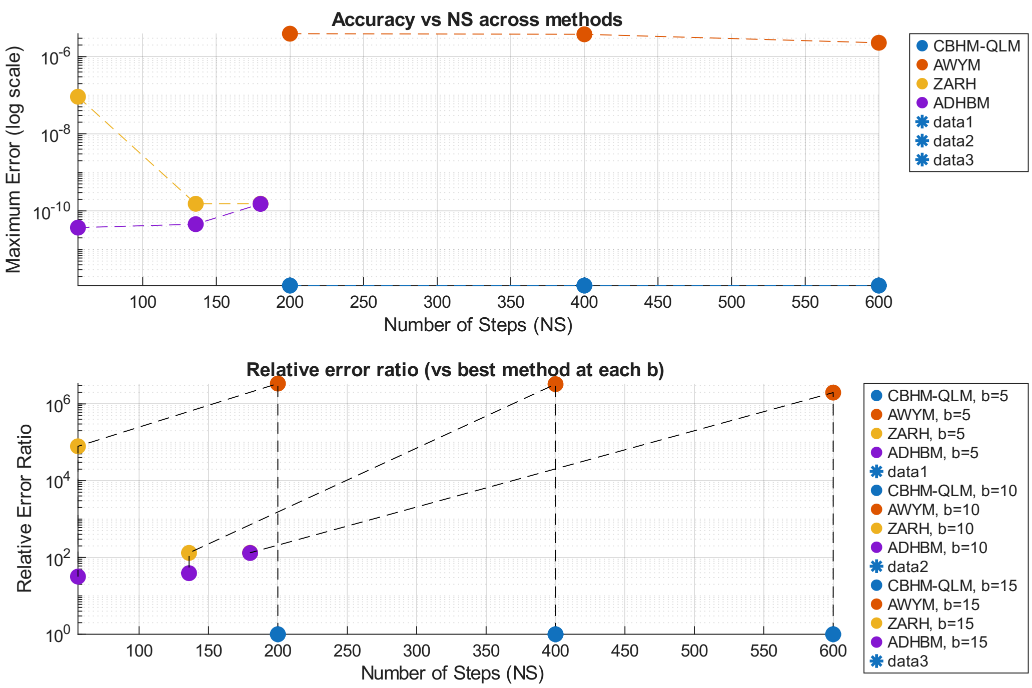


Figure 3: Accuracy comparison for Problem 2 at different values of b.

Table 2 and Figure 3 illustrate the performance of CBHM-QLM compared with ADHBM, AWYM, and ZARH, which are one-step block hybrid method, p-stable linear multistep method and four-step with a two off-step points method of order $p = 5, 6$, and 7, respectively, for different values of h. The results demonstrate that CBHM-QLM achieves superior accuracy across all tested cases. This indicates that CBHM-QLM is highly competitive, even against higher-order block methods.

Problem 3
Linear IVP from [1]:

$$y''' + y = 0, \quad y(0) = 1, \quad y'(0) = -1, \quad y''(0) = 1, \quad t \in [0, 1],$$

with exact solution

$$y(t) = e^{-t}.$$

Table 3: Performance comparison for Problem 3.

h	Tol	Method	NS	NFEVAL	NJAC	NITERS	Time_s	MaxErr_y
10^{-1}	10^{-12}	VBHM-QLM	9	117	72	18	0.030156	$5.2770e^{-13}$
10^{-2}	10^{-12}		12	156	96	24	0.036903	$5.0720e^{-13}$
10^{-2}	10^{-14}		17	234	144	36	0.030682	$5.6060e^{-15}$
10^{-3}	10^{-12}		12	156	96	24	0.031425	$5.0520e^{-13}$
10^{-1}		CBHM-QLM	11	139	84	21	0.028553	$2.9490e^{-13}$
		ZOK	–	–	–	–	–	$8.2005e^{-11}$
		ZOA	–	–	–	–	–	$3.44135e^{-12}$
		ADHBM	–	–	–	–	–	$4.1843e^{-12}$
10^{-3}		CBHM-QLM	100	1300	800	200	0.041450	$3.0850e^{-20}$

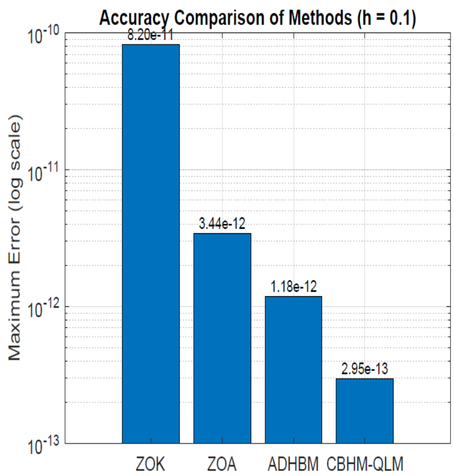


Figure 4: Accuracy comparison for Problem 3.

Table 3 and Figure 4 compare the performance of VBHM-QLM and CBHM-QLM for different step sizes h and tolerance levels. In addition, CBHM-QLM is evaluated against ADHBM, ZOA, and ZOK, which are hybrid block methods of orders $p = 5$, 5, and 7, respectively. The results demonstrate that CBHM-QLM achieves higher accuracy and greater reliability than these existing methods.

Problem 4
Nonlinear IVP from [2]:

$$y''' - y'(2ty'' + y') = 0, \quad y(0) = 0, \quad y'(0) = 0, \quad y''(0) = \frac{1}{2}, \quad t \in [0, 1],$$

with exact solution

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

Table 4: Data of VBHM-QLM for Problem 4.

h	Tol	NS	NFEVAL	NJAC	NITERS	Time_s	MaxErr_y
10^{-1}	10^{-6}	4	76	56	14	0.0254	$1.6190e^{-6}$
10^{-1}	10^{-8}	7	139	104	26	0.0369	$1.4700e^{-6}$
10^{-2}	10^{-6}	8	148	108	27	0.0394	$1.7100e^{-6}$
10^{-2}	10^{-8}	9	190	140	35	0.0318	$1.0460e^{-6}$

Table 5: Absolute error comparison for Problem 4.

t	CBHM-QLM	ADGE
0.1	$5.0340e^{-11}$	$1.9315e^{-8}$
0.2	$1.5250e^{-9}$	$5.6083e^{-7}$
0.3	$5.9790e^{-9}$	$3.7551e^{-6}$
0.4	$1.5230e^{-8}$	$1.3403e^{-5}$
0.5	$3.1540e^{-8}$	$3.2591e^{-5}$
0.6	$5.7840e^{-8}$	$5.8165e^{-5}$
0.7	$9.8160e^{-8}$	$7.1524e^{-5}$
0.8	$1.5820e^{-7}$	$2.5648e^{-4}$
0.9	$2.4640e^{-7}$	$1.7092e^{-4}$
1	$3.7570e^{-7}$	$6.7064e^{-4}$

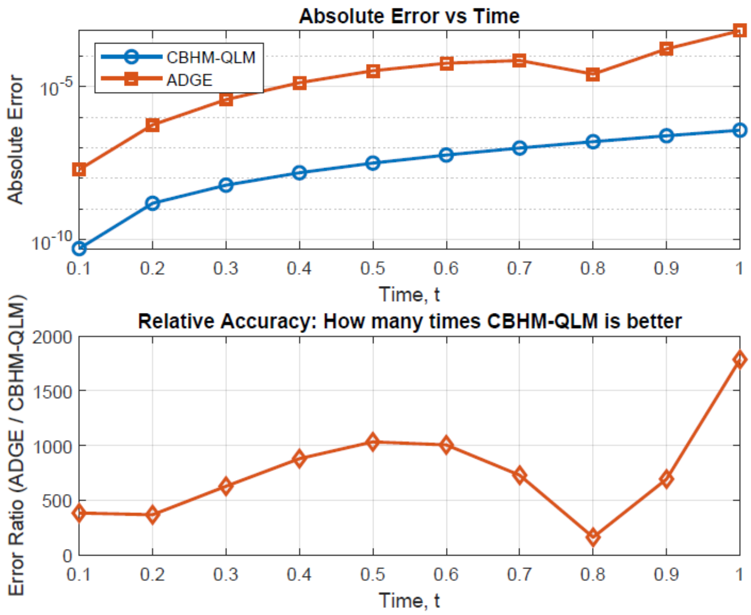


Figure 5: Accuracy comparison for Problem 4.

Table 4 presents the results obtained using VBHM-QLM for Problem 4, while Table 5 and Figure 5 compare the performance of CBHM-QLM and ADGE. CBHM-QLM solves the problem using $h = 0.1$, whereas ADGE, a symmetric implicit higher-order method, uses $h = 0.01$. The results demonstrate that CBHM-QLM achieves significantly higher accuracy than ADGE, despite the latter being a ninth-order method ($p = 9$).

Problem 5

Consider the three-dimensional system from [10]:

$$
\begin{aligned}
y_1''' &= \tfrac{1}{68}(817y_1 + 1393y_2 + 448y_3), \quad y_1(0) = 2, \quad y_2(0) = -2, \quad y_3(0) = -12,\\
y_2''' &= -\tfrac{1}{68}(1141y_1 + 2837y_2 + 896y_3), \quad y_1'(0) = 12, \quad y_2'(0) = 28, \quad y_3'(0) = -33,\\
y_3''' &= \tfrac{1}{136}(3059y_1 + 4319y_2 + 1592y_3), \quad y_1''(0) = 20, \quad y_2''(0) = -52, \quad y_3''(0) = 5,
\end{aligned}
$$

with exact solution

$$
\begin{aligned}
y_1(t) &= e^t - 2e^{2t} + 3e^{-3t},\\
y_2(t) &= 3e^t + 2e^{2t} - 7e^{-3t},\\
y_3(t) &= -11e^t - 5e^{2t} + 4e^{-3t}.
\end{aligned}
$$

Table 6: Performance comparison of maximum error for Problem 5.

N	ROC	CPU Time	CBHM-QLM	ROC	CPU Time	BVM5	ROC	CPU Time	TOMs
10	—	0.0541	$1.6350e^{-8}$	—	0.360	$5.4460e^{-3}$	—	0.359	$1.6360e^{-2}$
20	6.00	0.0577	$2.5490e^{-10}$	5.83	0.469	$9.5900e^{-5}$	6.12	0.374	$2.3480e^{-4}$
40	6.04	0.0411	$3.8760e^{-12}$	5.73	0.609	$1.8040e^{-6}$	6.04	0.547	$3.5480e^{-6}$
80	3.28	0.0610	$3.9790e^{-13}$	5.92	0.981	$2.9810e^{-8}$	6.02	0.719	$5.4580e^{-8}$
160	0.81	0.1105	$2.2740e^{-13}$	5.82	1.249	$5.2910e^{-10}$	6.01	1.062	$8.4730e^{-10}$
320	0.42	0.2711	$1.7050e^{-13}$	5.62	2.620	$1.0740e^{-11}$	6.04	2.093	$1.2850e^{-11}$

Table 6 compares the maximum absolute error, ROC, and CPU time (in seconds) at different step sizes (N) for CBHM-QLM, BVM5, and TOMs, which are methods of orders $p = 5$, 6, and 6, respectively. Here, BVM5 and TOMs denote a boundary value method and top-order methods, respectively. Figure 6 further illustrates their comparative performance. The results indicate that CBHM-QLM consistently outperforms the other methods in terms of accuracy while maintaining competitive computational efficiency.

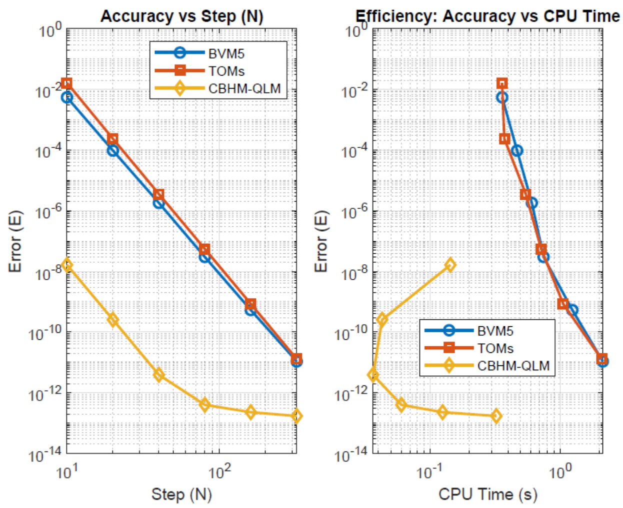


Figure 6: Accuracy comparison for Problem 5.

Problem 6: Thin Film Problem

Finally, we consider the thin film flow problem [1]:

$$y''' - y^{-\tau} = 0, \quad y(0) = y'(0) = y''(0) = 1, \quad t \in [0, 1],$$

where τ is a stiffness parameter. Since there is no closed-form solution, we solve this problem numerically with $\tau = 2$.

Table 7: Comparison of the approximation solution for Problem 6.

t	Exact solution Ref(MRKM)	THMD	ADHBM	CBHM-QLM
0.0	0.000000000	1.000000000	1.000000000	1.000000000
0.2	1.221211030	1.2212084858	1.2212100045	1.221210005
0.4	1.488834893	1.4888467642	1.4888347799	1.488834780
0.6	1.807361404	1.8073467642	1.8073613977	1.807361378
0.8	2.179819234	2.1797930619	2.1798192339	2.179819234
1.0	2.608275822	2.6082338883	2.6082748676	2.608274860

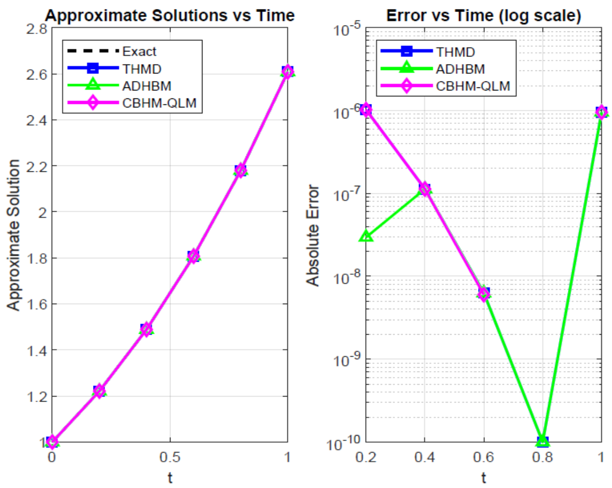


Figure 7: Accuracy comparison for Problem 6.

Table 7 and Figure 7 present the accuracy of THMD, ADHBM, and CBHM-QLM using $h = 0.01$ for Problem 6. The results demonstrate that CBHM-QLM achieves higher accuracy than the alternative methods, namely, THMD and ADHBM, which are one-step block hybrid and third-order Runge–Kutta methods, respectively.

5 Conclusion

This study presents a novel adaptive block hybrid method that integrates quasilinearization with an embedded procedure for the direct solution of general third-order IVPs. The proposed method is A-stable and convergent. Numerical experiments on a variety of test problems demonstrate that the method consistently achieves high accuracy and competitive computational efficiency compared with existing methods. These results demonstrate the effectiveness of the proposed approach and highlight its potential for solving complex dynamical systems and other higher-order differential equations arising in science and engineering.

References

[1] O. Adeyeye and Z. Omar, Solving third order ordinary differential equations using one-step block method with four equidistant generalized hybrid points. IAENG Int. J. Appl. Math., **49** (2019), no. 2, pp. 2–15.

[2] L.O. Adoghe, B.G. Ogunware, B. Nikouravan and E.O. Omole, A family of symmetric implicit higher order methods for the solution of third order initial value problems in ordinary differential equations. Theor. Math. Appl., **6** (2016), no. 3, pp. 67–84.

[3] M. Alkasassbeh and Z. Omar, Hybrid one-step block fourth derivative method for the direct solution of third order initial value problems of ordinary differential equations. Int. J. Pure Appl. Math., **119** (2018), no. 1, pp. 207–224.

[4] R. Allogmany and F. Ismail, Implicit three-point block numerical algorithm for solving third order initial value problem directly with applications. Mathematics, **8** (2020), no. 10, article 1771. https://doi.org/10.3390/math8101771

[5] A.R. Appadu, Optimized low dispersion and low dissipation Runge-Kutta algorithms in computational aeroacoustics. Appl. Math. Inf. Sci., **8** (2014), no. 1, pp. 57–68.

[6] A.R. Appadu, and A.A.I. Peer, Optimized weighted essentially nonoscillatory third-order schemes for hyperbolic conservation laws. J. Appl. Math., **2013** (2013), article ID 428681, pp. 1–12. http://dx.doi.org/10.1155/2013/428681

[7] D.O. Awoyemi, A P-stable linear multistep method for solving general third order ordinary differential equations. Int. J. Comput. Math., **80** (2003), no. 8, pp. 985–991. https://doi.org/10.1080/0020716031000079572

[8] J.C. Butcher, *Numerical Methods for Ordinary Differential Equations*, 3rd ed., John Wiley & Sons, Chichester, 2016.

[9] M.K. Duromola and A.L. Momoh, Hybrid numerical method with block extension for direct solution of third order ordinary differential equations. Am. J. Comput. Math., **9** (2019), no. 2, pp. 68–80. https://doi.org/10.4236/ajcm.2019.92006

[10] S. Jator, T. Okunlola, T. Biala and R. Adeniyi, Direct integrators for the general third-order ordinary differential equations with an application to the Korteweg–de Vries equation. Int. J. Appl. Comput. Math., **4** (2018), article number 110. https://doi.org/10.1007/s40819-018-0542-6

[11] Y.D. Jikantoro, F. Ismail, N. Senu and Z.B. Ibrahim, A new integrator for special third order differential equations with application to thin film flow problem. Indian J. Pure Appl. Math., **49** (2018), no. 1, pp. 151–167. https://doi.org/10.1007/s13226-018-0259-6

[12] L.V. Kantorovich, On Newton's method for functional equations. Doklady Akademii Nauk SSSR, **59** (1948), pp. 1237–1240.

[13] L.V. Kantorovich and G.P. Akilov, *Functional Analysis*, 2nd ed., Pergamon Press, Oxford, 1982.

[14] J.D. Lambert, *Computational Methods in Ordinary Differential Equations*, John Wiley & Sons, London, 1973.

[15] J.O. Kuboye, O.F. Quadri and O.R. Elusakin, Solving third order ordinary differential equations directly using hybrid numerical models. J. Niger. Soc. Phys. Sci., **2** (2020), pp. 69–76. https://doi.org/10.46481/jnsps.2020.43

[16] K.O. Lawal, Y.A. Yahaya and S.D. Yakubu, Four-step block method for solving third order ordinary differential equation. International Journal of Mathematics Trends and Technology, **57** (2018), no. 5, pp. 331–344. https://doi.org/10.14445/22315373/IJMTT-V57P546

[17] Z.A. Majid, N.A. Azmi, M. Suleiman and Z.B. Ibrahim, Solving directly general third order ordinary differential equations using two-point four step block method. Sains Malaysiana, **41** (2012), no. 5, pp. 623–632.

[18] M. Mechee, N. Senu, F. Ismail, B. Nikouravan and Z. Siri, A three-stage fifth-order Runge-Kutta method for directly solving special third-order differential equation with application to thin film flow problem. Math. Probl. Eng., **2013** (2013), article ID 795397, pp. 1–7. https://doi.org/10.1155/2013/795397

[19] S. Mehrkanoon, A direct variable step block multistep method for solving general third-order ODEs. Numer. Algor., **57** (2011), no. 1, pp. 53–66. https://doi.org/10.1007/s11075-010-9413-x

[20] A. Olagunju and E. Adeyefa, Hybrid block method for direct integration of first, second and third order IVPs. Cankaya University Journal of Science and Engineering, **18** (2021), no. 1, pp. 1–8.

[21] Z. Omar and R. Abdelrahim, Application of single step with three generalized hybrid points block method for solving third order ordinary differential equations. J. Nonlinear Sci. Appl., **9** (2016), pp. 2705–2717. https://dx.doi.org/10.22436/jnsa.009.05.67

[22] Z. Omar and J.O. Kuboye, Developing block method of order seven for solving third order ordinary differential equations directly using multistep collocation approach. Int. J. Appl. Math. Stat., **53** (2015), no. 3, pp. 165–173.

[23] A.L. Osa and O.E. Olaoluwa, A fifth-fourth continuous block implicit hybrid method for the solution of third order initial value problems in ordinary differential equations. Appl. Comput. Math., **8** (2019), no. 3, pp. 50–57. https://doi.org/10.11648/j.acm.20190803.11

[24] M.A. Rufai and H. Ramos, A variable step-size fourth-derivative hybrid block strategy for integrating third-order IVPs, with applications. Int. J. Comput. Math., **99** (2022), no. 2, pp. 292–308.

[25] S.D. Yakubu, Family of one-step A-stable optimized third derivative hybrid block methods for solving general second-order IVPs. AL-Rafidain Journal of Computer Sciences and Mathematics, **17** (2023), no. 2, pp. 125–140. https://doi.org/10.33899/csmj.2023.143686.1086

[26] S.D. Yakubu and P. Sibanda, A novel one-step optimized hybrid block method for solving general second-order ordinary differential equations. Recent Advances in Natural Sciences, **3** (2025), no. 1, article ID 135, pp. 1–12. https://doi.org/10.61298/rans.2025.3.1.135

[27] S.D. Yakubu and P. Sibanda, Implicit one-step optimized fourth-derivative hybrid block method for directly solving general third-order IVPs. Recent Advances in Natural Sciences, **3** (2025), no. 2, article ID 142, pp. 1–12. https://doi.org/10.61298/rans.2025.3.2.142

Saidu D. Yakubu
School of Mathematics, Statistics and Computer Science,
University of KwaZulu-Natal, Pietermaritzburg, South Africa;
Department of Mathematics,
Ibrahim Badamasi Babangida University, Lapai, Nigeria.
saiduyakubu2014@gmail.com

Precious Sibanda
School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Pietermaritzburg, South Africa.
sibandap@ukzn.ac.za

Sicelo P. Goqo
School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Pietermaritzburg, South Africa.
goqos@ukzn.ac.za

Saheed O. Akindeinde
Department of Mathematics and Statistical Sciences, Botswana International University of Science and Technology, Palapye, Botswana.
akindeindes@biust.ac.bw

Please, cite to this paper as published in
Armen. J. Math., V. **18**, N. 9(2026), pp. 1–29
https://doi.org/10.52737/18291163-2026.18.09-1-29