Unilag Journal of Mathematics and Applications,
Volume 5, Issue 1 (2025), Pages 83–109.
ISSN: 2805 3966. URL: http://lagjma.edu.ng

A DERIVATIVE–FREE BLOCK HYBRID METHOD FOR NUMERICAL QUADRATURE

RICHARD O. AKINOLA*, EZEKIEL O. OMOLE, JOSHUA SUNDAY, AND ELEOJO R. AKOR

ABSTRACT. We derive a new ninth-order block hybrid method for the numerical solution of systems of differential equations and we compare results of numerical experiments with an already existing method in the literature. Both methods are bye-products of linear multistep methods using the interpolation and collocation approach. We show computationally that in the absence of round-off errors, the solution obtained by solving systems of differential equations by the existing block hybrid method derived by differentiating the continuous scheme at a particular off–grid point is the same as those obtained in the new method which is derivative free. Besides, the new block hybrid method which is derivative free results in a well conditioned system as opposed to the ill–conditioned one in the literature. Therefore, providing an answer to Shampine's claim that matrices arising from the numerical approximation of stiff initial value problems using Linear Multistep Methods are mostly ill–conditioned. Finally, we showed computationally how an LU-type preconditioned Quasi Minimal Residual with a fixed default tolerance reduced the condition number of the old and new methods, with the latter resulting in the smallest minimum norm of residual.

1. INTRODUCTION

In this paper, we derive a new ninth-order, self-starting, zero and $A(\alpha)$-stable, 'derivative-free' and convergent block hybrid method for the numerical integration of (non)-linear, (non)-stiff systems of differential equations. Both methods are bye-products of linear multistep methods using the collocation approach which is not strange why ill-conditioning arose from the matrices derived from them confirming the statement made by Shampine [1]. We show computationally that in the absence of round-off errors, the solution obtained by solving systems of differential equations by the existing block hybrid method [2] derived by differentiating the continuous scheme at a particular off-grid

2010 *Mathematics Subject Classification.* 65D30, 65L04, 65L05, 65L06.

Key words and phrases. Block hybrid methods, non-singular, A(α)-stable, Convergence, ill-conditioning .

Submitted: June 25, 2025. Revised: September 4, 2025. Accepted: September 24, 2025.

* Correspondence.

point is the same as those obtained by the new method which is derivative free. Several other authors Adee *et al* [3], Aboiyar *et al* [4], Fotta *et al* [5], Ayinde *et al* [6], [7, 8, 9], Odejinde and Adeniran [10] have used the 'derivative-free' approach in deriving block hybrid methods without neither mentioning ill-conditioning nor any comparison made with other methods which used it.

The term 'derivative-free' is loosely used in this context to mean that the new block hybrid method does not involve finding the derivative of the continuous scheme at any off-grid. In addition to this, we compared results of numerical experiments with the exact solution (where it exists) as well as an already existing method in the literature. Besides, in all numerical examples considered in this paper the new block hybrid method gives better conditioned system with condition numbers less than a quarter of those obtained in [2]. The condition number of a matrix is defined as the ratio of the largest to the smallest singular value of a matrix [11]. A high condition number means solving an ill–conditioned system.

1.1. **Literature Review.** Ill-conditioning is a phenomenum often encountered when solving systems of linear equations and aside the fact that it leads to loss of the logarithm of the condition number significant digits, it also means one is solving a nearly singular system [12, 13, 14, 15]. This means there is a need to find an appropriate preconditioner to reduce the condition numbers. We used a LU-type [16] preconditioner with a Quasi-Minimal Residual (QMR) iterative solver [17, 18, 19, 20, 21, 22, 23, 24]. The choice of the LU-type preconditioner stems from the work of Gogoleva [25], while QMR is informed by Demmel's [26, p. 321] decision tree for choosing a particular iterative method. Besides, using the default tolerance, both GMRES and Bi-CGStab did not give better approximations except QMR albeit due to the ill–conditioning of the resultant matrices.

Furthermore, the results of numerical experiments where an LU-type preconditioned Quasi Minimal Residual (QMR) with a fixed default tolerance showed that the new block hybrid method gives better norm of residual than the one in Akinola *et al* [2]. Hence, going forward, this paper serves as a precaution as there is no need to differentiate the continuous scheme and evaluating at any off–grid point in deriving first–order derivative methods for the numerical integration of first–order IVPs.

In the earlier work, the method shares loads of perculiarities with the present work in the sense that they are both of 9th–order, zero and $A(\alpha)$-stable albeit $\alpha = 74^\circ$ ($\alpha = 10^\circ$ in the present work) and convergent. In fact, as shown with numerical examples in [2], the existing method has less number of function evaluations and out-performed a 14th–order method presented in [27]. Nevertheless, as will be discovered in this paper, it suffers the disadvantage that the corresponding matrix obtained in comparison to the one in this work had very high condition numbers.

Omole [28] used 4-grid and four–off grid points: $\{\frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \frac{7}{2}\}$ for the solution of fourth order initial value problems. In the earlier work [2], we used the following three-off grid points $\{\frac{3}{2}, \frac{5}{2}, \frac{7}{2}\}$ and the four-grid points $\{1, 2, 3, 4\}$ as interpolating points while in this present work, we retained the same four-grid points but added the off grid point $\frac{9}{2}$ resulting in the following four-off grid interpolating points $\{\frac{3}{2}, \frac{5}{2}, \frac{7}{2}, \frac{9}{2}\}$. It should be mentioned that while the added off-grid point was an interpolating point in the present work, in the former, it was a collocation point, for more on off grid points the interested reader should read [29, 30, 31, 32, 33, 34, 35].

Some authors have used Hermite, Legendre, Laguerre, Chebyshev polynomials as basis functions in deriving their numerical methods, but here, we use the interpolation and collocation approach of Onumanyi et al [36].

2. Materials and Methods

In this section, we present the new block hybrid method, show that it is of order nine and present the corresponding error constant. This is then followed by showing that the new method is zero stable, convergent, $A(\alpha)$-stable with $\alpha = 10^\circ$ and to cap it all we present a newton-based algorithm for the new block hybrid method. We begin by pointing out the marked differences between the new method and the one in [2].

In describing the new method, we assume a first derivative block hyrid method

$$y_{n+j} = \alpha_0(x)y_n + h\sum_j \beta_j(x)f_{n+j},$$

for $j \in \{0, 1, \frac{3}{2}, 2, \frac{5}{2}, 3, \frac{7}{2}, 4, \frac{9}{2}\}$. We used

$$y(x) = \sum_{i=0}^{9} a_i x^i, \tag{2.1}$$

as a basis function where the a_i's are nonzero polynomial coefficients. In this context, for $j \in \{0, 1, \frac{3}{2}, 2, \frac{5}{2}, 3, \frac{7}{2}, 4, \frac{9}{2}\}$, we arrived at the 10 by 10 system of equations

$$\begin{aligned} y(x_n) &= y_n \\ y'(x_{n+j}) &= f_{n+j}, \end{aligned}$$

from where the a_i's are found. Here, a_0 equals α_0; the a_i's for $i = 1(1)9$ respectively equals $\beta_0, \beta_1, \beta_{\frac{3}{2}}, \beta_2, \beta_{\frac{5}{2}}, \beta_3, \beta_{\frac{7}{2}}, \beta_4, \beta_{\frac{9}{2}}$. Plugging the continuous coefficients into the continuous formulation,

$$\begin{aligned} y(x) &= \alpha_0(x)y_n + h\sum_j \beta_j(x)f_{n+j} \\ &= \alpha_0(x)y_n + h\big[\beta_0(x)f_n + \beta_1(x)f_{n+1} + \beta_{\frac{3}{2}}(x)f_{n+\frac{3}{2}} + \beta_2(x)f_{n+2} \\ &+ \beta_{\frac{5}{2}}(x)f_{n+\frac{5}{2}} + \beta_3(x)f_{n+3} + \beta_{\frac{7}{2}}(x)f_{n+\frac{7}{2}} + \beta_4(x)f_{n+4} + \beta_{n+\frac{9}{2}(x)f_{n+\frac{9}{2}}}\big]. \end{aligned}$$

If we let $w = x_{n+1} - x$ for ease of notation, then the continuous coefficients are as shown in Appendix A. Notice that unlike the earlier work in [2] in which to obtain the discrete scheme for y_{n+1}, we differentiated the continuous formulation before evaluating it at $w = -\frac{7h}{2}$. Here, to obtain the discrete scheme for y_{n+1}, we did not differentiate the continuous scheme at all and that is why we used the term ”derivative–free” in the title of this paper, rather we evaluated it at $w = 0$. The derivation of the new block hybrid method is as explained below.

We evaluated the continuous formulation at $w = 0, -\frac{h}{2}, -h, -\frac{3h}{2}, -2h, -\frac{5h}{2}, -3h, -\frac{7h}{2}$, we obtained respectively the discrete schemes which becomes the block hybrid method

$$
\begin{aligned}
y_{n+1} &= y_n + \frac{h\left[473977f_n + 6190578f_{n+1} - 14256264f_{n+\frac{3}{2}} + 21960504f_{n+2} - 22333032f_{n+\frac{5}{2}}\right]}{2041200} \\
&+ \frac{h\left[15056670f_{n+3} - 6504408f_{n+\frac{7}{2}} + 1635759f_{n+4} - 182584f_{n+\frac{9}{2}}\right]}{2041200}, \quad (2.2) \\
y_{n+\frac{3}{2}} &= y_n + \frac{h\left[20759f_n + 287046f_{n+1} - 581818f_{n+\frac{3}{2}} + 936468f_{n+2} - 958194f_{n+\frac{5}{2}} + 647690f_{n+3}\right]}{89600} \\
&- \frac{h\left[280206f_{n+\frac{7}{2}} - 70533f_{n+4} + 7878f_{n+\frac{9}{2}}\right]}{89600}, \quad (2.3) \\
y_{n+2} &= y_n + \frac{h\left[59143f_n + 814932f_{n+1} - 1601616f_{n+\frac{3}{2}} + 2762856f_{n+2} - 2761488f_{n+\frac{5}{2}}\right]}{255150} \\
&+ \frac{h\left[1860780f_{n+3} - 803952f_{n+\frac{7}{2}} + 202221f_{n+4} - 22576f_{n+\frac{9}{2}}\right]}{255150}, \quad (2.4) \\
y_{n+\frac{5}{2}} &= y_n + \frac{h\left[605495f_n + 8353350f_{n+1} - 16467450f_{n+\frac{3}{2}} + 28962900f_{n+2} - 27460530f_{n+\frac{5}{2}}\right]}{2612736} \\
&+ \frac{h\left[18890250f_{n+3} - 8182350f_{n+\frac{7}{2}} + 2060325f_{n+4} - 230150f_{n+\frac{9}{2}}\right]}{2612736}, \quad (2.5) \\
y_{n+3} &= y_n + \frac{h\left[649f_n + 8946f_{n+1} - 17608f_{n+\frac{3}{2}} + 30888f_{n+2} - 28584f_{n+\frac{5}{2}}\right]}{2800} \\
&+ \frac{h\left[20990f_{n+3} - 8856f_{n+\frac{7}{2}} + 2223f_{n+4} - 248f_{n+\frac{9}{2}}\right]}{2800}, \quad (2.6) \\
y_{n+\frac{7}{2}} &= y_n + \frac{h\left[2162377f_n + 29837178f_{n+1} - 58823814f_{n+\frac{3}{2}} + 103389804f_{n+2} - 96271182f_{n+\frac{5}{2}}\right]}{9331200} \\
&+ \frac{h\left[73295670f_{n+3} - 27390258f_{n+\frac{7}{2}} + 7276059f_{n+4} - 816634f_{n+\frac{9}{2}}\right]}{9331200}, \quad (2.7) \\
y_{n+4} &= y_n + \frac{h\left[29578f_n + 407232f_{n+1} - 800256f_{n+\frac{3}{2}} + 1402056f_{n+2} - 1294848f_{n+\frac{5}{2}}\right]}{127575} \\
&+ \frac{h\left[972480f_{n+3} - 317952f_{n+\frac{7}{2}} + 123786f_{n+4} - 11776f_{n+\frac{9}{2}}\right]}{127575}, \quad (2.8) \\
y_{n+\frac{9}{2}} &= y_n + \frac{h\left[20727f_n + 288198f_{n+1} - 574074f_{n+\frac{3}{2}} + 1017684f_{n+2} - 965682f_{n+\frac{5}{2}}\right]}{89600} \\
&+ \frac{h\left[1748170f_{n+3} - 278478f_{n+\frac{7}{2}} + 141669f_{n+4} + 4986f_{n+\frac{9}{2}}\right]}{89600}. \quad (2.9)
\end{aligned}
$$

From the above block hybrid schemes, we obtained the following vectorized continuous coefficients that will be used in calculating the order:

$$\alpha_0 = -\begin{bmatrix}1\\1\\1\\1\\1\\1\\1\\1\end{bmatrix}, \quad \alpha_1 = \begin{bmatrix}1\\0\\0\\0\\0\\0\\0\\0\end{bmatrix}, \quad \alpha_{\frac{3}{2}} = \begin{bmatrix}0\\1\\0\\0\\0\\0\\0\\0\end{bmatrix}, \quad \alpha_2 = \begin{bmatrix}0\\0\\1\\0\\0\\0\\0\\0\end{bmatrix}, \quad \alpha_{\frac{5}{2}} = \begin{bmatrix}0\\0\\0\\1\\0\\0\\0\\0\end{bmatrix},$$

and

$$\alpha_3 = \begin{bmatrix}0\\0\\0\\0\\1\\0\\0\\0\end{bmatrix}, \quad \alpha_{\frac{7}{2}} = \begin{bmatrix}0\\0\\0\\0\\0\\1\\0\\0\end{bmatrix}, \quad \alpha_4 = \begin{bmatrix}0\\0\\0\\0\\0\\0\\1\\0\end{bmatrix}, \quad \alpha_{\frac{9}{2}} = \begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\1\end{bmatrix}.$$

In the same vein,

$$\beta_0 = \begin{bmatrix} \frac{67711}{291600} \\ \frac{20759}{89600} \\ \frac{8449}{36450} \\ \frac{605495}{2612736} \\ \frac{649}{2800} \\ \frac{2162377}{9331200} \\ \frac{29578}{127575} \\ \frac{2961}{12800} \end{bmatrix}, \quad \beta_1 = \begin{bmatrix} \frac{343921}{113400} \\ \frac{143523}{44800} \\ \frac{45274}{14175} \\ \frac{464075}{145152} \\ \frac{639}{200} \\ \frac{1657621}{518400} \\ \frac{6464}{2025} \\ \frac{144099}{44800} \end{bmatrix}, \quad \beta_{\frac{3}{2}} = - \begin{bmatrix} \frac{594011}{85050} \\ \frac{290909}{44800} \\ \frac{266936}{42525} \\ \frac{2744575}{435456} \\ \frac{2201}{350} \\ \frac{9803969}{1555200} \\ \frac{266752}{42525} \\ \frac{287037}{44800} \end{bmatrix}, \quad \beta_2 = \begin{bmatrix} \frac{101669}{9450} \\ \frac{234117}{22400} \\ \frac{51164}{4725} \\ \frac{268175}{24192} \\ \frac{3861}{350} \\ \frac{957313}{86400} \\ \frac{51928}{4725} \\ \frac{254421}{22400} \end{bmatrix}, \quad \beta_{\frac{5}{2}} = - \begin{bmatrix} \frac{310181}{28350} \\ \frac{479097}{44800} \\ \frac{153416}{14175} \\ \frac{1525585}{145152} \\ \frac{3573}{350} \\ \frac{5348399}{518400} \\ \frac{143872}{14175} \\ \frac{482841}{44800} \end{bmatrix},$$

and

$$\beta_3 = \begin{bmatrix} \frac{501889}{68040} \\ \frac{64769}{8960} \\ \frac{62026}{8505} \\ \frac{3148375}{435456} \\ \frac{2099}{280} \\ \frac{2443189}{311040} \\ \frac{64832}{8505} \\ \frac{74817}{8960} \end{bmatrix}, \quad \beta_{\frac{7}{2}} = - \begin{bmatrix} \frac{30113}{9450} \\ \frac{140103}{44800} \\ \frac{14888}{4725} \\ \frac{151525}{48384} \\ \frac{1107}{350} \\ \frac{507227}{172800} \\ \frac{11776}{4725} \\ \frac{139239}{44800} \end{bmatrix}, \quad \beta_4 = \begin{bmatrix} \frac{181751}{226800} \\ \frac{70533}{89600} \\ \frac{22469}{28350} \\ \frac{228925}{290304} \\ \frac{2223}{2800} \\ \frac{808451}{1036800} \\ \frac{13754}{14175} \\ \frac{141669}{89600} \end{bmatrix}, \quad \beta_{\frac{9}{2}} = - \begin{bmatrix} \frac{22823}{255150} \\ \frac{3939}{44800} \\ \frac{11288}{127575} \\ \frac{115075}{1306368} \\ \frac{31}{350} \\ \frac{408317}{4665600} \\ \frac{11776}{127575} \\ -\frac{2493}{44800} \end{bmatrix}.$$

Next, we state the following lemma with a proof. **Lemma 1.** Each of the discrete schemes that constitute the block hybrid method (3)–(10) has order nine.

Proof. Substituting the above values of α's, β's into the formula for calculating the order of a Linear Multistep Method and after appropriate algebraic simplifications as shown in [37] and [38], we have $\mathbf{C_0} = \mathbf{C_1} = \mathbf{C_2} = \mathbf{C_3} = \mathbf{C_4} = \mathbf{C_5} = \mathbf{C_6} = \mathbf{C_7} = \mathbf{C_8} = \mathbf{C_9} = \mathbf{0}$. The non–zero error constants ($\mathbf{C_{10}}$) are as tabulated in Table 1.

TABLE 1. Error constant of each of the discrete schemes that constitute the block hybrid method.

y_i	Error Constant $\mathbf{C_{10} \neq 0}$
y_{n+1}	$1.809836508548893 \times 10^{-4}$
$y_{n+\frac{3}{2}}$	$1.785387311662946 \times 10^{-4}$
y_{n+2}	$1.792909807956104 \times 10^{-4}$
$y_{n+\frac{5}{2}}$	$1.788583195211839 \times 10^{-4}$
y_{n+3}	$1.792689732142857 \times 10^{-4}$
$y_{n+\frac{7}{2}}$	$1.786382437079368 \times 10^{-4}$
y_{n+4}	$1.802861062120321 \times 10^{-4}$
$y_{n+\frac{9}{2}}$	$1.715632847377232 \times 10^{-4}$

2.1. **Zero Stability, Convergence and Region of Absolute Stability of the New Block Hybrid Method.** In this section, we examine the stability properties of the new block hybrid method and plot its region of absolute stability.

We start by re-writing the block method in the form :

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_{n+1} \\ y_{n+\frac{3}{2}} \\ y_{n+2} \\ y_{n+\frac{5}{2}} \\ y_{n+3} \\ y_{n+\frac{7}{2}} \\ y_{n+4} \\ y_{n+\frac{9}{2}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_{n-7} \\ y_{n-6} \\ y_{n-5} \\ y_{n-4} \\ y_{n-3} \\ y_{n-2} \\ y_{n-1} \\ y_{n} \end{bmatrix} + hZ \begin{bmatrix} f_{n+1} \\ f_{n+\frac{3}{2}} \\ f_{n+2} \\ f_{n+\frac{5}{2}} \\ f_{n+3} \\ f_{n+\frac{7}{2}} \\ f_{n+4} \\ f_{n+\frac{9}{2}} \end{bmatrix}$$

$$+ h \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{67711}{291600} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{20759}{89600} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{8449}{36450} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{605495}{2612736} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{649}{2800} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2162377}{9331200} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{29578}{127575} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2961}{12800} \end{bmatrix} \begin{bmatrix} f_{n-7} \\ f_{n-6} \\ f_{n-5} \\ f_{n-4} \\ f_{n-3} \\ f_{n-2} \\ f_{n-1} \\ f_{n} \end{bmatrix},$$

where

$$Z = \begin{bmatrix} \frac{343921}{113400} & -\frac{594011}{85050} & \frac{101669}{9450} & -\frac{310181}{28350} & \frac{501889}{68040} & -\frac{30113}{9450} & \frac{181751}{226800} & -\frac{22823}{255150} \\ \frac{143523}{44800} & -\frac{290909}{44800} & \frac{234117}{22400} & -\frac{479097}{44800} & \frac{64769}{8960} & -\frac{140103}{44800} & \frac{70533}{89600} & -\frac{3939}{44800} \\ \frac{45274}{14175} & -\frac{266936}{42525} & \frac{51164}{4725} & -\frac{153416}{14175} & \frac{62026}{8505} & -\frac{14888}{4725} & \frac{22469}{28350} & -\frac{11288}{127575} \\ \frac{464075}{145152} & -\frac{2744575}{435456} & \frac{268175}{24192} & -\frac{1525585}{145152} & \frac{3148375}{435456} & -\frac{151525}{48384} & \frac{228925}{290304} & -\frac{115075}{1306368} \\ \frac{639}{200} & -\frac{2201}{350} & \frac{3861}{350} & -\frac{3573}{350} & \frac{2099}{280} & -\frac{1107}{350} & \frac{2223}{2800} & -\frac{31}{350} \\ \frac{1657621}{518400} & -\frac{9803969}{1555200} & \frac{957313}{86400} & -\frac{5348399}{518400} & \frac{2443189}{311040} & -\frac{507227}{172800} & \frac{808451}{1036800} & -\frac{408317}{4665600} \\ \frac{6464}{2025} & -\frac{266752}{42525} & \frac{51928}{4725} & -\frac{143872}{14175} & \frac{64832}{8505} & -\frac{11776}{4725} & \frac{13754}{14175} & -\frac{11776}{127575} \\ \frac{144099}{44800} & -\frac{287037}{44800} & \frac{254421}{22400} & -\frac{482841}{44800} & \frac{74817}{8960} & -\frac{139239}{44800} & \frac{141669}{89600} & \frac{2493}{44800} \end{bmatrix}.$$

From the above equation and for the purpose of the discussion in this section, we define the matrices W, X and Y as follows:

$$W = \begin{bmatrix} \alpha_1 & \alpha_{\frac{3}{2}} & \alpha_2 & \alpha_{\frac{5}{2}} & \alpha_3 & \alpha_{\frac{7}{2}} & \alpha_4 & \alpha_{\frac{9}{2}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$X = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\alpha_0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

and

$$Y = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{67711}{291600} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{20759}{89600} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{8449}{36450} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{605495}{2612736} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{649}{2800} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2162377}{9331200} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{29578}{127575} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2961}{12800} \end{bmatrix}.$$

Now, using the matrices W and X as defined above, we study the zero stability of the block hybrid method as shown below

$$\det(RW - X) = R^8 - R^7 = 0,$$

R equals zero (multiplicities 7) and R equals one. The new method is zero stable because the roots of the characteristic polynomial (i.e., $R = 0$ has multiplicities seven and $R = 1$ as shown above) have respectively modulus zero which is less than or equal to one and those of modulus one (i.e, $R = 1$) is distinct. In addition to this, according to Henrici [39], the new block hybrid method is consistent because each of the discrete schemes that constitute the block is of order nine as shown in Lemma 1 which is greater than one. Therefore, since the new method is zero stable and consistent, it is convergent (see, for example, [40, 41, 42]).

In plotting the region of absolute stability of the method, we used the stability polynomial

$$M(z) = |Ww - X - Yw - Zwz|,\ z = \lambda h,\ w = e^{i\theta},\ i^2 = -1,\ \theta \in [0, 2\pi],$$

and $y' = \lambda y$ is the usual test equation. Thus,

$$\begin{aligned} M(z) &= |Ww - X - Yw - Zwz| \\ &= \frac{7560w^8 - 945w^7)z^8 + (-55308w^8 - 10221w^7)z^7 + (254502w^8 - 65628w^7)z^6}{1935360} \\ &\quad + \frac{(-858704w^8 - 296212w^7)z^5 + (2185960w^8 - 976360w^7)z^4 + (-4139520w^8 - 2331840w^7)z^3}{1935360} \\ &\quad + \frac{(5550720w^8 - 3857280w^7)z^2 + (-4730880w^8 - 3978240w^7)z + 1935360w^8 - 1935360w^7}{1935360}. \end{aligned}$$

Newton's method was used to obtain the roots of $M(z) = 0$ where the region of absolute stability is defined as

$$E(w, z) = \{z \in \mathbb{C} : \rho(w, z) = 1, |w| \le 1\},$$

which is as shown in Figure 1.

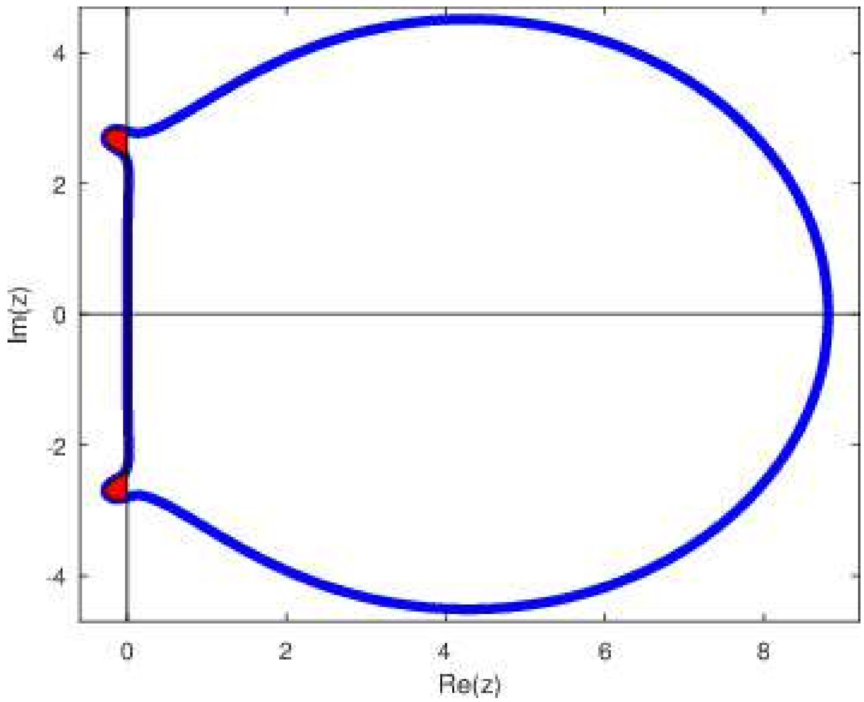


FIGURE 1. The stability region of the ninth–order block hybrid method are the red patches on the imaginary axis.

The above analysis now leads to the following algorithm and implementation.

2.2. **Implementation and Algorithm.** In this section, we show how Newton's method is implemented in solving the nonlinear system of eight equations in eight unknowns of (3)–(10) *i.e.*, $\mathbf{F}(\mathbf{y}) = \mathbf{0}$ where the unknowns are y_i, for

$$i = \left\{n+1, n+\frac{3}{2}, n+2, n+\frac{5}{2}, n+3, n+\frac{7}{2}, n+4, n+\frac{9}{2}\right\}.$$

We make it categorically clear at this juncture that we are only interested in the unknown y_{n+1} but the contribution from the remaining contribute to the accuracy of the block hybrid method. Newton's method applied to $\mathbf{F}(\mathbf{y}) = \mathbf{0}$ is as follows

$$\mathbf{G}(\mathbf{y}^{(k)})\Delta\mathbf{y}^{(k)} = -\mathbf{F}(\mathbf{y}^{(k)}), \tag{2.10}$$

where $\mathbf{G}(\mathbf{y}^{(k)})$ is the Jacobian of $\mathbf{F}(\mathbf{y})$. Find the PLU factorization of $\mathbf{G}(\mathbf{y}^{(k)})$, *i. e.*,
$P\mathbf{G}(\mathbf{y}^{(k)}) = LU$, where P is a permutation matrix, L and U are lower and upper triangular matrices. With $P\mathbf{G}(\mathbf{y}^{(k)}) = -P\mathbf{F}(\mathbf{y})$, if we let $\mathbf{r} = -P\mathbf{F}(\mathbf{y}^{(k)})$, then we first solve $L\mathbf{v} = \mathbf{r}$ and $U\Delta\mathbf{y}^{(k)} = \mathbf{v}$ for $\Delta\mathbf{y}^{(k)}$. Only one PLU factorization is needed at each iteration. The algorithm self starts and does not need any predictor–corrector to predict the starting values. The algorithm is given below

Algorithm 2.1. **Input:** Step size, `tol`, the differential equation and corresponding initial conditions, Jacobian of the new method.
For $k = 0, 1, 2, \cdots,$

(1) Evaluate $\mathbf{F}(\mathbf{y}^{(k)})$ from (3)–(10).
(2) Factorize $[L, U, P] = LU(\mathbf{G}(\mathbf{y}^{(k)}))$.
(3) Solve $L\mathbf{v}^{(k)} = \mathbf{r}^{(k)}$ for $\mathbf{v}^{(k)}$.
(4) Solve $U\Delta\mathbf{y}^{(k)} = \mathbf{v}^{(k)}$ for $\Delta\mathbf{y}^{(k)}$.
(5) Update $\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \Delta\mathbf{y}^{(k)}$.
(6) Continue until Netwon's method converges.

Output: y_{k+1}.

Stop the algorithm as soon as $\|\Delta\mathbf{y}^{(k)}\| <$ `tol` and $\|\mathbf{F}(\mathbf{y}^{(k)})\| <$ `tol`, where `tol` is a user defined tolerance.

3. Results and discussion

In this section, we present the result of seven numerical experiments to ascertain the performance of our method in comparison with those of Akinola et al [2], [43], [44], [45] and [46]. Results are presented by means of figures and tables. Throughout this section and where necessary, we computed the `cputime` thrice before finding the average.

Example 3.1. The non–linear stiff problem [43]

$$u_1'(x) = -2u_1(x) + u_2(x) + 2\sin(x), \quad u_1(0) = 2,$$
$$u_2'(x) = 998u_1(x) - 999u_2(x) + 999(\cos(x) - \sin(x)), \quad u_2(0) = 3.$$

The solution to this problem can be found in Figure 2 and Table 2. The result of the table shows that the new method outperformed those of Yakubu and Sibanda [43] which is a very recent study.

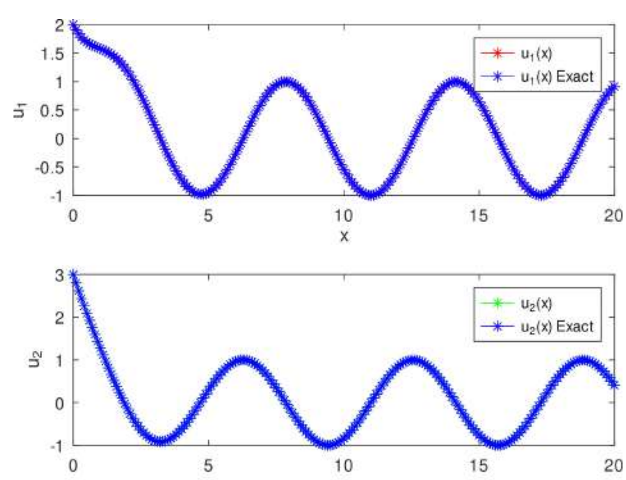


FIGURE 2. The solution of Example 3.1 and the exact on $x = [0, 20]$ with $h = 0.1$.

TABLE 2. Max. Error comparison table for Example 3.1.

h	Max Error [43]	Max Error [44]	Max Error (Our Method)
4×10^{-1}	4.9×10^{-04}	8.9×10^{-07}	3.0×10^{-08}
2×10^{-1}	2.2×10^{-09}	5.9×10^{-09}	6.4×10^{-11}
1×10^{-1}	3.9×10^{-11}	4.5×10^{-11}	1.1×10^{-13}
5×10^{-2}	6.4×10^{-13}	2.9×10^{-13}	1.5×10^{-14}

Example 3.2. The Van–der Pol Oscillator

$$u_1'(t) = u_2(t), \;\; u_1(0) = 2,$$
$$u_2'(t) = \mu(u_2(t) - u_1^2(t)u_2(t)) - u_1(t), \;\; u_2(0) = 0 \;\text{ and }\; \mu = 5.$$

We compared the results of our method with those of `ode15s` and it is shown in Figure 3.

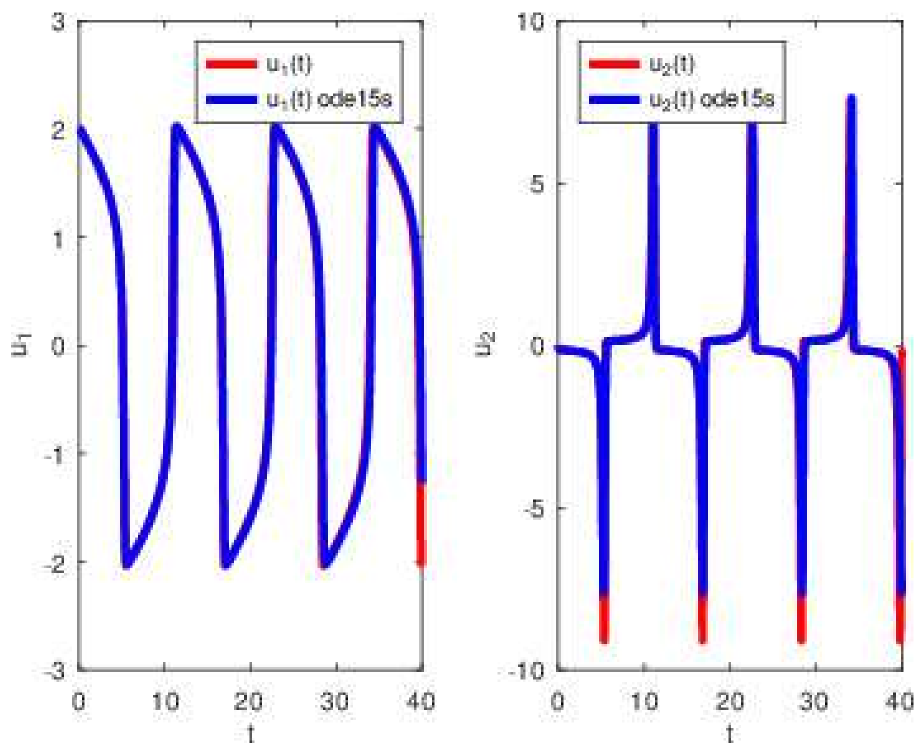


FIGURE 3. Result of Example 3.2 for $h = 0.01$.

Example 3.3. We consider the system [43]

$$\begin{aligned} v_1'(t) &= -21v_1 + 19v_2 - 20v_3 \\ v_2'(t) &= 19v_1 - 21v_2 + 20v_3 \\ v_3'(t) &= 40v_1 - 40v_2 - 40v_3, \end{aligned}$$

TABLE 3. Max. Error comparison table for Example 3.3

Steps	Max Error [43]	Max Error [45]	Max Error [46]	Max Error (Our Method)
2.0×10^{1}	3.3×10^{-03}	3.0×10^{-02}	7.7×10^{-02}	7.8×10^{-03}
4.0×10^{1}	1.1×10^{-04}	3.5×10^{-03}	1.2×10^{-02}	2.6×10^{-04}
8.0×10^{1}	2.4×10^{-06}	2.2×10^{-04}	1.9×10^{-04}	1.3×10^{-06}
1.6×10^{2}	4.1×10^{-08}	5.8×10^{-06}	1.5×10^{-06}	5.8×10^{-09}
3.2×10^{2}	6.5×10^{-10}	1.1×10^{-07}	3.1×10^{-08}	1.6×10^{-11}

on the interval $[0,4]$ with $[v_1(0), v_2(0), v_3(0)]^T = [1, 0, -1]^T$ and exact solution

$$\begin{aligned} v_1(t) &= 0.5[\exp(-2t) + \exp(-40t)(\cos(40t) + \sin(40t))] \\ v_2(t) &= 0.5[\exp(-2t) - \exp(-40t)(\cos(40t) + \sin(40t))] \\ v_3(t) &= \exp(-40t)[\sin(40t) - \cos(40t)]. \end{aligned}$$

The results of this example are in Table 3.

The last column of Table 3 shows that our method performed better than those of [43], [45] and [46].

Example 3.4. The non–linear stiff Kaps problem [47]

$$\begin{bmatrix} u_1'(x) \\ \\ u_2'(x) \end{bmatrix} = \begin{bmatrix} -(1002u_1(x) + 1000u_2^2(x)) \\ \\ u_1(x) - u_2(x) - u_2^2(x) \end{bmatrix}, \text{ with } u_1(0) = 1, u_2(0) = 1.$$

The solution is $u_1(x) = \exp(-2x)$ and $u_2(x) = \exp(-x)$. Results of numerical experiments using a constant step size of $h = 0.1$ are shown in Table 4.

Though the new method and the one in [2] are different, Table 4 shows that in the absence of round-off errors, the results are the same. In fact, in this example, the condition number of the matrix of the system solved at the root by the hybrid method in Akinola et al [2], is more than four times (92296 versus 22860) that of the new method which is derivative free. Though one has a smaller condition number than the other but in fairness, both condition numbers are large and that is why in the last section of this paper, we show how these can be circumvented via preconditioning. Besides, as shown in columns four and seven the `cputime` is lesser than that of Akinola et al [2]. Results of simulation showed that our method performed at par with the exact solution for both $u_1(x)$ and $u_2(x)$.

Example 3.5. The non-linear stiff problem of Gear [48] which has no solution,

$$\begin{aligned} u_1'(x) &= -0.013u_1 - 1000u_1u_3 \\ u_2'(x) &= -2500u_2u_3 \\ u_3'(x) &= -0.013u_1 - 1000u_1u_3 - 2500u_2u_3, \end{aligned}$$

with initial conditions

$$\begin{bmatrix} u_1(0) \\ u_2(0) \\ u_3(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix},$$

and $h = 0.1$.

Example 3.6. The linear stiff IVP [49]

$$\begin{bmatrix} u_1'(x) \\ u_2'(x) \\ u_3'(x) \\ u_4'(x) \end{bmatrix} = \begin{bmatrix} -u_1(x) \\ -10u_2(x) \\ -100u_3(x) \\ -1000u_4(x) \end{bmatrix}, \quad \begin{bmatrix} u_1(0) \\ u_2(0) \\ u_3(0) \\ u_4(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix},$$

with a fixed step size of $h = 0.1$.

Results of the numerical experiment is as shown in Figure 3.6 and Table 5. A closer look at Table 5 reveals that there is no huge disparity in the `cputime` between the new method and those of Akinola et al [2]. But just as in the previous example, we noticed that the condition number of the system in Akinola et al [2], is four times that of the present work (217309 against 54214). This might have accounted for the slight difference between the last few digits for all the u values for both $x = 5$ and $x = 50$. However, aside the better condition number of the new method, there is not much significant differences in the results. Figure 3.6 also reveals that aside between $x \in [0.1, 0.2]$ for both u_3 and u_4 the new method performed favourably with the exact in the absence of round off errors.

Example 3.7. The Fatunla [50] problem is well known to be stiff

$$\begin{bmatrix} u_1'(x) \\ u_2'(x) \\ u_3'(x) \\ u_4'(x) \\ u_5'(x) \\ u_6'(x) \end{bmatrix} = \begin{bmatrix} -10u_1(x) + 100u_2(x) \\ -100u_1(x) - 10u_2(x) \\ -4u_3(x) \\ -u_4(x) \\ -0.5u_5(x) \\ -0.1u_6(x) \end{bmatrix}, \quad \begin{bmatrix} u_1(0) \\ u_2(0) \\ u_3(0) \\ u_4(0) \\ u_5(0) \\ u_6(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

TABLE 4. Absolute errors of our method with [2] on Example 3.4.

x	u_i	Akinola et al [2]	CPU Time(s)	$\kappa(G)$	New Method	CPU Time(s)	$\kappa(G)$
5	u_1	4.84×10^{-07}	7.23×10^{-5}	92309	4.84×10^{-07}	6.96×10^{-5}	22863
	u_2	5.28×10^{-08}			5.28×10^{-08}		
50	u_1	3.96×10^{-46}	8.66×10^{-5}	92296	3.96×10^{-46}	8.56×10^{-5}	22860
	u_2	1.50×10^{-27}			1.50×10^{-27}		

TABLE 5. Absolute errors of our method with Akinola et al [2], on Example 3.6.

x	u_i	Akinola et al [2]	CPU Time(s)	$\kappa(G)$	New Method	CPU Time(s)	$\kappa(G)$
5	u_1	4.88×10^{-15}	1.05×10^{-4}	217309	4.94×10^{-15}	1.05×10^{-4}	54214
	u_2	2.47×10^{-25}			2.47×10^{-25}		
	u_3	0			0		
	u_4	0			0		
50	u_1	1.54×10^{-33}	1.06×10^{-4}	217309	1.41×10^{-33}	1.16×10^{-4}	54214
	u_2	0			0		
	u_3	0			0		
	u_4	0			0		

From Table 6, it is obvious that the explanations given in the previous examples holds sway except the fact that there is a clear difference in the `cputimes` albeit the new method had a slightly higher cputime than that of Akinola et al [2]. Nevertheless, there is no difference in accuracy.

TABLE 6. Absolute errors of our method with [2] on Example 3.7.

x	u_i	Akinola et al [2]	CPU Time(s)	$\kappa(G)$	New Method	CPU Time(s)	$\kappa(G)$
5	u_1	2.60×10^{-22}	1.59×10^{-3}	19539	2.60×10^{-22}	1.68×10^{-3}	4865
	u_2	8.02×10^{-23}			8.02×10^{-23}		
	u_3	8.64×10^{-16}			8.64×10^{-16}		
	u_4	4.88×10^{-15}			4.94×10^{-15}		
50	u_1	0	1.68×10^{-3}	19539	0	1.73×10^{-3}	4865
	u_2	0			0		
	u_3	0			0		
	u_4	1.39×10^{-033}			1.41×10^{-33}		

3.1. **Preconditioning Strategies.** As shown in the previous section, we encountered ill–conditioning in the last three numerical experiments considered. In this section, we used an LU–type preconditioner to differentiate between the performance of the two methods under discussion in this paper. All plots were on a semilogy scale and in some selected numerical examples in the last section, the new method had the smallest norm of residual. Next, we start by presenting how to implement QMR based algorithm below.

3.1.1. *Implementation and Algorithm.* We find the LU factorization of the matrix $\mathbf{G}$ as the preconditioner such that $M = LU(\mathbf{G})$ without filling [25], where L and U are Lower-Upper triangular matrices. We find the LU factorization of $\mathbf{G}$ [16], that is $M = LU(\mathbf{G}) = M_1M_2$, where $M_1 = L$ and $M_2 = U$. Since M_1 and M_2 are triangular matrices, M is easy to invert because it is the product of two triangular matrices. We solve the system

$$\mathbf{G}\Delta\mathbf{y} = -\mathbf{F},$$

by replacing it with the preconditioned system

$$M_1^{-1}\mathbf{G}M_2^{-1}M_2\Delta\mathbf{y} = -M_1^{-1}\mathbf{F}.$$

Hence, we let $\hat{\mathbf{G}} = M_1^{-1}\mathbf{G}M_2^{-1}$, $\Delta\hat{\mathbf{y}} = M_2\Delta\mathbf{y}$ and $\hat{\mathbf{F}} = -M_1^{-1}\mathbf{F}$ and solve for $\Delta\hat{\mathbf{y}}$ in

$$\hat{\mathbf{G}}\Delta\hat{\mathbf{y}} = \hat{\mathbf{F}},$$

using the Quasi-Minimum-Residual (QMR) iterative method [17, 18, 19]. Therefore, we solve

$$M_2\Delta\mathbf{y} = \Delta\hat{\mathbf{y}},$$

to obtain $\Delta\mathbf{y}$. This now leads to the following corresponding algorithm.

Algorithm 3.1. **Input:**Same as Algorithm 2.1 above.
For $k = 0, 1, 2, \cdots,$

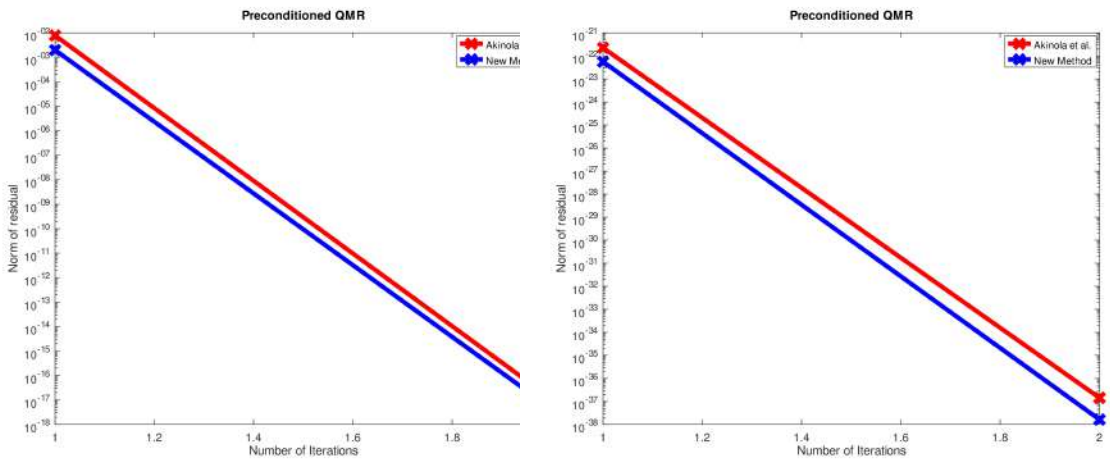
(1) Form $\mathbf{F}(\mathbf{y}^{(k)})$.
(2) Find the LU factorization of $[M1, M2] = \mathtt{LU}(\mathbf{G}(\mathbf{y}^{(k)}))$.
(3) Solve $\mathbf{G}(\mathbf{y}^{(k)})\Delta\mathbf{y}^{(k)} = \mathbf{F}(\mathbf{y}^{(k)})$, for $\Delta\mathbf{y}^{(k)}$ using QMR, that is

$$[\Delta\mathbf{y}^{(k)}, \mathtt{FLAG,\ RELRES,\ ITER,\ RESVEC}] = \mathtt{qmr}(\mathbf{G}(\mathbf{y}^{(k)}), -\mathbf{F}(\mathbf{y}^{(k)}), [\ ], [\ ], M1, M2);$$

(4) Apply Newton update $\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \Delta\mathbf{y}^{(k)}$.
(5) Continue until convergence.

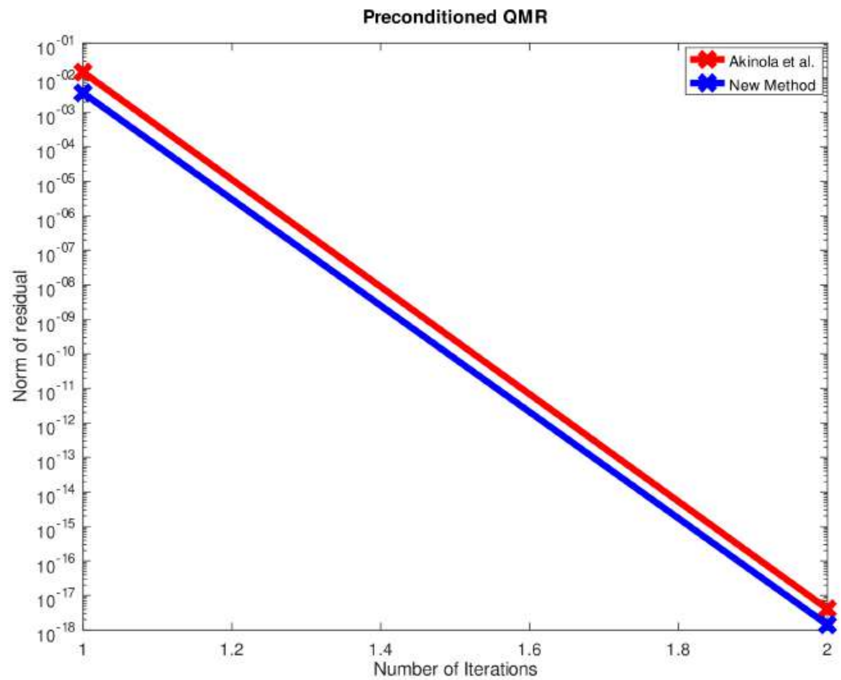
Output: y_{k+1}.

In each of the following Figures 4, 5, 6 and 7 for both $x = 5$ as well as $x = 50$, we compared the residual norm versus number of iterations upon an application of the preconditioned QMR algorithm on the four numerical examples of the previous section. With a default tolerance of 10^{-6} the results of numerical examples showed that QMR converged after two iterations with the residual norm in the present work less than that of Akinola et al [2], which was our aim. Hence, the new method gives better results asides being well conditioned without preconditioning than an earlier work in the literature. Therefore, we recommend the new method for the numerical integration of differential equations.

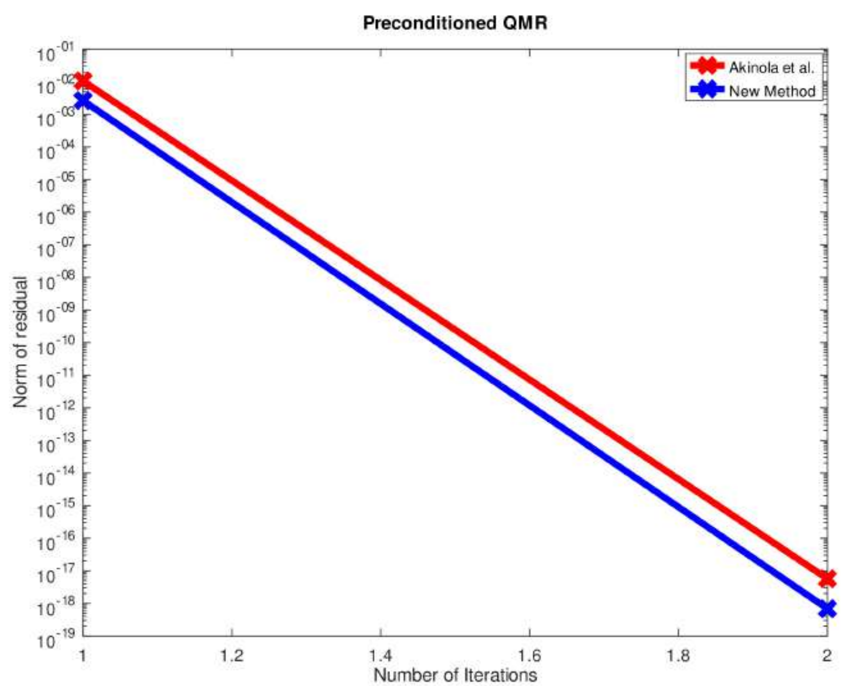


(A) $x = 5$. (B) $x = 50$.

FIGURE 4. Residual norm versus number of iterations of the block hybrid method of Akinola et al [2], with those in the present work for $x = 5$ and $x = 50$ on Example 3.4.

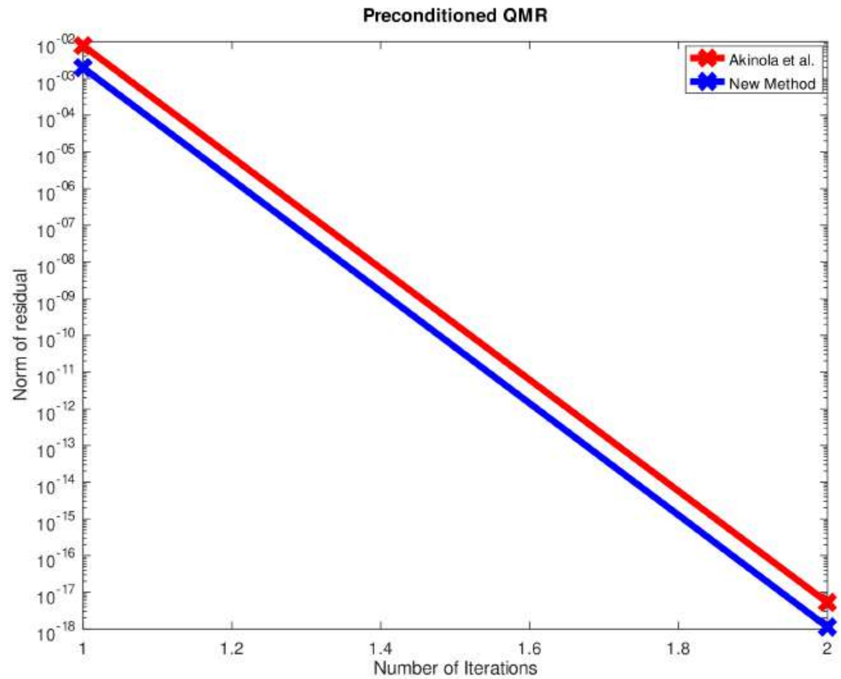


(A) $x = 5$

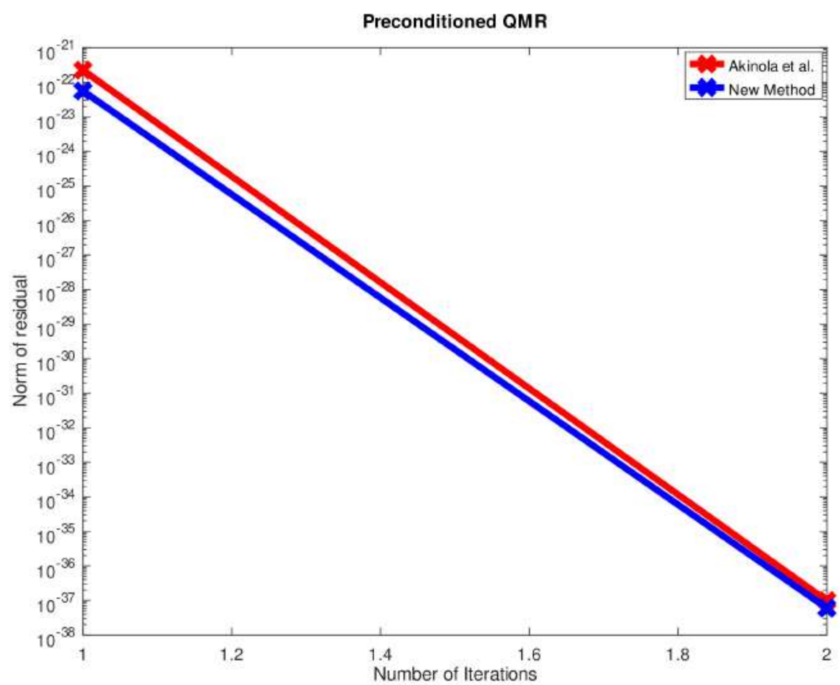


(B) $x = 50$

FIGURE 5. Residual norm versus number of iterations of the block hybrid method of Akinola et al [2], with those in the present work for $x = 5$ and $x = 50$ on Example 3.5.

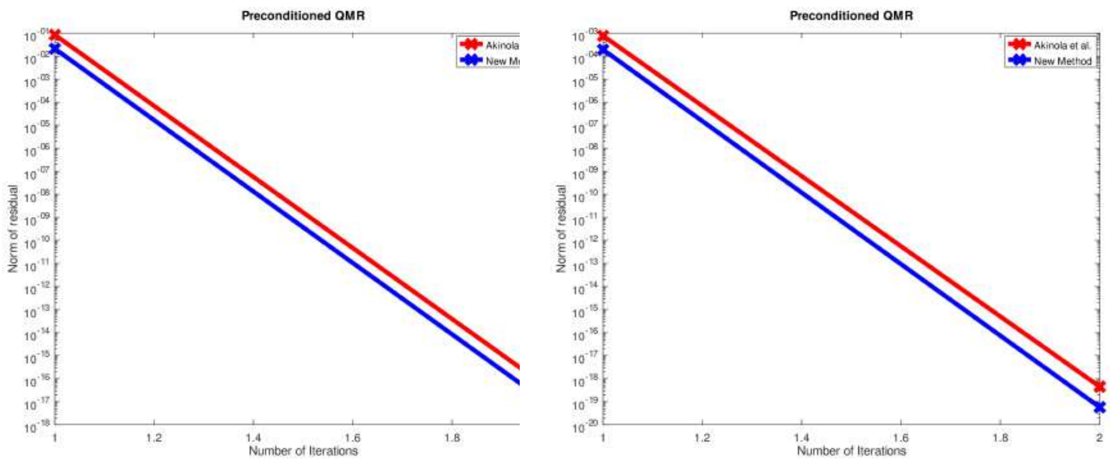


(A) $x = 5$



(B) $x = 50$

FIGURE 6. Residual norm versus number of iterations of the block hybrid method of Akinola et al [2], with those in the present work for $x = 5$ and $x = 50$ on Example 3.6.



(A) $x = 5$ (B) $x = 50$

FIGURE 7. Residual norm versus number of iterations of the block hybrid method of Akinola et al [2], with those in the present work for $x = 5$ and $x = 50$ on Example 3.7.

Table 7 shows the gains made at reducing the condition numbers in both methods using an LU–type preconditioned QMR.

TABLE 7. Condition number of the preconditioned system.

Example	Size of $\hat{G}$	nnz$(\hat{G})$	$\kappa(\hat{G})$
3.4	16×16	230	1.000
3.5	24×24	402	1.000
3.6	32×32	232	1.000
3.7	48×48	462	1.000

4. CONCLUSION

We developed a new derivative–free, zero stable, convergent block hybrid linear multistep method for the numerical approximation of stiff and non–stiff IVPs. It was observed that the new method gave favourable results that the works of [43], [45] and [46] in terms of maximum absolute error. We also showed computationally that the new method when compared to another existing ninth–order block hybrid method gives the same result with better condition numbers. In addition, applying an LU–type preconditioner using a Quasi-Minimal Residual iterative method showed that the new method gives a smaller residual norm than the existing method though both converged to the default tolerance with the same number of iterations.

Acknowledgment. The authors acknowledge the comments of anonymous referees whose comments helped improve this manuscript. We also acknowledge TETFund for providing funds for the publication of this article through the TETFund Institutional Based Research IBR07/038. We are equally indebted to the Office of Research and Development of the University of Jos, Jos, Nigeria.

Authors Contributions. All authors contributed equally to this research.

Authors' Conflicts of interest. The authors declare no conflict of interest.

Funding Statement. . This research received financial support from the Tertiary Education Trust fund (TETFund) in the form of a travel grant to the first author to present this paper at the 9^th International Conference on Advanced Computational Methods in ENgineering and Applied Mathematics (ACOMEN2025), University of Ghent, Belgium from September, 15–19, 2025.

REFERENCES

[1] L. F. Shampine. *Ill-conditioned matrices and the integration of stiff ODEs*. Journal of Computational and Applied Mathematics. (1993), 48, 279–292.
[2] R. O. Akinola, E. O. Omole, J. Sunday, S. Y. Kutchin. *A Ninth-Order First Derivative Method for Numerical Integration*. J. Nig. Soc. Phys. Sci. 7 (2025). DOI:10.46481/jnsps.2025.2028.
[3] S. O. Adee, V. O. Atabo. *Improved two-Point Block Backward Differentiation Formulae for Solving First Order Stiff Initial Value Problems of Ordinary Differential Equations*. Nigerian Annals of Pure and Applied Sciences, Vol 3 No 2. (June 2020), pp. 200–209.
[4] T. Aboiyar, T. Luga, B. V. Iyorter. *Derivation of Continuous Linear Multistep Methods Using Hermite Polynomials as Basis Functions.* American Journal of Applied Mathematics and Statistics. (2015). 3(6):220–225.
[5] A. U. Fotta, T. J. Alabi, B. Abdulqadir. *Block Method with One Hybrid Point for the Solution of First Order Initial Value Problems of Ordinary Differential Equations*. International Journal of Pure and Applied Mathematics, Vol. 103 No. 3, 511521.
[6] S. O. Ayinde, E. A. Ibijola. *A New Numerical Method for Solving First Order Differential Equations*. American Journal of Applied Mathematics and Statistics, Vol. 3, No. 4, 156160. DOI: 10.12691/ajams-3-4-4
[7] P. Oliver. *A Family of Linear Multistep Methods for the Solution of Stiff and Non-stiff ODEs*. IMA Journal of Numerical Analysis, Volume 2, Issue 3, (July 1982), Pages 289–301.
[8] Q. A. Huang, W. Jiang, J. Z. Yang et al. Linear multi-step methods and their numerical stability for solving gradient flow equations. Adv Comput Math 49, 39 (2023). https://doi.org/10.1007/s10444-023-10043-1.
[9] C. Y. Ishola, O. A. Oyelami, C. C. Elebor. A web-based tool for generating block linear multistep methods. Journal of NAMP 70, (2025) 167-178.
[10] S. A. Odejide, A. O. Adeniran. *A Hybrid Linear Collocation Multistep Scheme for Solving First Order Initial Value Problems.* Journal of the Nigerian Mathematical Society. Vol. (2012). 31 : 229–241.
[11] L. N. Trefethen, D. Bau III. *Numerical Linear Algebra*. SIAM. (1997).
[12] G. H. Golub, J. H. Wilkinson. *Ill-conditioned eigensystems and the com- putation of the Jordan canonical form*. SIAM review. (1976). 18 (4), 578–619.
[13] G. Peters, J. H. Wilkinson. *Inverse iteration, ill–conditioned equations and Newtons method*. SIAM review. (1979). 21(3), 339–360.
[14] M. Farooq, A. Salhi. *Improving the solvability of Ill–conditioned systems of linear equations by reducing the condition number of their matrices*. Journal of the Korean Mathematical Society, 2011, 48(48), 939–952.
[15] C. C. Douglas, L. Lee, M. Yeung. *On Solving Ill Conditioned Linear Systems*. Procedia Computer Science. (2016). 80, pp. 941–950.
[16] G. H. Golub, C. F. Van Loan. *Matrix calculations.* p. 548. (1999).

[17] R. W. Freund and N. M. Nachtigal. *QMR: a Quasi-Minimal Residual method for non-Hermitian linear systems*. Numer. Math., 60 (1991) pp. 315–339.
[18] R. W. Freund. *Transpose-Free Quasi-Minimal Residual Methods for Non-Hermitian Linear Systems*. SIAM J. Sci. Stat. Comput.
[19] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhour, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the solution of linear systems: Building blocks for iterative methods.* SIAM, 2nd ed., 1994.
[20] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Minneapolis: University of Minnesota MN. p. 567. (2003).
[21] Y. Saad. ILUT: *A dual threshold incomplete ILU factorization.* Numer Linear Algebra Appl. 1 739–748, 1994.
[22] M. Balandin, E. P. Shurina. *Methods for solving SLAE of large dimension*. Novosibirsk: NSTU. p. 70. (2000).
[23] S. E. Saukh. *Incomplete column-row factorization of matrices for the iterative solution of large systems of equations*. Electronic Modeling 32(6) 3–14. (2010).
[24] T. Y. Chen. *Preconditioning Sparse Matrices for Computing Eigenvalues and Solving Linear Systems of Equations*. University of California at Berkeley. p. 135, (2011).
[25] S. Y. Gogoleva. *Preconditioning Based on LU Factorization in Iterative Method for Solving Systems of Linear Algebraic Equations with Sparse Matrices.*
[26] J. Demmel. *Computational Linear Algebra. Theory and applications.* Moscow: Mir. p. 430, (2001).
[27] D. G. Yakubu, A. Shokri, G. M. Kumleng, D. Marian. *Second Derivative Block Hybrid Methods for the the numerical integration of Differential systems*. Fractal Fract. 2022, 6, 386. https://doi.org/10.3390/fractalfract6070386.
[28] E. O. Omole, L. A. Ukpebor. *A Step by Step Guide on Derivation and Analysis of a New Numerical Method for Solving Fourth-order Ordinary Differential Equations*. Mathematics Letters. Vol. 6, No. 2, (2020), pp. 13–31. doi: 10.11648/j.ml.20200602.12
[29] W. B. Gragg, H. J. Stetter. *Generalized multistep predictorcorrector methods*. J. ACM (1964), 11, 188–209.
[30] J. C. Butcher. *A multistep generalization of RungeKutta method with four or five stages*. J. Ass. Comput. Mach. (1967), 14, 8499.
[31] W. C. Gear. *Hybrid multistep method for initial value in ordinary differential equations*. J. SIAM Numer. Anal. (1965), 2, 69–86.
[32] M. Urabe. *An implicit onestep method of highorder accuracy for the numerical integration of ordinary differential equations*. J. Numer. Math. 1970, 15, 151–164.
[33] T. A. Mitsui. *Modified version of Urabes implicit singlestep method*. J. Comp. Appli. Math. 1987, 20, 325–332.
[34] R. O. Akinola, K. J. Ajibade. *A Proof of the Non-Singularity of the D Matrix Used in Deriving the two–Step Butchers Hybrid Scheme for the Solution of Initial Value Problems*. Journal of Applied Mathematics and Physics, 9, 3177–3201. (2021).
[35] R. O. Akinola. *An accurate implementation of the two-step Butcher's Hybrid Scheme on Initial Value Problems*. BSc Thesis, University of Jos, Jos, Nigeria. (Unpublished) (2001).
[36] P. Onumanyi, D. O. Awoyemi, S. N. Jator, U. W. Sirisena (1994). *New Linear Multistep Methods with Continuous Coefficients for First Order Initial Value Problems*. Journal of Nigeria Mathematics Society, 13 : 37-51.
[37] R. O. Akinola, A. Shokri, S. W. Yao, S. Y. Kutchin. *Circumventing Ill–Conditioning Arising from Using Linear Multistep Methods In Approximating the Solution of Initial Value Problems*. Mathematics 2022, 10, 2910. https://doi.org/10.3390/math10162910
[38] R. O. Akinola, A. S. Akoh. *A Seventh–Order Computational Algorithm for the Solution of Stiff Systems of Differential Equations.* Journal of the Nigerian Mathematical Society. Vol. 42, Issue 3, pp. 215 - 242. (2023).
[39] P. Henrici. *Discrete Variable Methods in Ordinary Differential Equations*. John Wiley, New York, p. 407. (1962).

[40] J. D. Lambert. *Computational Methods in Ordinary Differential Equations*. John Wiley, New York. (1973).

[41] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations.* John Wiley and Sons Ltd. (2016).

[42] J. C. Butcher. *Numerical methods for ordinary differential equations in the 20th century.* Journal of Computational and Applied Mathematics, Volume 125, Issues 1-2, (2000), Pages 1-29

[43] S. D. Yakubu, P. Sibanda. *One–Step Family of Three Optimized Second–Derivative Hybrid Block Methods for Solving First–Order Stiff Problems.* Hindawi Journal of Applied Mathematics (2024), Article ID 5078943, https://doi.org/10.1155/2024/5078943

[44] O. A. Akinfenwa, R. I. Abdulganiy, B. I. Akinnukawe, S. A. Okunuga. *Seventh order hybrid block method for solution of rst order sti systems of initial value problems.* Journal of the Egyptian Mathematical Society, vol. 28, no. 1, p. 28, 2020.

[45] S. E. Ogunfeyitimi, M. N. O. Ikhile. *Generalized second derivative linear multistep methods based on the methods of enright.* International Journal of Applied and Computational Mathematics, vol. 6, no. 3, p. 76, 2020.

[46] S. E. Ogunfeyitimi, M. N. O. Ikhile. *Multi-block generalized Adams-type integration methods for dierential algebraic equations.* International Journal of Applied and Computational Mathematics, vol. 7, no. 5, p. 197, 2021.

[47] P. Kaps, S. W. H. Poon, T. D. Bui. *Rosenbrock methods for Stiff ODEs: A comparison of Richardson extrapolation and embedding technique*. Computing 34 (1) (1985) 17–40.

[48] W. C. Gear. *DIFSUB for Solution of ordinary differential equations*. Comm. ACM 1971, 14, 185190.

[49] W. H. Enright. *Second derivative multistep methods for stiff ordinary differential equations*. SIAM J. Numer. Anal. 1974, 11, 321– 341.

[50] S. O. Fatunla. *Numerical integrators for stiff and highly oscillatory differential equations*. J. Math. Comput. 1980, 34, 373–390.

Appendix A. Continuous Coefficients

In this Appendix we present the continuous coefficients. The continuous coefficients are the elements of the first row of the inverse of **D**.

$$\alpha_0(w) = 1,$$

$$\beta_0(w) = -\frac{160w^9 + 2520hw^8 + 16560h^2w^7 + 58800h^3w^6 + 121842h^4w^5}{2041200h^8} - \frac{147735h^5w^4 + 98010h^6w^3 + 28350h^7w^2 - 473977h^9}{2041200h^8},$$

$$\beta_1(w) = \frac{320w^9 + 4680hw^8 + 27360h^2w^7 + 78960h^3w^6 + 102564h^4w^5}{113400h^8} - \frac{9135h^5w^4 + 197940h^6w^3 + 237330h^7w^2 + 113400h^8w - 343921h^9}{113400h^8},$$

$$\begin{aligned}
\beta_{\frac{3}{2}}(w) &= -\frac{1120w^9 + 15750hw^8 + 86760h^2w^7 + 225750h^3w^6 + 223524h^4w^5}{85050h^8} \\
&\quad - \frac{171675h^5w^4 + 578340h^6w^3 + 396900h^7w^2 - 594011h^9}{85050h^8}, \\
\beta_2(w) &= \frac{560w^9 + 7560hw^8 + 39240h^2w^7 + 92400h^3w^6 + 68607h^4w^5}{18900h^8} \\
&\quad - \frac{101745h^5w^4 + 210735h^6w^3 + 99225h^7w^2 - 203338h^9}{18900h^8}, \\
\beta_{\frac{5}{2}}(w) &= -\frac{1120w^9 + 14490hw^8 + 70920h^2w^7 + 152250h^3w^6 + 84924h^4w^5}{28350h^8} \\
&\quad - \frac{191205h^5w^4 + 310380h^6w^3 + 132300h^7w^2 - 310181h^9}{28350h^8}, \\
\beta_3(w) &= \frac{2240w^9 + 27720hw^8 + 128160h^2w^7 + 253680h^3w^6 + 109116h^4w^5}{68040h^8} \\
&\quad - \frac{336735h^5w^4 + 487620h^6w^3 + 198450h^7w^2 - 501889h^9}{68040h^8}, \\
\beta_{\frac{7}{2}}(w) &= -\frac{160w^9 + 1890hw^8 + 8280h^2w^7 + 15330h^3w^6 + 5292h^4w^5}{9450h^8} \\
&\quad - \frac{21105h^5w^4 + 28620h^6w^3 + 11340h^7w^2 - 30113h^9}{9450h^8}, \\
\beta_4(w) &= \frac{1120w^9 + 12600hw^8 + 52560h^2w^7 + 92400h^3w^6 + 26334h^4w^5}{226800h^8} \\
&\quad - \frac{130725h^5w^4 + 169890h^6w^3 + 66150h^7w^2 - 181751h^9}{226800h^8}, \\
\beta_{\frac{9}{2}}(w) &= -\frac{160w^9 + 1710hw^8 + 6840h^2w^7 + 11550h^3w^6 + 2772h^4w^5}{255150h^8} \\
&\quad - \frac{16695h^5w^4 + 21060h^6w^3 + 8100h^7w^2 - 22823h^9}{255150h^8}.
\end{aligned}$$

RICHARD OLATOKUNBO AKINOLA*
DEPARTMENT OF MATHEMATICS, FACULTY OF NATURAL SCIENCES, UNIVERSITY OF JOS, JOS, PLATEAU STATE, NIGERIA.
E-mail address: roakinola@gmail.com/akinolar@unijos.edu.ng

JOSHUA SUNDAY
DEPARTMENT OF MATHEMATICS, FACULTY OF NATURAL SCIENCES, UNIVERSITY OF JOS, JOS, PLATEAU STATE, NIGERIA
E-mail address: joshuasunday2000@gmail.com

EZEKIEL O. OMOLE
LANDMARK UNIVERSITY, DEPARTMENT OF PHYSICAL SCIENCES, MATHEMATICS PROGRAMME, OMU–ARAN, KWARA STATE, NIGERIA
E-mail address: omolez247@gmail.com

ELEOJO RACHEL AKOR
DEPARTMENT OF MATHEMATICS, FACULTY OF NATURAL SCIENCES, UNIVERSITY OF JOS, JOS, PLATEAU STATE, NIGERIA
E-mail address: eleojorachelakor@gmail.com