Bima Journal of Science and Technology, Vol. 8(3B) Oct, 2024 ISSN: 2536-6041

DOI: 10.56892/bima.v8i3B.842

An Ideal Symmetric Predictor-Corrector Approach for Solving Second-Order Ordinary Differential Equations

Friday Oghenerukevwe Obarhua

Department of Mathematical Sciences, The Federal University of Technology, P. M. B. 704, Akure, Nigeria.

Corresponding Author: obarhuafo@futa.edu.ng

ABSTRACT

The second order initial value problems are generally integrated numerically by development of numerical methods designed specifically for second order differential equations. In this work, a symmetric corrector method is presented capable to integrate real-life problems modeled into second order ordinary differential equations directly. The power series is used as a basis function which was interpolated and collocation was done at its second derivative of only grid points. This is to ensure that the hybrid points are at the $y-$ $function$. The resulting systems of equations is solved and after necessary simplifications continuous method is obtained. Attempts were also made to develop predictors of the same order with the method to circumvent the inherent demerit of predictor methods. The method is of optimal order, symmetric, consistent, and zero-stable. The discrete method obtained is applied to solve real-life problems with better performance.

Keywords: Symmetric, Optimal method, Differential system, Predictor-Corrector method.

INTRODUCTION

In mathematics, mathematical modeling is a key tool for the analysis of a wide range of real-world problems ranging from physics and engineering to chemistry, biology and even economics using differential equations Hritonenko and Yatsenko (1999). Many of the principles, or laws, underlying the behavior of the natural world are statements or relations involving rates at which things happen and these are expressed in differential equations.

This research article considered an ideal symmetric method for direct integration of general second order with initial and boundary conditions of ordinary differential equations (ODEs) of the type;

General second order IVP

$$y'' = f\left(t, y, y'\right); \; y(t_0) = y_0, y'(t_0) = y_1 \quad (1)$$

Special second-order IVP

$$y'' = f\left(t, y\right); \; y(t_0) = y_0, y'(t_0) = y_1 \quad (2)$$

General second order BVP with Dirichlet boundary conditions

$$y'' = f\left(t, y, y'\right); \; y(t_0) = y_0, y(\varpi) = y_1 \quad (3)$$

General second order BVP with Newmann boundary conditions

$$y'' = f\left(t, y, y'\right); \; y'(t_0) = y_0, y(\varpi) = y_1 \quad (4)$$

Second-order system of equations

$$y_1'' = f_1\left(t, y_1, y_1'\right); \; y_1(t_0) = y_0, y_1'(t_0) = y_1$$
$$y_2'' = f_2\left(t, y_2, y_2'\right); \; y_2(t_0) = y_2, y_2'(t_0) = y_3 \quad (5)$$

It is well known that these problems (1–5), particularly the non-linear ones, may or may not be solved in a closed form. Even in those cases, the problems are typically reduced to systems of first order equations, which can be solved using any equivalent methods. Numerous authors have discussed these types of methods, including Olabode and Momoh (2016), Areo and Adeniyi (2013), and Kayode and Obarhua (2013), to name a few

Despite the traditional approach's triumphs, many authors were drawn to its problems, which led them to examine the direct method as an alternative to decreasing order. Obarhua and Kayode (2016), Kayode and Adeyeye (2011, 2013), and Ramos and Rufai (2019). A two-step two-point implicit hybrid predictor-corector method was presented by Kayode and Adeyeye (2013) to solve (1) directly. Ogunfeyitimi and Ikhile (2019) presented second derivative generalized extended backward difference formula to solve stiff order boundary value problems. A generalized cash-type second derivative extended backward differential formula was introduced by Okor *et al.* (2022) as a boundary value approach for the stiff system of type (4). Omole *et al.* (2023) have also proposed an algebraic order nine approach for solving second order boundary and initial value problems.

Therefore, the methods that these authors have proposed are unable to directly solve problems (1–5) on their own, and furthermore, their order and accuracy are insufficient to handle these complex problems. Nevertheless, these authors have also introduced hybrid points at both $f-function$ and $y-function$, which raises the function evaluations in the $f-function$ computational effort and, as a result, lowers the accuracy of their methods, Kayode and Adeyeye (2011).

Therefore, this study was motivated by the setbacks caused by the increase in function evaluations, reduced order of accuracy. The interest of this study is to develop method with hybrid points at $y-function$ only for directly solving (1–5) which is more effective and accurate in performance by effecting where points in the power series used as basis function as the approximate solution. The hybrid points' position is capable of reducing function evaluation and increase order of accuracy. Additionally, the proposed method is made to ensure that the starting methods and the corrector (primary) method have the same order of accuracy.

Derivation of the Method

This section shows the derivation of a continuous symmetric implicit two-point hybrid method for the solution of problems (1-5). Consider the equally spaced points on the integration interval given by

$$a = x_0 < x_1 < \cdots < x_{N-1} < x_N = b\,, \qquad (6)$$

with a specified positive integer step size given by $h = x_{n+1} - x_n,\ n = 1, \cdots, N;\ N = \dfrac{b-a}{h}$.

Assuming the power series as the approximate solution

$$y(x) = \sum_{j=0}^{2(k+1)} a_j x^j\,. \qquad \mathbf{(7)}$$

The second derivative of (7) as compared with (1) gives

$$f\left(x,\ y,\ y'\right) = \sum_{j=2}^{2(k+1)} \left(j(j-1) a_j x^{j-2} \right). \qquad (8)$$

Equation (7) is interpolated at the grid and off-grid points $x = x_{n+i}, i = 0, \lambda, 1, \tau$ where $0 < \lambda < 1$ and $1 < \tau < 2$. Equation (8) was collocated at three grid points $x = x_{n+i},\ i = 0, 1, 2$ which resulted and expressed inform of matrix equation given below;

$$AB = C\,, \qquad (9)$$

DOI: 10.56892/bima.v8i3B.842

$$A=\begin{pmatrix} 0 & 0 & 2 & 6x_n & 12x_n^2 & 20x_n^3 & 30x_n^4 \\ 0 & 0 & 2 & 6x_{n+1} & 12x_{n+1}^2 & 20x_{n+1}^3 & 30x_{n+1}^4 \\ 0 & 0 & 2 & 6x_{n+2} & 12x_{n+2}^2 & 20x_{n+2}^3 & 30x_{n+2}^4 \\ 1 & x_n & x_n^2 & x_n^3 & x_n^4 & x_n^5 & x_n^6 \\ 1 & x_{n+r} & x_{n+\lambda}^2 & x_{n+\lambda}^3 & x_{n+\lambda}^4 & x_{n+\lambda}^5 & x_{n+\lambda}^6 \\ 1 & x_{n+1} & x_{n+1}^2 & x_{n+1}^3 & x_{n+1}^4 & x_{n+1}^5 & x_{n+1}^6 \\ 1 & x_{n+\tau} & x_{n+\tau}^2 & x_{n+\tau}^3 & x_{n+\tau}^4 & x_{n+\tau}^5 & x_{n+\tau}^6 \end{pmatrix}, \quad B=\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{pmatrix}, \quad C=\begin{pmatrix} f_n \\ f_{n+1} \\ f_{n+2} \\ y_n \\ y_{n+\lambda} \\ y_{n+1} \\ y_{n+\tau} \end{pmatrix}.$$

The values of $a_j\,'s,\ \ j=0(1)6$ were determined, using Gaussian elimination method to be Substituted back into (7) and using the transformation in Obarhua (2019), $t=\frac{1}{h}\left(x-x_{n+k-1}\right), \frac{dt}{dx}=\frac{1}{h}$ and simplifying the result gives a continuous hybrid linear multistep method given as;

$$y_k(x)=\sum_{j=0}^{k-1}\alpha_j(x)y_{n+j}+\{\partial_1(x)y_{n+\lambda}+\partial_2(x)y_{n+\tau}\}+h^2\sum_{j=0}^{k}\beta_j(x)f_{n+j}, \tag{10}$$

with the coefficients $\alpha_j{}'s,\ \partial_i{}'s$ and $\beta_j{}'s$ as functions of t obtained to be

$$\alpha_0(t)=\frac{(2t-1)(\tau-2t)(\lambda-2t)\begin{pmatrix} t^3(48\lambda^2\tau+48\lambda\tau^2-192\lambda\tau+128+48\lambda^3+48\tau^3-192\lambda^2-192\tau^2 \\ +128\lambda+128\tau)+t^2(24\lambda^3\tau+24\lambda\tau^3-96\lambda^3-96\tau^3+364\lambda^2+364\tau^2 \\ -196\lambda-196\tau-192\lambda^2\tau-192\lambda\tau^2+428\lambda\tau+24\lambda^2\tau^2-260)+t(12\lambda^3\tau^2 \\ +12\lambda^2\tau^3-228\lambda\tau-96\lambda^2\tau^2-48\lambda^3\tau-48\lambda\tau^3+214\lambda\tau^2+214\lambda^2\tau-10\lambda \\ -10\tau+32\lambda^3+32\tau^3-98\lambda^2-98\tau^2+120)+(16\lambda^3+16\tau^3-65\lambda^2 \\ -65\tau^2-5\lambda\tau+6\lambda^3\tau^3+91\lambda^2\tau^2-24\lambda^2\tau^3-24\lambda^3\tau^2+60\lambda+60\tau-49\lambda^2\tau \\ -49\lambda\tau^2+16\lambda^3\tau+16\lambda\tau^3) \end{pmatrix}}{\begin{pmatrix} 60\lambda^2\tau+60\lambda\tau^2-49\lambda^3\tau^2-49\lambda^2\tau^3+16\lambda^4\tau^2+16\lambda^2\tau^4+91\lambda^3\tau^3-24\lambda^3\tau^4 \\ -24\lambda^4\tau^3+6\lambda^4\tau^4-5\lambda^2\tau^2-65\lambda^3\tau-65\lambda\tau^3+16\lambda^4\tau+16\lambda\tau^4 \end{pmatrix}}$$

$$\partial_1(t)=\frac{2t(2t-1)(\tau-2t)\begin{pmatrix} t^3(128+48\tau^3-192\tau^2+128\tau)+t^2(-260-96\tau^3+364\tau^2)+t(-10\tau+32\tau^3 \\ -98\tau^2+120)+(16\tau^3-65\tau^2+60\tau) \end{pmatrix}}{\lambda(\lambda-1)(\lambda-\tau)\begin{pmatrix} 60\lambda+60\tau-49\lambda^2\tau-49\lambda\tau^2+16\lambda^3\tau+16\lambda\tau^3+91\lambda^2\tau^2-24\lambda^2\tau^3 \\ -24\lambda^3\tau^2+6\lambda^3\tau^3-5\lambda\tau-65\lambda^2-65\tau^2+16\lambda^3+16\tau^3 \end{pmatrix}}$$

DOI: 10.56892/bima.v8i3B.842

$$\alpha_1(t)=\frac{2t(\tau-2t)(\lambda-2t)\begin{pmatrix}t^3(48\lambda^2\tau+48\lambda\tau^2-240\lambda\tau-240\tau^2-240\lambda^2+48\lambda^3+48\tau^3+320\lambda+320\tau)\\+t^2(24\lambda^3\tau+24\lambda\tau^3-240\lambda^2\tau-240\lambda\tau^2+24\lambda^2\tau^2+740\lambda\tau+580\lambda^2+580\tau^2\\-120\lambda^3-120\tau^3-720\lambda-720\tau)+t(12\lambda^3\tau^2+12\lambda^2\tau^3-720\lambda\tau-120\lambda^2\tau^2\\-60\lambda^3\tau-60\lambda\tau^3+370\lambda^2\tau+370\lambda\tau^2+400\lambda+400\tau+80\lambda^3+80\tau^3-360\lambda^2\\-360\tau^2)+(145\lambda^2\tau^2-30\lambda^2\tau^3-30\lambda^3\tau^2-180\lambda^2\tau-180\lambda\tau^2+40\lambda^3\tau+40\lambda\tau^3\\+200\lambda\tau+6\lambda^3\tau^3)\end{pmatrix}}{(\tau-1)(\lambda-1)\begin{pmatrix}60\lambda+60\tau-49\lambda^2\tau-49\lambda\tau^2+16\lambda^3\tau+16\lambda\tau^3+91\lambda^2\tau^2-24\lambda^2\tau^3\\-24\lambda^3\tau^2+6\lambda^3\tau^3-5\lambda\tau-65\lambda^2-65\tau^2+16\lambda^3+16\tau^3\end{pmatrix}}$$

$$\partial_2(t)=\frac{2t(2t-1)(\lambda-2t)\begin{pmatrix}t^3(128+48\lambda^3-192\lambda^2+128\lambda)+t^2(-260-96\lambda^3+364\lambda^2-196\lambda)\\+t(-10\lambda+32\lambda^3-98\lambda^2+120)+16\lambda^3-65\lambda^2+60\lambda\end{pmatrix}}{\tau(\tau-1)(\lambda-\tau)\begin{pmatrix}60\lambda+60\tau-49\lambda^2\tau-49\lambda\tau^2+16\lambda^3\tau+16\lambda\tau^3+91\lambda^2\tau^2-24\lambda^2\tau^3\\-24\lambda^3\tau^2+6\lambda^3\tau^3-5\lambda\tau-65\lambda^2-65\tau^2+16\lambda^3+16\tau^3\end{pmatrix}}$$

$$\beta_0(t)=\frac{t(2t-1)(\tau-2t)(\lambda-2t)\begin{pmatrix}t^2(-60\lambda^2\tau-60\lambda\tau^2+384+12\lambda^2\tau^2+292\lambda\tau+84\lambda^2+84\tau^2-368\lambda\\-368\tau)+t(-30\lambda^2\tau^2-678\lambda\tau+146\lambda^2\tau+146\lambda\tau^2+782\lambda+782\tau\\-184\lambda^2-184\tau^2-780)+(96\lambda^2+96\tau^2+391\lambda\tau+21\lambda^2\tau^2-92\lambda^2\tau\\-92\lambda\tau^2-390\lambda-390\tau+360)\end{pmatrix}}{6\begin{pmatrix}60\lambda+60\tau-49\lambda^2\tau-49\lambda\tau^2+16\lambda^3\tau+16\lambda\tau^3+91\lambda^2\tau^2-24\lambda^2\tau^3\\-24\lambda^3\tau^2+6\lambda^3\tau^3-5\lambda\tau-65\lambda^2-65\tau^2+16\lambda^3+16\tau^3\end{pmatrix}}$$

DOI: 10.56892/bima.v8i3B.842

$$\beta_1(t)=\frac{t(2t-1)(\tau-2t)(\lambda-2t)\begin{pmatrix}t^2(-36\lambda^2\tau-36\lambda\tau^2+12\lambda^2\tau^2+76\lambda\tau-36\lambda^2-36\tau^2+112\lambda+112\tau)\\+t(-18\lambda^2\tau^2-48\lambda\tau+38\lambda^2\tau+38\lambda\tau^2-160\lambda-160\tau+56\lambda^2+56\tau^2)\\+(-80\lambda\tau-9\lambda^2\tau^2+28\lambda^2\tau+28\lambda\tau^2)\end{pmatrix}}{3\begin{pmatrix}60\lambda+60\tau-49\lambda^2\tau-49\lambda\tau^2+16\lambda^3\tau+16\lambda\tau^3+91\lambda^2\tau^2-24\lambda^2\tau^3\\-24\lambda^3\tau^2+6\lambda^3\tau^3-5\lambda\tau-65\lambda^2-65\tau^2+16\lambda^3+16\tau^3\end{pmatrix}}$$

$$\beta_2(t)=\frac{t(2t-1)(\tau-2t)(\lambda-2t)\begin{pmatrix}t^2(-12\lambda^2\tau-12\lambda\tau^2+4\lambda\tau+12\lambda^2\tau^2-12\lambda^2-12\tau^2+16\lambda+16\tau)\\+t(-6\lambda^2\tau^2+6\lambda\tau+2\lambda^2\tau+2\lambda\tau^2-10\lambda-10\tau+8\lambda^2+8\tau^2)\\+(-5\lambda\tau-3\lambda^2\tau^2+4\lambda^2\tau+4\lambda\tau^2)\end{pmatrix}}{6\begin{pmatrix}60\lambda+60\tau-49\lambda^2\tau-49\lambda\tau^2+16\lambda^3\tau+16\lambda\tau^3+91\lambda^2\tau^2-24\lambda^2\tau^3\\-24\lambda^3\tau^2+6\lambda^3\tau^3-5\lambda\tau-65\lambda^2-65\tau^2+16\lambda^3+16\tau^3\end{pmatrix}} \tag{11}$$

The first derivatives of the coefficients are

$$\alpha_0'(t)=\frac{2\begin{pmatrix}t^5(1152\lambda^2\tau+1152\lambda\tau^2-4608\lambda\tau+1152\lambda^3+1152\tau^3-4608\lambda^2-4608\tau^2+3072\lambda+3072\tau\\+3072)+t^4(-480\lambda^3\tau-480\lambda\tau^3-480\lambda^2\tau^2-480\lambda^4-480\tau^4-480\lambda^3-480\tau^3+7920\lambda^2+\\7920\tau^2-6480\lambda-6480\tau-480\lambda^2\tau-480\lambda\tau^2+7920\lambda\tau-6480)+t^3(-2400\lambda^2\tau-2400\lambda\tau^2\\+960\lambda^2\tau^2+960\lambda^4+960\tau^4-2400\lambda^3-2400\tau^3-2400\lambda^2-2400\tau^2+4000\lambda+4000\tau+\\960\lambda^3\tau+960\lambda\tau^3-2400\lambda\tau+4000)+t^2(-720\lambda\tau-480\lambda^2\tau^2-720-480\lambda^4-480\tau^4+\\1680\lambda^3+1680\tau^3-720\lambda^2-720\tau^2-720\lambda-720\tau-480\lambda^3\tau-480\lambda\tau^3+1680\lambda^2\tau+\\1680\lambda\tau^2)+(6\lambda^4\tau^4-24\lambda^3\tau^4-24\lambda^4\tau^3+91\lambda^3\tau^3+16\lambda^2\tau^4+16\lambda^4\tau^2+11\lambda^2\tau^2-49\lambda^2\tau^3-\\49\lambda^3\tau^2+16\lambda^4\tau+16\lambda\tau^4+60\lambda\tau+60\lambda^2+60\tau^2-65\lambda^3-65\tau^3+16\lambda^4+16\tau^4)\end{pmatrix}}{\lambda\tau\begin{pmatrix}\{60(\lambda+\tau)-49(\lambda^2\tau+\lambda\tau^2)+16(\lambda^3\tau+\lambda\tau^3)-24(\lambda^2\tau^3+\lambda^3\tau^2)+16(\lambda^3+\tau^3)-65(\lambda^2+\tau^2)\\-5\lambda\tau+91\lambda^2\tau^2+6\lambda^3\tau^3\}\end{pmatrix}}$$

$$\partial_1'(t)=\frac{2\begin{pmatrix}t^5(-3072-1152\tau^3+4608\tau^2-3072\tau)+t^4(6480+480\tau^4+480\tau^3-7920\tau^2+6480\tau)\\+t^3(-4000-960\tau^4+2400\tau^3-4000\tau)+t^2(720+480\tau^4-1680\tau^3+720\tau^2+720\tau)\\+(-60\tau^2+65\tau^3-16\tau^4)\end{pmatrix}}{\begin{pmatrix}\lambda(\lambda-1)(\lambda-\tau)\{60(\lambda+\tau)-49(\lambda^2\tau+\lambda\tau^2)+16(\lambda^3\tau+\lambda\tau^3)-24(\lambda^2\tau^3+\lambda^3\tau^2)+16(\lambda^3+\tau^3)\\-65(\lambda^2+\tau^2)-5\lambda\tau+91\lambda^2\tau^2+6\lambda^3\tau^3\}\end{pmatrix}}$$

$$\alpha_1'(t)=\frac{2\begin{pmatrix} t^5(1152\lambda^2\tau+1152\lambda\tau^2-5760\lambda\tau+1152\lambda^3+1152\tau^3-5760\tau^2-5760\lambda^2+7680\lambda+7680\tau) \\ +t^4(-480\lambda^3\tau-480\lambda\tau^3+8400\lambda\tau-480\lambda^2\tau^2-480\lambda^4-480\tau^4+8400\lambda^2+8400\tau^2-14400\lambda \\ -14400\tau)+t^3(960\lambda^3\tau+960\lambda\tau^3-3360\lambda^2\tau-3360\lambda\tau^2+960\lambda^2\tau^2+960\lambda^4+960\tau^4-3360\lambda^3 \\ -3360\tau^3+6400\lambda+6400\tau)+t^2(-480\lambda^3\tau-480\lambda\tau^3+2160\lambda^2\tau+2160\lambda\tau^2-2400\lambda\tau-480 \\ \lambda^2\tau^2-480\lambda^4-480\tau^4+2160\lambda^3+2160\tau^3-2400\lambda^2-2400\tau^2)+(6\lambda^4\tau^4-30\lambda^4\tau^3-30\lambda^3\tau^4 \\ +145\lambda^3\tau^3+40\lambda^2\tau^4+40\lambda^4\tau^2+200\lambda^2\tau^2-180\lambda^3\tau^2-180\lambda^2\tau^3) \end{pmatrix}}{\begin{pmatrix} (\tau-1)(\lambda-1)\{60(\lambda+\tau)-49(\lambda^2\tau+\lambda\tau^2)+16(\lambda^3\tau+\lambda\tau^3)-24(\lambda^2\tau^3+\lambda^3\tau^2)+16(\lambda^3+\tau^3) \\ -65(\lambda^2+\tau^2)-5\lambda\tau+91\lambda^2\tau^2+6\lambda^3\tau^3) \end{pmatrix}}$$

$$\partial_2'(t)=\frac{2\begin{pmatrix} t^5(-3072-1152\lambda^3+4608\lambda^2-3072\lambda)+t^4(6480+480\lambda^4+480\lambda^3-7920\lambda^2+6480\lambda) \\ +t^3(-4000-960\lambda^4+2400\lambda^3+2400\lambda^2-4000\lambda)+t^2(720+480\lambda^4-1680\lambda^3+720\lambda^2 \\ +720\lambda)+(-60\lambda^2+65\lambda^3-16\lambda^4) \end{pmatrix}}{\begin{pmatrix} \tau(\tau-1)(\lambda-\tau)\{60(\lambda+\tau)-49(\lambda^2\tau+\lambda\tau^2)+16(\lambda^3\tau+\lambda\tau^3)-24(\lambda^2\tau^3+\lambda^3\tau^2) \\ +16(\lambda^3+\tau^3)-65(\lambda^2+\tau^2)-5\lambda\tau+91\lambda^2\tau^2+6\lambda^3\tau^3\} \end{pmatrix}}$$

$$\beta_0'(t)=\frac{\begin{pmatrix} t^5(-2880\lambda^2\tau-2880\lambda\tau^2+14016\lambda\tau+576\lambda^2\tau^2+4032\tau^2+4032\lambda^2-17664\lambda-17664\tau+18432) \\ +t^4(1200\lambda^3\tau+1200\lambda\tau^3-480\lambda^2\tau-480\lambda\tau^2-18240\lambda\tau-240\lambda^3\tau^2-240\lambda^2\tau^3+960\lambda^2\tau^2- \\ 1680\lambda^2-1680\tau^2-1680\lambda^3-1680\tau^3+30960\lambda+30960\tau-38880)+t^3(-2144\lambda^3\tau-2144\lambda\tau^3+ \\ 8576\lambda^2\tau+8576\lambda\tau^2-4480\lambda\tau+96\lambda^3\tau^3+96\lambda^2\tau^3+96\lambda^3\tau^2-2144\lambda^2\tau^2+3616\lambda^3+3616\tau^3- \\ 9440\lambda^2-9440\tau^2-9440\lambda-9440\tau+24000)+t^2(624\lambda^3\tau+624\lambda\tau^3-4116\lambda^2\tau-4116\lambda\tau^2+ \\ 8220\lambda\tau-216\lambda^3\tau^3+624\lambda^3\tau^2+624\lambda^2\tau^3-1236\lambda^2\tau^2-2256\tau^3-2256\lambda^3+8220\lambda^2+8220\tau^2- \\ 4320\lambda-4320\tau-4320)+t(-120\lambda\tau+144\lambda^3\tau^3-576\lambda^3\tau^2-576\lambda^2\tau^3+2184\lambda^2\tau^2+384\lambda^3\tau+ \\ 384\lambda\tau^3-1176\lambda^2\tau-1176\lambda\tau^2+1440\lambda+1440\tau+384\lambda^3+384\tau^3-1560\lambda^2-1560\tau^2)+(-360 \\ \lambda\tau-21\lambda^3\tau^3-391\lambda^2\tau^2+92\lambda^2\tau^3+92\lambda^3\tau^2+390\lambda^2\tau+390\lambda\tau^2-96\lambda^3\tau-96\lambda\tau^3) \end{pmatrix}}{\begin{pmatrix} 360(\lambda+\tau)-49(\lambda^2\tau+\lambda\tau^2)+16(\lambda^3\tau+\lambda\tau^3)-24(\lambda^2\tau^3+\lambda^3\tau^2)+16(\lambda^3+\tau^3)-65(\lambda^2+\tau^2) \\ -5\lambda\tau+91\lambda^2\tau^2+6\lambda^3\tau^3 \end{pmatrix}}$$

DOI: 10.56892/bima.v8i3B.842

$$\beta_1'(t)=\frac{\begin{pmatrix} t^5(-1728\lambda^2\tau-1728\lambda\tau^2+3648\lambda\tau+576\lambda^2\tau^2-1728\lambda^2-1728\tau^2+5376\lambda+5376\tau)+t^4(720\lambda^3\tau \\ +720\lambda\tau^3+1440\lambda^2\tau+1440\lambda\tau^2-7920\lambda\tau-240\lambda^3\tau^2-240\lambda^2\tau^3+480\lambda^2\tau^2+720\lambda^3+720\tau^3+ \\ 720\lambda^2+720\tau^2-8640\lambda-8640\tau)+t^3(-1184\lambda^3\tau-1184\lambda\tau^3+1376\lambda^2\tau+1376\lambda\tau^2+5120\lambda\tau+ \\ 96\lambda^3\tau^3+96\lambda^3\tau^2+96\lambda^2\tau^3-1184\lambda^2\tau^2-1184\lambda^3-1184\tau^3+2560\lambda^2+2560\tau^2+2560\lambda+2560\tau) \\ +t^2(336\lambda^3\tau+336\lambda\tau^3-624\lambda^2\tau-624\lambda\tau^2-960\lambda\tau-144\lambda^3\tau^3+336\lambda^3\tau^2+336\lambda^2\tau^3-624\lambda^2\tau^2+ \\ 336\lambda^3+336\tau^3-960\lambda^2-960\tau^2)+(9\lambda^3\tau^3+80\lambda^2\tau^2-28\lambda^2\tau^3-28\lambda^3\tau^2) \end{pmatrix}}{\begin{pmatrix} 180(\lambda+\tau)-49(\lambda^2\tau+\lambda\tau^2)+16(\lambda^3\tau+\lambda\tau^3)-24(\lambda^2\tau^3+\lambda^3\tau^2)+16(\lambda^3+\tau^3)-65(\lambda^2+\tau^2) \\ -5\lambda\tau+91\lambda^2\tau^2+6\lambda^3\tau^3 \end{pmatrix}}$$

$$\beta_2'(t)=\frac{\begin{pmatrix} t^5(-576\lambda^2\tau-576\lambda\tau^2+192\lambda\tau+576\lambda^2\tau^2-576\lambda^2-576\tau^2+768\lambda+768\tau)+t^4(240\lambda^3\tau+ \\ 240\lambda\tau^3+480\lambda^2\tau+480\lambda\tau^2-480\lambda\tau-240\lambda^3\tau^2-240\lambda^2\tau^3+240\lambda^3+240\tau^3+240\lambda^2+ \\ 240\tau^2-720\lambda-720\tau)+t^3(-224\lambda^3\tau-224\lambda\tau^3-64\lambda^2\tau-64\lambda\tau^2+320\lambda\tau+96\lambda^3\tau^3+96 \\ \lambda^3\tau^2+96\lambda^2\tau^3-224\lambda^2\tau^2-224\lambda^3-224\tau^3+160\lambda^2+160\tau^2+160\lambda+160\tau)+t^2(48\lambda^3\tau+ \\ 48\lambda\tau^3-12\lambda^2\tau-12\lambda\tau^2-60\lambda\tau-72\lambda^3\tau^3+48\lambda^3\tau^2+48\lambda^2\tau^3-12\lambda^2\tau^2+48\lambda^3+48\tau^3- \\ 60\lambda^2-60\tau^2)+(3\lambda^3\tau^3-4\lambda^3\tau^2-4\lambda^2\tau^3+5\lambda^2\tau^2) \end{pmatrix}}{\begin{pmatrix} 360(\lambda+\tau)-49(\lambda^2\tau+\lambda\tau^2)+16(\lambda^3\tau+\lambda\tau^3)-24(\lambda^2\tau^3+\lambda^3\tau^2)+16(\lambda^3+\tau^3)- \\ 65(\lambda^2+\tau^2)-5\lambda\tau+91\lambda^2\tau^2+6\lambda^3\tau^3 \end{pmatrix}} \tag{12}$$

Evaluating (11) and (12) at $t=1$ yields the discrete scheme

$$y_{n+2}=\partial_2 y_{n+\tau}+\alpha_1 y_{n+1}-\partial_1 y_{n+\lambda}-\alpha_0 y_n+\frac{h^2}{6}(\beta_2 f_{n+2}-2\beta_1 f_{n+1}+\beta_0 f_n) \tag{13}$$

with first derivative

$$y'_{n+2}=\frac{1}{h}\left(\partial'_2 y_{n+\tau}+\alpha'_1 y_{n+1}-\partial'_1 y_{n+\lambda}-\alpha'_0 y_n\right)+\frac{h}{6}(\beta'_2 f_{n+2}-2\beta'_1 f_{n+1}+\beta'_0 f_n) \tag{14}$$

where

$$\partial_1=\left(-2(\tau-2)(4\tau^3+9\tau^2+178\tau-12)\right)$$

$$\partial_2=\left(2(\lambda-2)(4\lambda^3+9\lambda^2+178\lambda-12)\right)$$

$$\alpha_1=\begin{pmatrix} 2(\tau-2)(\lambda-2)\{6\lambda^3\tau^3+49\lambda^2\tau^2-20\lambda\tau+4(\lambda^3\tau+\lambda\tau^3)-18(\lambda^3\tau^2+\lambda^2\tau^3) \\ +8(\lambda^3+\tau^3)-20(\lambda^2+\tau^2)\} \end{pmatrix}$$

$$\alpha_0=\begin{pmatrix} -(\tau-2)(\lambda-2)\{21(\lambda^2\tau+\lambda\tau^2)-8(\lambda^3\tau+\lambda\tau^3)-12(\lambda^3\tau^2+\lambda^2\tau^3)+3\lambda\tau+19\lambda^2\tau^2 \\ +6\lambda^3\tau^3-18(\lambda+\tau)+9(\lambda^2+\tau^2)-12\} \end{pmatrix}$$

$$\beta_2=\left((\tau-2)(\lambda-2)\{-6(\lambda\tau^2+\lambda^2\tau)-4(\lambda^2+\tau^2)+6(\lambda-\tau)+3\lambda^2\tau^2+5\lambda\tau\}\right)$$

$$\beta_1 = \left(-(\tau-2)(\lambda-2)\{30(\lambda\tau^2+\lambda^2\tau)+20(\lambda^2+\tau^2)-48(\lambda-\tau)-15\lambda^2\tau^2-52\lambda\tau\}\right)$$

$$\beta_0 = \left((\tau-2)(\lambda-2)\{-6(\lambda\tau^2+\lambda^2\tau)+3\lambda^2\tau^2+5\lambda\tau-4(\lambda^2-\tau^2)+760(\lambda-\tau)-36\}\right) \quad (15)$$

$$\partial_1' = \left(-2(128-16\tau^4+2417\tau^3-252\tau^2+128\tau)\right)$$

$$\alpha_1' = \begin{pmatrix} 2\{240\lambda\tau-48(\lambda\tau^2+\lambda^2\tau)-30(\lambda^3\tau^4+\lambda^4\tau^3)-180(\lambda^3\tau^2+\lambda^2\tau^3)+40(\lambda^2\tau^4+\lambda^4\tau^2) \\ -48(\lambda^3+\tau^3)+240(\lambda^2+\tau^2)-320(\lambda-\tau)+6\lambda^4\tau^4+145\lambda^3\tau^3+200\lambda^2\tau^2\} \end{pmatrix}$$

$$\partial_2' = \left(2(128-16\tau^4+2417\tau^3-252\tau^2+128\tau)\right)$$

$$\alpha_0' = \begin{pmatrix} -2\{16(\lambda\tau^4+\lambda^4\tau)-49(\lambda\tau^3+\lambda^3\tau)-53(\lambda\tau^2+\lambda^2\tau)+252\lambda\tau+11\lambda^2\tau^2+91\lambda^3\tau^3+6\lambda^4\tau^4 \\ +16(\lambda^4+\tau^4)-113(\lambda^3+\tau^3)+252(\lambda^2+\tau^2)-128(\lambda+\tau)-24(\lambda^4\tau^3+\lambda^3\tau^4)+16(\lambda^4\tau^2 \\ +\lambda^2\tau^4)-49(\lambda^3\tau^2-\lambda^2\tau^3)-128\} \end{pmatrix}$$

$$\beta_2' = \begin{pmatrix} 64(\lambda\tau^3+\lambda^3\tau)-172(\lambda\tau^2+\lambda^2\tau)-100(\lambda^2\tau^3+\lambda^3\tau^2)+64(\lambda^3+\tau^3)+916(\lambda^2+\tau^2)+ \\ 208(\lambda+\tau)-28\lambda\tau+345\lambda^2\tau^2+27\lambda^3\tau^3 \end{pmatrix}$$

$$\beta_1' = \begin{pmatrix} 464(\lambda\tau^2+\lambda^2\tau)-128(\lambda\tau^3+\lambda^3\tau)+164(\lambda^2\tau^3+\lambda^3\tau^2)-128(\lambda^3+\tau^3)+592(\lambda^2+\tau^2) \\ -704(\lambda+\tau)-112\lambda\tau-672\lambda^2\tau^2-399\lambda^3\tau^3 \end{pmatrix}$$

$$\beta_0' = \begin{pmatrix} 314(\lambda^2 s+\lambda s^2)-32(\lambda s^3+\lambda^3 s)-4(\lambda^2 s^3+\lambda^3 s^2)+66(\lambda^3+s^3)-428(\lambda^2+s^2)+976(\lambda+s) \\ -768-964\lambda s-51\lambda^2 s^2+3\lambda^3 s^3 \end{pmatrix}$$

(16)

The values of λ and τ is taken at various points in the interval $\lambda \in (0,1)$ and $\tau \in (1,2)$ to obtain a particular discrete hybrid method. For the purpose of testing the properties of (13) and (14), the values are taken as $\lambda = \frac{1}{4}$ and $\tau = \frac{7}{4}$ to give;

$$y_{n+2} = \frac{1}{837}(2048y_{n+\frac{7}{4}} - 2422y_{n+1} + 2048y_{n+\frac{1}{4}} - 837y_n) + \frac{h^2}{372}(7f_{n+2} - 154f_{n+1} + 7f_n), \quad (17)$$

$$y'_{n+2} = \frac{1}{1494045h}\left(9565184y_{n+\frac{7}{4}} - 19350499y_{n+1} + 16421888y_{n+\frac{1}{4}} - 6636573y_n\right) + \frac{h}{189720}\left(31041f_{n+2} - 363478f_{n+1} + 15417f_n\right), \quad (18)$$

DOI: 10.56892/bima.v8i3B.842

respectively.
The order and error constants of the method (17) and its derivative (18) are respectively confirmed to be $p=6, c_{p+2}=-1.4818\times10^{-5}$ and $p=5, c_{p+2}=-5.7190\times10^{-5}$. The method is consistent and zero stable, satisfying the necessary and sufficient conditions for convergence of linear multistep methods.

Implementation of the method
We require additional closed starting values, $\beta_k=0$, for the evaluation of $f_{n+i}, i=1,2,\cdots,k$ since the method and its derivative are open, $\beta_k\neq0$.
In this work, attempts are made to derive the main closed predictors of the same order of accuracy as starting values. The following symmetric explicit predictor method and its derivative of the same order of accuracy with the corrector method are developed using the same procedure in section 2;

$$y_{n+2}=\frac{1}{2187}\left(3328y_{n+\frac{7}{4}}-2282y_{n+1}+3328y_{n+\frac{1}{4}}-2187y_n\right)+\frac{h^2}{729}\left(56f_{n+\frac{1}{4}}-7f_{n+1}+56f_{n+\frac{7}{4}}\right) \quad (19)$$

$$y'_{n+2}=\frac{1}{5404077h}\left(-7374464y_{n+\frac{7}{4}}+15674029y_{n+1}+10541440y_{n+\frac{1}{4}}-18841005y_n\right)+\frac{h}{2573370}\left(1691096f_{n+\frac{7}{4}}+3716789f_{n+1}+1400792f_{n+\frac{1}{4}}-173502f_n\right) \quad (20)$$

The main predictor (19) and its associated derivative (20) are each of order $p=6$ and $p=5$ respectively while their error constants c_{p+2} are -1.4818×10^{-5} and 5.7190×10^{-5} respectively.
Taylor series expansion was employed to generate other explicit schemes for $y_{n+\lambda}$, $y'_{n+\lambda}$ in Kayode and Obarhua (2015).

Numerical Experiments
Some numerical examples are presented to show the accuracy of the developed Hybrid Block Method (HBM). In the examples considered the absolute errors were obtained as $Err=|y_i-y(x_i)|$, where y_i is the approximate solution obtained using the new method BHM and $y(x_i)$ is the exact solution of the problem considered at the grid points.

DOI: 10.56892/bima.v8i3B.842

Problem 1: Dynamic Problem

A 10-kg mass is attached to a spring having a spring constant of 140N/m. the mass is started in motion from the equilibrium position with an initial velocity of 1m/s in the upward direction and with an applied external force $F(t) = 5\sin t$. Find the subsequent motion of the mass if the force due to air resistance is $-90\dot{x}N$.

Solution:

From the Newton's second law

$$m\ddot{x} = -kx - a\dot{x} + F(t) \tag{21}$$

or

$$\ddot{x} + \frac{a}{m}\dot{x} + \frac{k}{m}x = \frac{F(t)}{m} \tag{22}$$

When the system start with initial velocity v_0 and initial position x_0 at $t = 0$, with initial conditions

$$x(0) = x_0, \quad \dot{x}(0) = v_0 \tag{23}$$

If $m = 10,\ k = 140,\ a = 90$ and $F(t) = 5\sin t$. The equation of motion (21) yields

$$\ddot{x} + 9\dot{x} + 14x = \frac{1}{2}\sin t \tag{24}$$

Applying the initial conditions $x(0) = 0$ and $\dot{x}(0) = -1$ to equation (24) and solve using maple function to get

Exact solution: $$x(t) = -\frac{9}{50}e^{-2t} + \frac{99}{500}e^{-7t} - \frac{9}{500}\cos(t) + \frac{13}{500}\sin(t) \tag{25}$$

Table 1: Comparison Results of EAO2015 and ENM17.

t	$Ex - s$	$Cp - s$	EAO2015	ENM17
0.1	-0.064362051546	-0.064362051521	1.274442e-08	2.51324E-11
0.2	-0.084307205226	-0.084307205211	3.044226e-08	2.64530E-11
0.3	-0.084052253134	-0.084052253180	4.150135e-08	4.60000E-11
0.4	-0.075293042133	-0.075293042098	4.538448e-08	4.40671E-11
0.5	-0.063570639604	-0.063570639412	4.429806e-08	1.92371E-10
0.6	-0.051421170694	-0.051421170420	4.046609e-08	2.74100E-10
0.7	-0.039930529564	-0.039930529132	3.547450e-08	4.32623E-10
0.8	-0.029498658628	-0.029498656717	3.028463e-08	1.91120E-09
0.9	-0.020212691313	-0.020212690146	2.540758e-08	1.16734E-09
1.0	-0.012026994254	-0.012026992356	2.107144e-08	1.89888E-09

Table 1 shows the comparison of the absolute errors in Areo and Omojola (2015) and errors in the new hybrid predictor-corrector method for problem 1, for k=2, h=0.1. The new method has a better

DOI: 10.56892/bima.v8i3B.842

accuracy than Areo and Omojola (2015) of the same order of accuracy.

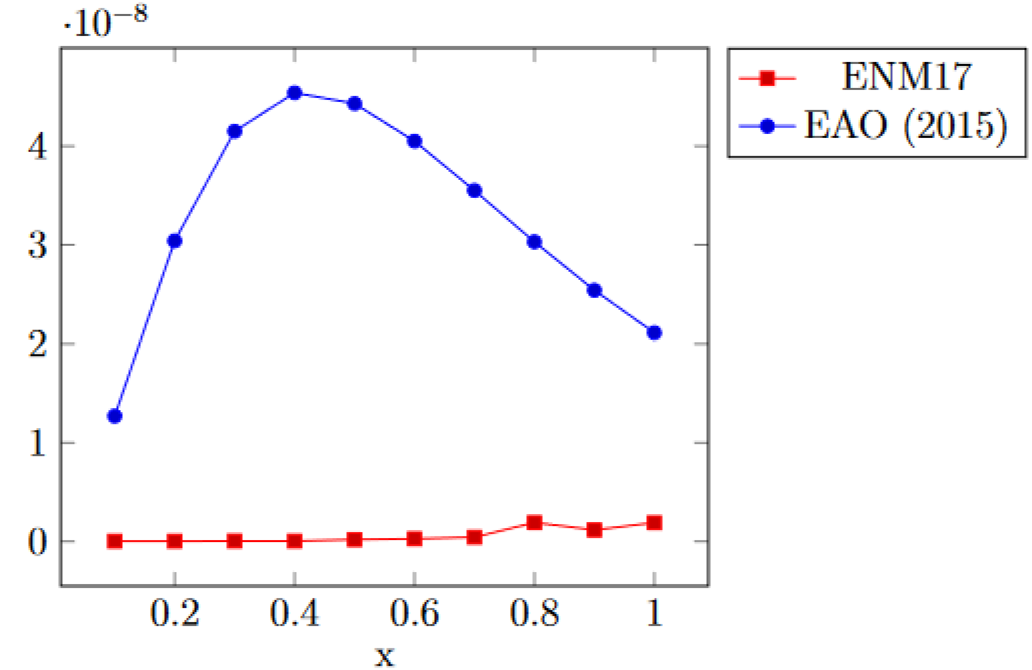

Figure 1: Error plot of Areo and Omojola (2015) and that of the new method, ENM17 for Problem 1.

Problem 2. Cooling of a body

The temperature y degrees of a body, t minutes after being placed in a certain room, satisfies the differential equation $3y'' + y' = 0$. By using the substitution $z = y'$, or otherwise, find y in terms of t given that $y = 60$ when $t = 0$ and $y = 35$ when $t = 6\ln 4$. Find after how many minutes the rate of cooling of the body will have fallen below one degree per minute, giving your answer correct to the nearest minute.

Problem formulation 2

$$y'' = -\frac{y'}{3},\ y(0) = 60,\ y'(0) = -\frac{80}{9},\ \ h = 0.1$$

Exact solution is

$$y(t) = \frac{80}{3} e^{-(\frac{1}{3})t} + \frac{100}{3}$$

DOI: 10.56892/bima.v8i3B.842

Table 2: Comparison Results of EOO2018, EO2019 and ENM17 for Problem 2.

t	$Ex-s$	$Cp-s$	EOO2018 $p=5, k=2$	EO2019 $p=6, k=2$	ENM17, $p=6, k=2$
0.1	59.125762679520165000	59.125762679520165487	3.55e-11	2.344791e-13	4.87021e-16
0.2	58.280186267509812000	58.280186267509812120	4.58e-11	2.202682e-13	1.20035e-16
0.3	57.462331147625591000	57.462331147625596700	7.00e-11	3.935749e-12	5.70104e-15
0.4	56.671288507811937000	56.671288507811938412	6.50e-12	2.704951e-12	1.41252e-15
0.5	55.906179330416379000	55.906179330416374360	3.33e-11	7.599112e-12	4.64026e-15
0.6	55.166153415412850000	55.166153415412858000	4.20e-11	1.569518e-12	8.01700e-15
0.7	54.450388435647511000	54.450388435647431560	4.38e-11	2.756872e-12	7.94415e-14
0.8	53.758089023057302000	53.758089023057356000	1.07e-10	4.375392e-11	5.40034e-14
0.9	53.088485884845809000	53.088485884845636000	6.58e-11	6.474571e-11	1.73041e-13
1.0	52.440834948634382000	52.440834948633421000	6.69e-10	9.100178e-11	9.61140e-13

Table 2 shows the comparison of the absolute errors in Omole and Ogunware (2018), Obarhua (2019) and errors in the new hybrid predictor-corrector method for problem 2, for k=2, h=0.1. The new method has a better accuracy than Omole and Ogunware (2018) and Obarhua (2019) of the same order of accuracy and block methods.

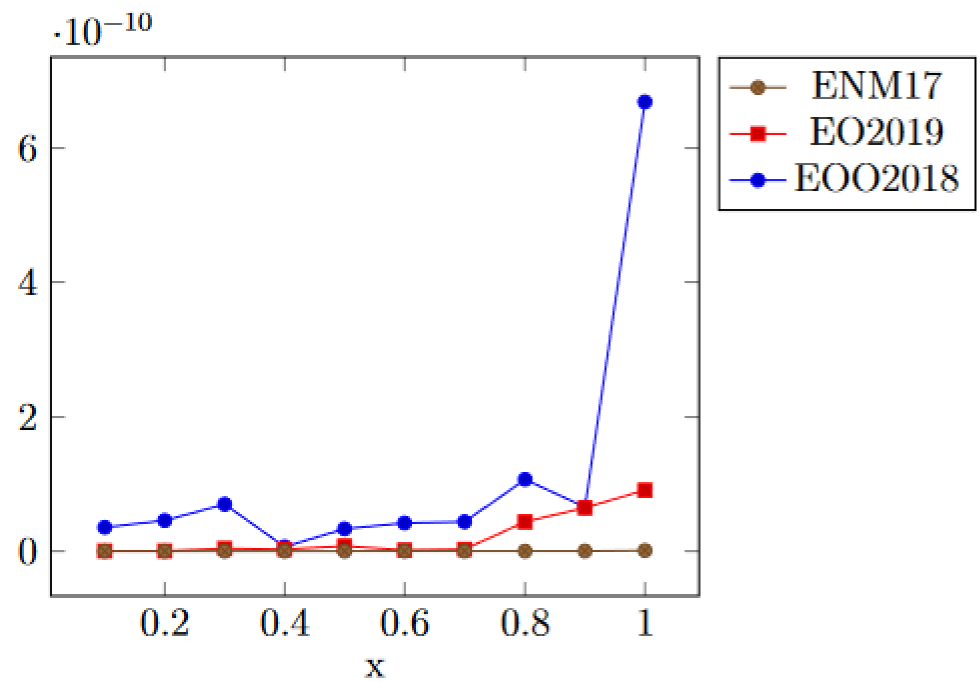

Figure 2: Error plot of Omole and Ogunware (2018), Obarhua (2019) and that of the new hybrid method for Problem 2.

DOI: 10.56892/bima.v8i3B.842

Problem 3: (Two body Problem)

$$y_1'' = \frac{-y_2}{r}, \quad y_1(0) = 1, \quad y_1'(0) = 0,$$

$$y_2'' = \frac{-y_1}{r}, \quad y_2(0) = 0, \quad y_2'(0) = 1$$

$$r = \sqrt{y_1^2 + y_2^2}$$

Exact solution is $y_1(t) = \cos t; \;\; y_2(t) = \sin t$

The maximum errors $\left| y_{exact} - y_{computed} \right|$ obtained with the method for Problem 3, the execution time in microseconds t_e and the total steps taken are compared with that of [9] two step two point method.

Table 3: Shows the numerical solution of our method compared with the method of EKA2013 and EOETAL2023.

TOL	EKA2013			EOETAL2023	ENM17		
	TS	**MAXE**	t_e		**TS**	**MAXE**	t_e
10^{-2}	33	9.763298E-08	635	1.35782080E-11	33	1.3452E-13	55
10^{-4}	55	4.170707E-10	1346	1.96524162E-10	55	2.2415E-16	58
10^{-6}	74	2.100171E-12	2614	3.79459225E-10	74	4.3425E-17	62
10^{-8}	130	3.214551E-15	2788	5.62369776E-10	130	3.3256E-17	120
10^{-10}	278	2.473336E-17	5590	7.45242199E-10	278	5.4132E-18	140

Table 3 shows the comparison of the absolute errors in Kayode and Adeyeye (2013), Omole *et al.* (2023) and errors in the new hybrid predictor-corrector method for problem 3. The new method has a better accuracy and execution time than Kayode and Adeyeye (2013) and Omole *et al.* (2023).

DOI: 10.56892/bima.v8i3B.842

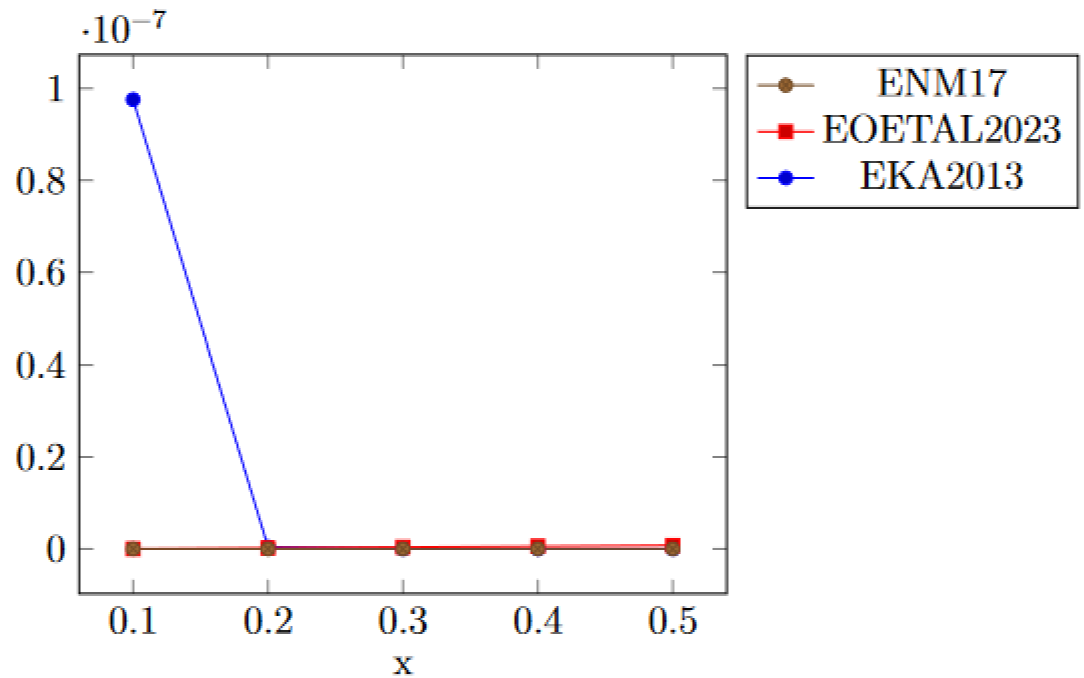

Figure 3: Error plot of Kayode and Adeyeye (2013), Omole *et al.* (2023) and that of the new hybrid method for Problem 3.

Discussion of Results

Table 1 shows the outcome of the newly developed method in comparison to Areo and Omojola's (2015) method of the same order of accuracy. The outcome demonstrates that the new method outperformed Areo and Omojola's (2015).

Problem 2 in Omole and Ogunware (2018) and Obarhua (2019) was solved using the new method (17). Despite having larger stepnumbers, Table 2's results are more accurate than those of Omole and Ogunware (2018) and Obarhua (2019).

For the purpose of comparison with Kayode and Adeyeye (2013)'s 3-step method and higher order of accuracy and Omole *et al.* (2023)'s of order nine, the new method was similarly applied to solve Problem 3 in both studies. The proposed method outperformed Kayode and Adeyeye (2013) and Omole *et al.* (2023) in terms of accuracy and efficiency, according to the data displayed in Table 3.

Additionally, error plots were utilized to assess the new method's smoothness, consistency, and convergence in comparison to the methods that were already in use and were shown Figures 1-3. The new method's smooth, consistent, and convergent nature over the considered existing method is demonstrated by the curves in Figures 1-3.

Conclusion

A new hybrid predictor-corrector method for the direct solution of universal second order initial value problems of ODEs has been developed in this study. The method is developed in such a way that the hybrid points are at $y-function$ which enhanced the reduction of function evaluation.Two real-world engineering problems that were modeled as second order (IVPs) were solved using the new method, and Tables 1 and 2

DOI: 10.56892/bima.v8i3B.842

demonstrate how much more accurate the new method is than the previous ones. The Tables 3 demonstrate that the novel method outperforms the methods of Kayode and Adeyeye (2013) and Omole *et al.* (2023) in terms of efficiency and provides a better approximation. The curves illustrated the efficacy and efficiency of the method. As a result, any type of second order ordinary differential equation can be solved using this method.

Abbreviations

TOL – Tolerance

TS – Total Steps taken

MAXE – Magnitude error of the computed solution

t_e – The execution time taken in microseconds

$Ex-s-$ Exact solution

$Cp-s-$ Computed solution

$EAO2015-$ Areo and Omojola (2015)

$EOO2018-$ Omole and Ogunware (2018)

$EO2019-$ Obarhua (2019)

$EKA2013-$ Kayode and Adeyeye (2013)

$EOETAL2023-$ Omole *et al.* (2023)

$ENM17-$ New method (17)

REFERENCES

Areo, E.A. and Adeniyi, R.B. (2013): A Self-Starting Linear Multistep Method for Direct Solution of Initial Value Problems of Second Order Ordinary Differential Equations. *Intern. J. of Pure and Applied Mathematics, 82(3), 345-364.*

Areo, E.A. and Omojola, M.T. (2015): A New One-Twelfth Step Continuous Block Method for the Solution of Modeled Problems of Ordinary Differential Equations. *Amer. J. of Comput. Math., 5, 447-457.*

Hritonenko, H. and Yatsenko, Y. (1999): Aggregate Models of Economics Dynamics. In: Mathematical Modeling in Economics, Ecology and the Environment. Springer, Bosten, *M.A. Vol. 34, pp. 27-40.* https://doi.org/10.1007/978-1-4419-9733-3_2.

Kayode, S.J. and Adeyeye, O. (2013): Two-step two-point Hybrid Methods for General Second Order Differential Equations, Afri, *J. of Math. & Compt. Sci. Res., 6(10), 191-196.*

Kayode, S.J. and Adeyeye, O., (2011): A 3-Step Hybrid Method for Direct Solution of Second Order Initial Value Problems. *Aust. J. Basic Appl. Sci., 5(12), 2121-2126.*

Kayode, S.J. and Obarhua, F.O. (2013): Continuous $y-function$ Hybrid Methods for Direct Solution of Differential Equations. *Intern. J. of Diff. Eq. 12(1), 37-48.*

Kayode, S.J. and Obarhua, F.O., (2015): 3-Step *y-function* Hybrid Methods for Direct Numerical Integration of Second Order IVPs in ODEs. *Theo. Math. & Appl., 5(1), 39-51.*

Obarhua, F.O. (2019): An Efficient Two-Step Symmetric Hybrid Block Method for Solving Second-Order Initial Value Problems of Ordinary Differential Equations. *Int. J. Res. & Sci. Innov. Vol. 6, Issue 11, pp. 200-208.*

DOI: 10.56892/bima.v8i3B.842

Obarhua, F.O. and Kayode, S.J., (2016): Symmetric Hybrid Linear Multistep Method for General Third Order Differential Equations. *Open Access Library Journal*, ***3***: *e2583*, *http://dx.doi.org/10.4236/oalib.1102583*

Okor, T. and Nwachuckwu, G.C. (2022): Generalized Cash-Type Second Derivative Generalized Extended Backward Differentiation Formulas for Stiff Systems of ODEs. *Journal of the Nigerian Mathematical Society, Vol. 41, Issue 2, pp. 163-191.*

Olabode, B.T. and Momoh, A.L (2016).: Continuous Hybrid Multistep Methods with Legendre Basis Function for Direct Treatment of Second Order stiff ODEs, *Amer. J. of Comput. & Appli. Maths. 6(2) 38-49.*

Omole, E.O. and Ogunware, B.G. (2018): 3-Point Single Block Method (3PSHBM) for Direct Solution of General Second Order Initial Value Problem of Ordinary Differential Equations. *Journal of Scientific Research & Reports, 20(3) 1-11.*

Omole, E.O., Obarhua, F.O., Familua, A.B. and Shokri, A. (2023): Algorithms of Algebraic Order Nine for Numerically Solving Second-Order Boundary and Initial Value Problems in Ordinary Differential Equations. *Int. J. Mathematics in Operational Research, Vol. 25, No. 3, pp. 343-368.*

Ogunfeyitimi, S.E. and Ikhile, M.N.O., (2019): Second Derivative Generalized Extended Backward Differentiation Formulas for Stiff Problems. *Journal of the Korea Society Ind. 23(3), 170-202.* *https://*DOI;10.12941/jksiam.2019.23.179.

Ramos, H. and Rufai, M.A. (2019): A Third-Derivative Two-Step Block Falkner-Type Method for Solving General Second-Order Boundary-Value Systems. *Mathematics and Computer in Simulation (MATCOM), Vol. 165, No. C, pp. 139-155*, Elsevier, DOI: 10.1016/j.matcom.2019.

DOI: 10.56892/bima.v8i3B.842