International Journal of Mechanical Engineering and Technology (IJMET)
Volume 9, Issue 13, December 2018, pp. 1577–1592, Article ID: IJMET_09_13_160
Available online at http://www.iaeme.com/ijmet/issues.asp?JType=IJMET&VType=9&IType=13
ISSN Print: 0976-6340 and ISSN Online: 0976-6359

MATHEMATICA COMPUTER PROGRAMMING CODES OF EXPONENTIALLY FITTED CONCURRENT MILNE'S DEVICE FOR SOLVING SPECIAL PROBLEMS

Jimevwo Godwin Oghonyon, Timothy Olusoji Mosaku
Department of Mathematics, Covenant University, Ota, Ogun State, Nigeria

Matthew RemilekunOdekunle
Department of Mathematics, ModibboAdama University of Technology, Yola, Nigeria

Ogbu Famous Imaga
Department of Building Technology, Covenant University, Ota, Ogun State, Nigeria

Temitope Abodunrin
Department of Physics, Covenant University, Ota, Ogun State, Nigeria

ABSTRACT

Over the years, scientific computing has contributed immensely to computational mathematics. Mathematica computer programming codes is known to provide easy computation and quick results. This research article is specifically built to generate Mathematica computer programming codes of exponentially fitted concurrent Milne's device (EFCMD) for solving special problems. Exponentially fitted concurrent Milne's device is formulated via collocation/interpolation with power series as the approximate solution. Analyzing the EFCMD will produce the main local truncation error (MLTE) after showing the order, thereby bringing forth the bounds of convergence. Numerical results were shown to demonstrate the functioning of Mathematica programming codes of EFCMD for resolving special problems at some selected bounds of convergence. The finished results were obtained with the assistance of Mathematica 9 kernel. Numerical results display that EFCMD do better than existing methods in terms of the maximum errors in the least studied bound of convergence as a result of varying/designing a suitable pace size, ascertain bound of convergence and error control.

Keywords: Mathematica computer programming codes, EFCMD, Bound of convergence, Main local truncation error.

Cite this Article: Jimevwo Godwin Oghonyon, Matthew Remilekun Odekunle, Timothy Olusoji Mosaku, Ogbu Famous Imaga and Temitope Abodunrin, Mathematica Computer Programming Codes of Exponentially Fitted Concurrent Milne's Device For Solving Special Problems, International Journal of Mechanical Engineering and Technology, 9(13), 2018, pp. 1577–1592.
http://www.iaeme.com/ijmet/issues.asp?JType=IJMET&VType=9&IType=13

1. INTRODUCTION

This study views special problems having a particular feature of the approximative solution been acknowledged in advance. Such particular problems are of the form D' Ambrosio et al. (2011); Gatuschi (2013); Lambert (1973) and Ngwane &Jator (2013).

$$y'' = f(x, y), \quad y(x_0) = y_0, \quad y'(x_0) = y'_0 \quad \text{for } x\epsilon[x_0, X] \tag{1}$$

Where $f: R \times R^k \to R^k$, k is the proportion of the physical organization.D

Equation (1) is known to satisfy both theorems stated below Bond (2009) and Ken et al. (2011) arises from areas of science discipline and applied sciences such as Newtonian mechanics, celestial bodies/universe, quanta theory, control theory, electrical circuit and biology. Scientific computing of distinct technique shave been established by well-known exponentially fitted method with recognized frequency and most of the important results exist in literatures. (SeeAnake et al. (2012); Anake&Adoghe(2013);Edeki et al. (2015); Eke et al. (2017); Eke et al. (2018); Gatuschi (2013); Lambert (1973); Owoloko et al. (2015)). Bookmen have proposed and execute equation (1) to produce the sought-after result. This includes; Jikantoro et al. (2015) computed zero-dissipative trigonometrically fitted hybrid method for numerical solution of oscillatory problems applying MAPLE 16. Ngwane &Jator (2013), Ngwane &Jator (2013) and Ngwane &Jator (2014) implemented all computations using Matlab programming language. Once more,D' Ambrosio et al. (2011), D' Ambrosio et al. (2012),Jator (2010),Majid et al. (2012) deploy C language and executed on DYNIX/ptx operating system to solve directly general third order ODEs using two-point four step block method. Ngwane &Jator (2015) and Ngwane &Jator (2017) accomplished numerical application employing a composed encrypt in Mathematica 10. 0 to show the efficiency and accuracy of the techniques. Nevertheless, Adejumo et al. (2014) and Calvo et al. (2015) implemented all numerical computations on a PC using PYTHON programming language. Waeleh& Majid (2016) executed A 4-point block method for solving higher order ODEs employing Matlab ode45 and ode45: Runge-Kutta-Dormand-Prince ODE solver. Waeleh (2011) carried-out a new algorithm for solving higher order IVPs of ODEs utilizing C language. Several gaps were observed from the authors listed above. This includes; using a fixed step size and inability to decide on a suitable step size, lack of bound of convergence to ensure convergence of the method and lastly, lengthy computation without controlling the error.

From the gaps named earlier, this study in addition is built to design a suitable step size and varying the step size, determine the bounds of convergence to check convergence and control error. This is primarily the objective of developing a Mathematica programming code of exponentially fitted concurrent Milne's device for solving special problems. (SeeAscher

&Petzoid (1998);Dormand (1996);Faires& Burden (2012); Lambert (1973);Lambert (1991);Oghonyon et al. (2015), Oghonyon et al. (2016), Oghonyon et al. (2016),Oghonyon et al. (2018), Oghonyon et al. (2018) and Oghonyon et al. (2018).

Definition: we consider$d-$ concurrent$,b-$ stage method, when r refers to concurrent size and $\hat{h}$ is step size while concurrent size in time,$d\hat{h}$**.** Consider$t=0,1,2,\dots$ form concurrent amount and$n=rb$, then $d-$ concurrent, $b-$ stage method is composed as next universal category:

$$\overline{Y}_\tau=\sum_{v=1}^{d}A_vY_{\tau-v}+\hat{h}\sum_{v=0}^{d}B_vF_{\tau-v}, \qquad (2)$$

where

$$Y_\tau=[g_{n+1},\dots,g_{n+i},\dots,g_{n+b}]^{\overline{T}},$$

$$F_\tau=\left[\bar{f}_{n+1},\dots,\bar{f}_{n+i},\dots,\bar{f}_{n+b}\right]^{\overline{T}}$$

A_v and B_v are $k\times k$ constants rectangular array of quantities. (SeeIbrahim et al. (2007); Oghonyon et al. (2017)).

Thus, setting off from over explanatory statement, a concurrentsystem has numerical gains for each practical application program, the end result is evaluated to a greater magnitude at the same time interval. The amount of stages relies on the construction of the concurrent system. Hence, using these techniques can permit more immediate and faster results of the problem which can be treated to give the sought-after accuracy. Please refer toMajid & Suleiman (2007), Majid & Suleiman (2008Oghonyon et al. (2016), Oghonyon et al. (2016), Oghonyon et al. (2018), Oghonyon et al. (2018) and Oghonyon et al. (2018) for more information.

Theorem (Weierstrass Approximation Theorem)

Let $f:R\to R$ be continuous and $2\pi-$periodic. Then for each $\varepsilon>0$, there exists a trigonometric polynomial $P(x)=\sum_{j=-n}^{k}c_j\,e^{ijx}$ such that for all x, $|f(x)-P(x)|<\varepsilon$. Tantamountly, as for any such f, there must exist a successive polynomial such that $P_n\to f$ in a uniform manner on R. (Bond, (2009)).

Theorem (Existence and Uniqueness)

Let $f(x,y)$ exist and remain continuous for all stages(x,y) in the neighborhood$D$constituted by $a\le x\le b$, $-\infty<y<\infty$, where a and b are finite real constants and let there subsists a constant quantity$L$ such that for any $x\in[a,b]$ and any two numbers y and y^*, $|f(x,y)-f(x,y^*)|\le L|y-y^*|$.

We acknowledge this prediction as theLipschitz consideration, since there is precisely a single function $y(x)$ having the following four properties:

- $y(x)$ is continuous and differentiable for $x\in[a,b]$,
- $y''(x)=f(x,y(x)),x\in[a,b],f\in C^2[a,b]$,
- $y(x_0)=y_0$,
- $y'(x_0)=y_0'$. (Look Ken et al (2011)).

The subject remainder is as follows: in Subsection 2, we focus on the Mathematica Computer Programming Codes of Exponentially Fitted Method. In Subsection 3, we provide Mathematica Computer Programming Codes of the studied methods with the Numerical Results and

Discussion. Last be not least, Subsection 4 gives similar Conclusion as cited in Akinfenwa et al. (2013), Oghonyon et al. (2016), Oghonyon et al. (2016), Oghonyon et al. (2018), Oghonyon et al. (2018) and Oghonyon et al. (2018).

MATERIALS AND METHODS

Under this subsection, the objective to be reached is to devise Mathematica computer programming codes of exponentially fitted concurrent Milne's device. This consist of a unification, $j-$ stepconcurrent predictor system and $j-1-$ stepconcurrent corrector system ofsimilar order. Uniting the representation as

$$y(x) = \sum_{i=0}^{\bar{k}} \alpha_i y^*{}_{n-i}$$

$$+h^2 \sum_{i=0}^{\bar{k}} \beta_i(u) f_{n-i}, \qquad (3)$$

$$y(x) = \sum_{i=0}^{\bar{k}} \alpha_i y^*{}_{n-i}$$

$$+h^2 \sum_{i=1}^{k} \beta_i^*(u) f_{n+i}. \qquad (4)$$

Equation (3) and (4) form the concurrent predictor system and concurrent corrector system of the exponentially fitted concurrent Milne's device with $u = w\bar{h}$, $\beta_i(u)$, $i = 0, 1, 2$ containing characteristics that depends on varying the step size and frequency. Noting that y_{n+k} is the approximants to the exact solutions$y(x_{n+i})$, i.e.,$y(x_{n+i}) \approx y_{n+i}$, and $f(x_{n+i}, y^*{}_{n+i}) \approx f_{n+i}$wherei $= 0, 1, 2$. From equations (3) and (4), exponentially trigonometrically fitted method is used together with interpolation/collocation method to approximate the precise solution $y(x)$ on distinct time intervals of $[x_{n-i}, x_n]$by interpolating subroutine of the form (6)

$$y(x) = \sum_{i=0}^{k} \bar{a}_i \left(\frac{x-x_n}{h}\right)^i + \sum_{i=0}^{1} \frac{e^{wx}}{i!}. \qquad (5)$$

Revising (5) and expanding will give rise to the exponentially fitted method presented in Mathematica programming codes of the form

$$y[x\_] = a[0] + a[1]\frac{(x-x[n])}{h} + a[2]\frac{(x-x[n])^2}{h^2} + a[3]\frac{(x-x[n])^3}{h^3} + a[4]\left(1 + \frac{w(x-x[n])}{h} + \frac{w^2(x-x[n])^2}{2h^2} + \frac{w^3(x-x[n])^3}{6h^3} + \frac{w^4(x-x[n])^4}{24h^4}\right), \qquad (6)$$

where $a_0, a_1, a_2, a_3$and $a_4$for k=4 are unchanging parameters necessary to ensurea particular fashion. Presuming the condition that equation (6) matches the precise solution at about some pick outtime intervalx_{n-1}, x_n to yield approximation as

$$y(x_{n-i}) \approx y_{n-i}, \qquad y(x_n) \approx y_n. \qquad (7)$$

Demanding the interpolation function (7) meets equation (1) at stages where $x_{n+i}, i = 0, 1, 2, 3$ willproduce approximation of the following

$$y''(x_{n-i}) \approx f_{n-i}, \bar{y}''(x_{n+i}) \approx f_{n+i},$$

$$i = 0, 1, 2. \qquad (8)$$

Joining the approximations of (7) and (8) will lead to five-fold systems of equation which gives riseto $Ax = b$. Solving the systems of equation will produce the exponential system of equations constituted as the Mathematica kernel programming codes

$$\mathrm{matrixa} = \left\{\begin{array}{c} \{1,0,0,0,1\}, \\ \left\{1,-1,1,-1,1-w+\frac{w^2}{2}-\frac{w^3}{6}+\frac{w^4}{24}\right\}, \\ \{0,0,2,0,w^2\}, \\ \left\{0,0,2,-6,w^2-w^3+\frac{w^4}{2}\right\}, \\ ,\{0,0,2,-12,w^2-2w^3+2w^4\} \end{array}\right\};$$

$$b = \{y[n], y[n-1], f[n], f[n-1], f[n-2]\};$$

$\{c, e, l, q, u\} = \mathrm{Inverse}[\mathrm{matrixa}].b$ (10)

$$\mathrm{matrixa} = \left\{\begin{array}{c} \{1,0,0,01\}, \\ \left\{1,-1,1,-1,1-w+\frac{w^2}{2}-\frac{w^3}{6}+\frac{w^4}{24}\right\}, \\ \left\{0,0,2,6,w^2+w^3+\frac{w^4}{2}\right\}, \\ \{0,0,2,12,w^2+2w^3+2w^4\}, \\ \left\{0,0,2,18,w^2+3w^3+\frac{9w^4}{2}\right\}, \end{array}\right\};$$

$$b = \left\{\begin{array}{c} y[n], y[n-1], f[n+1], f[n+2], \\ f[n+3] \end{array}\right\};$$

$\{c, e, l, q, t\} = \mathrm{Inverse}[\mathrm{matrixa}].b,$ (11)

to geta_i, $i = 0, 1, 2, 3, 4, 5$ and substituting value of a_i's into (6) will result in the continuous exponentially fitted concurrent Milne's device as

$$y[x\_] = \left(1+\frac{(x-x[n])}{h}\right)y[n] + \left(-\frac{(x-x[n])}{h}\right)y[n-1] + \left(-\frac{1}{w^4} + \frac{\left((-12w)+\frac{(7w^4)}{2}\right)}{(12w^4)}\frac{(x-x[n])^1}{h} + \frac{(-6w^2+6w^4)}{(12w^4)}\frac{(x-x[n])^2}{h^2} + \frac{((-2w^3)+(3w^4))}{(12w^4)}\frac{(x-x[n])^3}{h^3} + \left(\frac{1}{w^4}\right)\frac{(x-x[n])^4}{h^4}\right) f[n]h^2 + \left(\left(\frac{2}{w^4}\right) + \frac{((24w)+(3w^4))}{(12w^4)}\frac{(x-x[n])}{h} + \left(\frac{1}{(w^2)}\right)\frac{(x-x[n])^2}{h^2} + \frac{((4w^3-4w^4)}{(12w^4)}\frac{(x-x[n])^3}{h^3} - \left(\frac{2}{w^4}\right)\frac{(x-x[n])^4}{h^4}\right) f[n -$$

$$1]h^2 + \left(\left(\frac{-1}{w^4}\right) + \frac{\left(-12w-\frac{w^4}{2}\right)}{(12w^4)}\frac{(x-x[n])}{h} - \left(\frac{1}{2w^2}\right)\frac{(x-x[n])^2}{h} + \frac{\left(-2w^3+3w^4\right)}{(12w^4)}\frac{(x-x[n])^3}{h^3} + \left(\frac{1}{w}\right)\frac{(x-x[n])^4}{h^4}\right) f[n-2]h^2, \qquad (12)$$

$$y[x] = \left(1 + \frac{(x-x[n])}{h}\right)y[n] + \left(-\frac{(x-x[n])}{\bar{h}}\right)y[n-1] + \left(\left(-\frac{1}{w^4}\right) - \frac{\left(12w-\frac{47w^4}{2}\right)}{(12w^4)}\frac{(x-x[n])}{h} + \frac{\left(-6w^2-18w\right)}{(12w)}\frac{(x-x[n])^2}{h^2} - \frac{\left(2w^3+5w^4\right)}{(12w^4)}\frac{(x-x[n])^3}{h^3} + \left(\frac{1}{w^4}\right)\frac{(x-x[n])^4}{h^4}\right) f[n+1]h^2 + \left(\left(\frac{2}{w^4}\right) + \frac{\left(-24w+27w^4\right)}{(12w^4)}\frac{(x-x[n])}{h} - \frac{\left(-12w^2+18w^4\right)}{(12w^4)}\frac{(x-x[n])}{h} - \frac{\left(\left(-4w^3\right)-\left(8w^4\right)\right)}{12w^4}\frac{(x-x[n])^3}{h^3} - \left(\frac{2}{w^4}\right)\frac{(x-x[n])^4}{h^4}\right) f[n+2]h^2 + \left(\left(\frac{-1}{w^4}\right) - \frac{\left(12w-19w^4\right)}{(120w^9)}\frac{(x-x[n])}{h} - \frac{\left(6w^2-6w^4\right)}{(12w^4)}\frac{(x-x[n])^2}{h^2} - \frac{\left(\left(2w^3\right)+(3w)\right)}{(12w)}\frac{(x-x[n])^3}{h^3} + \left(\frac{1}{w^4}\right)\frac{(x-x[n])^4}{h^4}\right) f[n+3]h^2. \quad (13)$$

Evaluating the continuous exponentially fitted concurrent Milne's device of equation (12) and (13) at pick outstages of x_{n+i}, $i = 1, 2, 3$ will bring forth exponentially fitted concurrent Milne's device

$$y[x\_] = y[n] + y[n-1] + h^2(\beta_0(w,x)f[n] + \beta_1(w,x)f[n-1] + \beta_2(w,x)f[n-2]), \qquad (14)$$

$$y[x\_] = y[n] + y[n-1] + h^2(\beta_0(w,x)f[n+1] + \beta_1(w,x)f[n+2] + \beta_2(w,x)f[n+3]), \qquad (15)$$

where w is the frequency, $\beta_0(w,x), \beta_1(w,x)and\beta_2(w,x)$are fixed constants.See Abell & Braselton (2009);Faires& Burden (2012); Ngwane &Jator (2013); Ngwane &Jator (2013); Ngwane &Jator (2014);Ngwane &Jator (2015); Ngwane &Jator (2017); Oghonyon et al. (2016), Oghonyon et al. (2016), Oghonyon et al. (2018), Oghonyon et al. (2018) and Oghonyon et al. (2018)for more details.

Devising Bounds of Convergence for Exponentially Fitted Concurrent Milne's Device:

Set in motion the Mathematica computer programming codes of exponentially fitted concurrent Milne's device, $j-$ stepconcurrent predictor system and $j-1-$ stepconcurrentcorrector system is treated ascon current predictor-corrector joint pair owning ilk range. Combining Asher &Petzoid (1998),Dormand (1996),Faires& Burden (2012), Lambert (1973), Lambert (1991),Oghonyon et al. (2016), Oghonyon et al. (2016), Oghonyon et al. (2018), Oghonyon et al. (2018) and Oghonyon et al. (2018), exponentially fitted concurrent Milne's device indicates that it is workable to find the approximate of the main local truncation error of the concurrent predictor-corrector joint pair in absence of approximating higher differential coefficients, y(x). Presume that $\tilde{\bar{p}} = \bar{\bar{p}}$, where $\bar{\bar{p}}$ and $\tilde{\bar{p}}$sets up range of concurrent predictor system and concurrent corrector system. Now, method of range$\tilde{\bar{p}}$, enquiry of$j-$ stepconcurrent predictor system will generate main local truncation errors

$$\tilde{\tilde{G}}^{[1]}_{\tilde{\tilde{p}}+5}\bar{h}^{\tilde{\tilde{p}}+5}\tilde{\tilde{g}}^{(\tilde{\tilde{p}}+5)}(x_n) = \tilde{g}(x_{n+1}) - \tilde{\tilde{g}}^{[\tilde{l}_1]}_{n+1} - \left(\frac{72+36w+12w^2-19w^3}{72w^3}\right) + O\left(h^{\tilde{\tilde{p}}+6}\right)$$

$$\tilde{\tilde{G}}^{[2]}_{\tilde{\tilde{p}}+5}h^{\tilde{\tilde{p}}+5}\tilde{\tilde{g}}^{(\tilde{\tilde{p}}+5)}(x_n) = \tilde{g}(x_{n+2}) - \tilde{\tilde{g}}^{[\tilde{l}_1]}_{n+2} + \left(\frac{22}{9}+\frac{15}{w^4}-\frac{2}{w^3}-\frac{2}{w^2}-\frac{4}{3w}\right) + O\left(h^{\tilde{\tilde{p}}+6}\right), \quad (16)$$

$$\tilde{\tilde{G}}^{[3]}_{\tilde{\tilde{p}}+5}h^{\tilde{\tilde{p}}+5}\tilde{\tilde{g}}^{(\tilde{\tilde{p}}+5)}(x_n) = \tilde{g}(x_{n+3}) - \tilde{\tilde{g}}^{[\tilde{l}_3]}_{n+3} + \left(\frac{227}{24}+\frac{80}{w^4}-\frac{3}{w^3}-\frac{9}{2w^2}-\frac{9}{2w}\right) + O\left(h^{\tilde{\tilde{p}}+6}\right).$$

Similar investigation of $j-1-$stepconcurrentcorrector gives rise to main local truncation errors

$$\bar{\bar{G}}^{[1]}_{\bar{\bar{p}}+5}h^{\bar{\bar{p}}+5}\bar{\bar{g}}^{(\bar{\bar{p}}+5)}(x_n) = \bar{g}(x_{n+1}) - \bar{\bar{g}}^{[\bar{q}_1]}_{n+1} + \left(\frac{22-16w-85w^3+4w^4+28w^5}{12w^3}\right) + O\left(h^{\bar{\bar{p}}+6}\right),$$

$$\bar{\bar{G}}^{[2]}_{\bar{\bar{p}}+5}h^{\bar{\bar{p}}+5}\bar{\bar{g}}^{(\bar{\bar{p}}+5)}(x_n) = \bar{g}(x_{n+2}) - \bar{\bar{g}}^{[\bar{q}_2]}_{n+2} + \left(\frac{-360+176w+48w^2+32w^3-157w^4}{12w^4}\right)$$

$$+O\left(h^{\bar{\bar{p}}+6}\right) \quad (17)$$

$$\bar{\bar{G}}^{[3]}_{\bar{\bar{p}}+5}h^{\bar{\bar{p}}+5}\bar{\bar{g}}^{(\bar{\bar{p}}+5)}(x_n) = \bar{g}(x_{n+3}) - \bar{\bar{g}}^{[\bar{q}_3]}_{n+1} + \left(-\frac{179}{12}-\frac{160}{w^4}+\frac{22}{w^3}+\frac{9}{w^2}+\frac{9}{w}\right) + O\left(h^{\bar{\bar{p}}+6}\right),$$

$\tilde{\tilde{G}}^{[1]}_{\tilde{\tilde{p}}+5}, \tilde{\tilde{G}}^{[2]}_{\tilde{\tilde{p}}+5}, \tilde{\tilde{G}}^{[3]}_{\tilde{\tilde{p}}+5}, \bar{\bar{G}}^{[1]}_{\bar{\bar{p}}+5}, \bar{\bar{G}}^{[2]}_{\bar{\bar{p}}+5}$ and $\bar{\bar{G}}^{[3]}_{\bar{\bar{p}}+5}$exist asexistent as separate entity of step-size $\bar{h}$ and $y(x)$behave as precise solution to differential coefficient fulfiliinginitial precondition$\bar{g}(x_n) \approx \bar{g}_n$. SeeAsher &Petzoid (1998),Dormand (1996),Faires& Burden (2012), Lambert (1973), Lambert (1991),Oghonyon et al. (2015), Oghonyon et al. (2016), Oghonyon et al. (2016), Oghonyon et al. (2018), Oghonyon et al. (2018) and Oghonyon et al. (2018)for further details.

Moving ahead, precondition for small valuates, $\bar{h}$reached

$$\tilde{\tilde{g}}^{(5)}(x_n) \approx \bar{\bar{g}}^{(5)}(x_n),$$

and application of Mathematica programming codes of exponentially fitted concurrent Milne's device trusts instantly on this precondition stated above.

Reducing further the main local truncation errors of (16) and (17) above, in a similar fashion, throwing off terms of degree $O\left(h^{\bar{\bar{p}}+6}\right)$, it turns easily to attain the mathematical calculation of main local truncation errors of exponentially fitted block Milne's device as

$$\bar{\bar{G}}^{[1]}_{\bar{\bar{p}}+5}h^{\bar{\bar{p}}+5}g^{(\bar{\bar{p}}+5)}(x_n) \approx \frac{510}{529}\left[\tilde{\tilde{g}}^{[\tilde{l}_1]}_{n+1} - \bar{\bar{g}}^{[\bar{q}_1]}_{n+1}\right] < \bar{\varepsilon}_1,$$

$$\bar{\bar{G}}^{[2]}_{\bar{\bar{p}}+5}h^{\bar{\bar{p}}+5}g^{(\bar{\bar{p}}+5)}(x_n) \approx \frac{471}{559}\left[\tilde{\tilde{g}}^{[\tilde{l}_2]}_{n+2} - \bar{\bar{g}}^{[\bar{q}_2]}_{n+2}\right] < \bar{\varepsilon}_2, \quad (18)$$

$$\bar{\bar{G}}^{[3]}_{\bar{\bar{p}}+5}h^{\bar{\bar{p}}+5}g^{(\bar{\bar{p}}+5)}(x_n) \approx \frac{358}{585}\left[\tilde{\tilde{g}}^{[\tilde{l}_3]}_{n+3} - \bar{\bar{g}}^{[\bar{q}_3]}_{n+3}\right] < \bar{\varepsilon}_3.$$

Mentioning the arguments that$\tilde{\tilde{g}}^{[\tilde{l}_1]}_{n+1} \neq \bar{\bar{g}}^{[\bar{q}_1]}_{n+1}, \tilde{\tilde{g}}^{[\tilde{l}_2]}_{n+2} \neq \bar{\bar{g}}^{[\bar{q}_2]}_{n+2}$ and $\tilde{\tilde{g}}^{[\tilde{l}_3]}_{n+3} \neq \bar{\bar{g}}^{[\bar{q}_3]}_{n+3}$known as predicted and corrected approximations brought forth by the exponentially fitted concurrent Milne's device of range $\bar{p}$, $\bar{G}^{[1]}_{\bar{p}+5}h^{\bar{p}+5}g^{(\bar{p}+5)}(x_n)$, $\bar{G}^{[2]}_{\bar{p}+5}h^{\bar{p}+5}g^{(\bar{p}+5)}(x_n)$ and $\bar{G}^{[3]}_{\bar{p}+5}h^{\bar{p}+5}g^{(\bar{p}+5)}(x_n)$

are separately called the main local truncation errors. $\bar{\varepsilon}_1, \bar{\varepsilon}_2$ and $\bar{\varepsilon}_3$ are bounds of convergence of exponentially fitted concurrent Milne's device.

Still, approximates of main local truncation error (18) applied, decide whether to accept final outcomes of iteration or settle to admit the answers of the present pace or redo iteration with smaller changing pace size. Process is veritably based on test as seen in (18). CheckAsher &Petzoid (1998),Dormand (1996),Faires& Burden (2012), Lambert (1973), Lambert (1991), Oghonyon et al. (2015), Oghonyon et al. (2016), Oghonyon et al. (2016), Oghonyon et al. (2018), Oghonyon et al. (2018) and Oghonyon et al. (2018)for more details. The main local truncation errors (18) is the bounds of convergence of the exponentially fitted concurrent Milne's, (estimate) for adjusting to convergence.

2. RESULTS AND DISCUSSION

This section presents the Mathematica computer programming codes of the mathematical results implemented using the exponentially fitted concurrent Milne's device. The finished results provided is attained with the support of Mathematica 9 kernel to demonstrate effectiveness and preciseness. See attached for Table 1.

Two problems were studied and solve employing EFCMD at various bound of convergence;10^{-3},10^{-5},10^{-7}, 10^{-8},10^{-9} ,10^{-11} and 10^{-13}. SeeAnake et al. (2014),Jator (2010), Ngwane &Jator (2014), Ngwane &Jator (2015), Ngwane &Jator (2017) and Odekunle et al. (2014) for more specifics. A Mathematica computer programming codes based on exponentially fitted block Milne's device is spelt out using Mathematica 9 kernel. This Mathematica kernel programming is carried out in a block by block mode as prescribed by the exponentially fitted block Milne's device. See appendix for EFCMD1 and EFCMD2

Problem 1: Consider the initial value ODE

$y'' + \omega y = 0,\ y(0) = 1, \quad y'(0)2, \quad \omega = 2.$

Exact Solution:$y(x) = \cos 2x + \sin 2x$.

Problem 2: Consider the nonlinear Duffing equation:

$y'' + y + y^3 = B\cos(\Omega x),$

$y(0) = C_0, \quad y'(0) = 0.$

ExactSolution: $y(x) = C_1 \cos(\Omega x) + C_2 \cos(3\Omega x) + C_3 \cos(5\Omega x) + C_4 \cos(7\Omega x),$

where $\Omega = 1.01, \quad B = 2 \times 10^{-3}, \quad C_0 = 0.200426728069, C_1 = 0.200179477536,$ $C_2 = 0.246946143 \times 10^{-3},\ C_3 = 0.304016 \times 10^{-6}$and $C_4 = 0.374 \times 10^{-9}$. Choose $w = 1.01$.

Table 1. [Table 1 establishesthe mathematical final answers of problem 1 and 2 employing EFCMD comparable to existence method]. Language stated on table 1 is delivered beneath.

T_{emd}	Max_{errs}	B_{cov}
A(α)-S	$1.32000e-05$	10^{-5}
EFCMD	$9.88520e-06$	10^{-5}
EFCMD	$9.88538e-06$	
EFCMD	$1.00659e-05$	
A(α)-S	$1.08990e-08$	10^{-8}
EFCMD	$4.89817e-09$	10^{-8}
EFCMD	$4.93823e-09$	
EFCMD	$4.97847e-09$	
A(α)-S	$4.32380e-11$	10^{-11}
EFCMD	$8.16480e-12$	10^{-11}
EFCMD	$1.68101e-11$	
EFCMD	$4.32290e-11$	
FSBP-BCM	$3.397282e-13$	10^{-13}
FSBP-BCM	$4.541350e-13$	
FSBP-BCM	$5.193623e-13$	
FSBP-BCM	$5.194734e-13$	
EFCMD	$8.282260e-14$	10^{-13}
EFCMD	$8.126830e-14$	
EFCMD	$9.592330e-14$	
TSDM	$3.30000e-03$	10^{-3}
BHTFM	$1.30000e-03$	
EFCMD	$6.49239e-04$	10^{-3}
EFCMD	$6.49369e-04$	
EFCMD	$6.62265e-04$	

BHT	7.70000e − 05	10^{-5}
BHTFM	5.60000e − 05	
BHTRKKNM	7.52000e − 05	
HLMMs	1.18000e − 05	
TSDM	6.40000e − 05	
EFCMD	6.50386e − 06	10^{-5}
EFCMD	650516e − 06	
EFCMD	6.63459e − 06	
BHTFM	1.40000e − 07	10^{-7}
BHTRKKNM	1.34000e − 07	
HLMMs	4.98000e − 07	
TSDM	1.00000e − 07	
EFCMD	6.50398e − 08	10^{-7}
EFCMD	6.50528e − 08	
EFCMD	6.63471e − 08	
BHTRKKNM	8.11000e − 09	10^{-9}
EFCMD	6.50398e − 10	10^{-9}
EFCMD	6.50527e − 10	
EFCMD	6.63470e − 10	
EFCMD	6.50402e − 12	10^{-11}
EFCMD	6.50538e − 12	
EFCMD	6.63494e − 12	
EFCMD	6.48648e − 14	10^{-13}
EFCMD	6.48925e − 14	
EFCMD	6.64746e − 14	

EFCMD: errors in EFCMD (Mathematica computer programming codes of exponentially fitted concurrent Milne's device) for tested problems 1 and 2.

T_{emd}: technique employed.

Max_{errs}: magnitude of the maximum errors in EFCMD.

B_{cOV}: bound of convergence.

A(α)-S: errors in A(α)-S (an A(α)-stable method for solving initial value problems of Ordinary differential equations) for tested

Problem 1. (SeeAnake et al., 2014).

BHT: errors in BHT (block hybrid trigonometrically fitted of $\delta = 10^{-6}$) for Tested problem 2. (SeeNgwane &Jator, 2015).

BHTFM: errors in BHTFM (block hybrid trigonometrically fitted method) for numerical tested problem 2. (SeeNgwane & Jator, 2013).

BHTRKNM: errors in BHTRKNM (block hybrid trigonometrically fitted Runge-Kutta-Nystrom method of $\delta = 10^{-6}$) for tested

Problem 2. (SeeNgwane &Jator, 2017)). FSBP-BCM errors in FSBP-BCM (five steps block predictor-block correc to rmethod for the solution of $y'' = f(x, y, y')$ fortested

Problem 1. (SeeOdekunle et al., 2014).HLMMs: errors in HLMMs (hybrid linear multistep methods) for tested problem 2. (See Jator, 2010).

TSDM: errors in TSDM (trigonometrically-fitted second derivative method) for tested

problem 2. (SeeNgwane &Jator, 2014).

Spelt algorithm for designing new pace size and evaluate magnitude of maximum errors using Mathematica kernel programming codes of concurrent Milne's device is been prescribed as follows:

Step 1: Choose the step size (h) for computing the methods.

Step 2: Same order of the concurrent predictor-corrector joint pair must be the similar.

Step 3: Step number of concurrent predictor method must be one step greater than concurrent corrector system.

Step 4: Define bound of convergence of the EFCMD.

Step 5: Input the EFCMD in Mathematica kernel 9

Step 6: Adopt Taylor's series method to generate the required initializations, otherwise avoid step 6 the move forward to step 7.

Step 7: Execute EFCMD in Mathematica

Kernel 9.

Step 8: If step 7 fails to converge, use this formula stated below to decide the appropriate step size for h to arrive at convergence and if not proceed to step 9.

$$qh = \left| \frac{\bar{\varepsilon}_1}{2\left(\tilde{\tilde{C}}_{p+6}^{[1]} -, \bar{\bar{C}}_{p+6}^{[1]}\right)} \right|^{\frac{1}{4}}.$$

Step 9: Evaluate magnitude of maximum errors after bound of convergence is attained

Step 10: Output maximum errors.

3. CONCLUSION

The mathematical final results have displayed the EFCMD is fulfilled with support of bound of convergence, designing a suitable step size and changing the step size. This convergence criteria decides either to accept or reject the iterations. Thus, this institute the performance of the EFCMD is discovered to yield an improve maximum errors than A(α)-S, BHT, BHMTB, BHTRKNM, FSBP-BCM, HLMMs and TSDM at all tried out convergence criteria of 10^{-3}, 10^{-5}, 10^{-7}, 10^{-8}, 10^{-9} ,10^{-11} and 10^{-13}as mentioned in Anake et al. (2014),Jator (2010), Ngwane &Jator (2014), Ngwane &Jator (2015), Ngwane &Jator (2017) and Odekunle et al. (2014). Therefore, it can be concluded that the developed Mathematica computer programming codes of exponentially fitted block Milne's device is suitable for solving special problems compared to the existing methods which solved problems using fixed step size, and lack of convergence criteria. Further work will hope to focus on the reverse exponentially fitted block Milne's device.

ACKNOWLEDGMENTS

Covenant University is deeply acknowledged for providing various supports which include the research grants.

REFERENCES

[1] Abell, M.L. &Braselton J.P. 2009. Mathematica By Example. 4th Edition. USA: Elsevier.

[2] Adejumo, G. Abioye, M.T.&Anake, T.A. 2014. Adoption of computer assisted language learning Software among Nigerian secondary school," EDULEARN14 Proceedings. 950-955.

[3] Akinfenwa, A.O.,Jator, S.N. & Yao, N.M. 2013. Continuous block backward differentiationformula for solving stiff ordinary differential equations. Computers andMathematicswith Applications 65(2013): 996-1005.

[4] Anake, T.A.Awoyemi D.O. & Adesanya, A.O.2012. One-step Implicit Hybrid Block Method for the Direct Solution of General Second Order ODES. IAENG International Journal of Applied Mathematics42(4): 224-228.

[5] Anake, T.A. &Adoghe, L.O. 2013. A Four Point Integration Method for the Solutions of IVP inODE. Australian Journal of Basic and Applied Sciences7(10): 467-473.

[6] Anake, T.A., Bishop, S.A., Adesanya, A.O. &Agarana, M.C.2014. An A(α)-stable method forsolving initial value problems of ordinary differential equations. Advances in Differential Equations and Control Processes 13(2014) 21-35.

[7] Ascher, U.M. &Petzoid, L.R.1998. Computer methods for ordinary differential equations and differential-algebraicequations. USA: SIAM.

[8] Bond, M.2009. Convolutions and the Weierstrassapproximation theorem. Michigan StateUniversity, USA.

[9] Calvo, M.Montijano J.I & Van,Daele, M. 2015. Exponentially fitted fifth-order two-step peerexplicit methods. AIP Conference Proceedings1648: (1) 1-4.

[10] D'Ambrosio, R. Esposito, E. & Paternoster, B.2011. Exponentially fitted two-step hybrid methodsfor y^''=f(x,y).Journal of Computational and Applied Mathematics235(16):4888-4897.

[11] D'Ambrosio, R. Esposito, E. & Paternoster, B.2012. Exponentially fitted two-StepRunge-Kuttamethods: construction and parameter selection. Applied Mathematics and

[12] Computation. 218(14): 7468-7480. Dormand, J.R.1996. Numerical methods for differential equations. London: CRC Press.

[13] Edeki, S.O., Ugbebor, O.O. &Owoloko, E. A. 2015. Analytical solutions of the Black-Scholes pricing model for European Option Valuation via a projected differential transformation method. Entropy. 17(11): 7510-7521.

[14] Eke, K. S. Oghonyon, J.G. &Davwaz, B. 2018. Some fixed point theorems for contractive maps in fuzzy G-partial metric spaces. International Journal of Mechanical Engineering and Technology. 9(8): 635-645.

[15] Eke, K.S., Imaga, O.F. &Odetunmibi, O.A. 2017. Convergence and stability of some modified iterative processes for a class of generalized contractive-like operators. IAENG International Journal of Computer Science. 44(4): 587-590.

[16] Faires, J.D. &Burden, R.L.2012. Initial-value problems for ODEs. USA: Brooks Cole. Gautschi, W.1961. Numerical integration of ordinary differential equations based ontrigonometric polynomials. Numerical Mathematics3(1961): 381-397.

[17] Ibrahim, Z.B. Othman, K.I. &Suleiman M. 2007. Implicit r-point block backward differentiationformula for solving first-order stiff ODEs. Applied Mathematics andComputation

[18] 186(1): 558-565.

[19] Jator, S.N.2010. On a class of hybrid methods for y^"=(x,y,y^'). International Journal of Pure and Applied Mathematics59(4): 381-395.

[20] Jikantoro, Y.D., Ismail, F.I. &Senu, N. 2015.Zero-dissipative trigonometrically fitted hybrid method for numerical solution of oscillatory problems. SainsMalaysiana. 44(3): 473-482.

[21] Ken, Y.L., Ismail, I.F. &Suleiman, M.2011. Block methods for special second order ODEs. USA:Lambert Academic Publishing.

[22] Lambert, J.D. 1973. Computational methods in ordinary differential equations. New York: JohnWiley & Sons.

[23] Lambert, J.D.1991. Numerical Methods for Ordinary Differential Systems. New York: John Wiley & Sons.

[24] Majid, Z.A. &Suleiman, M.B.2007. Implementation of four-point fully implicit block method forsolving ordinary differential equations. Applied Mathematics and Computation.184(2): 514-522.

[25] Majid, Z.A. &Suleiman, M.2008. Parallel direct integration variable step block method for solving large system of higher Order ODEs. World Academy of Science,Engineering and

[26] Technology40(2008): 71-75.

[27] Majid, Z.A., Azmi, N.A. Suleiman, M. & Ibrahim, Z.B. 2012. Solving directly general third order ordinary differential equations using two-point four step block method. SainsMalaysiana. 41(5): 623-632.

[28] Mehrkanoon, S., Majid, Z.A. &Suleiman, M.A. 2010.Variable step implicit block multistepmethod for solving first-order ODEs. Journal of Comp. and Appl. Mathematics233(9): 2387-2394.

[29] Ngwane, F.F. &Jator, S.N.2013. Block hybrid method using trigonometric basis for initial valueproblems with oscillating solutions. Numerical Algorithm63(4): 713-725.

[30] Ngwane, F.F. &Jator, S.N.2013. Solving oscillatory problems using a block hybrid Trigonometrically fitted method with two off-step points.

[31] Electronic Journal ofDifferential Equations20(2013): 119-132. Ngwane, F.F. &Jator, S.N.2014. Trigonometrically-fittedsecond derivative method for oscillatory problems.SpringerPlus 3(2014): 1-11.

[32] Ngwane, F.F. &Jator, S.N.2015. Solving the telegraph and oscillatory differential equations by ablock hybrid trigonometrically fitted algorithm. Hindawi Publishing Corporation (2015): 1-15.

[33] Ngwane, F.F. &Jator, S.N.2017. A trigonometrically fitted block method for solving oscillatorysecond-order initial value problems and Hamiltonian systems.Hindawi Publishing Corporation(2017): 1-14.

[34] Odekunle, M.R.Egwurube, M.O. Adesanya, A.O.&Udo, M.O.2014. Five steps block predictor-block corrector method for the solution of y^" f(x,⟦y,y⟧^').Applied Mathematics 5(8): 1252-1266.

[35] Oghonyon, J.G., S.A. Okunuga& S.A. Bishop 2015. A 5-step block predictor and 4-step corrector methods for solving general second order ODEs. Global Journal of Pure and Applied Mathematics 11(5): 3847-3862.

[36] Oghonyon, J.G.N.A. Omoregbe& S.A. Bishop 2016. Implementing an order six implicit block multistep method for third order ODEs using variable step size approach. Global Journal of Pure and Applied Mathematics 12(2): 1635-1646.

[37] Oghonyon, J.G.Ehigie, J. Bishop,S.A. 2016. Investigating the convergence of some selected properties on block predictor-corrector methods and it's applications. Journal of Engineering and Applied Sciences 11(11): 2402-2408.

[38] Oghonyon, J.G. Adesanya, O.A, Akewe, H. Okagwe, H.I. 2018.Softcode of Muliti-processing Milne's device for estimating first-order ordinary differential equations. Asian Journal of Scientific Research 11(4): 553-559.

[39] Oghonyon, J.G. Imaga, O.F. Ogunniyi, P.O. 2018. The reversed estimation of variable step size implementation for solving nonstiff ordinary differential equations. International Journal of Civil Engineering and Technology 9(8): 332-340.

[40] Oghonyon, J.G. Agboola, O.O Ogunniyi, P. O.& Adesanya, A.O. (2018). Computing oscillating vibrations employing exponentially fitted block Milne's device. International Journal of Mechanical Engineering and Technology. 9(8): 1234-1243.

[41] Owoloko, E.A., Omoregbe, N.A., Azeta, A.A., Edeki, S.O. (2015). Analytical solution of the Schwartz-moon growth option model revisited. International Journal of Mathematical Analysis. 9(21-24): 1021-1029.

[42] Waeleh, N. 2011. A New algorithm for solving higer order IVPs of ODEs. Applied Mathematical Sciences, 5(56): 2795-2805.

[43] Waeleh, N. & Majid, Z.A. 2016. A 4-point block method for solving higher order ordinary differential equations directly. International journal of Mathematics and Mathematical Sciences 2016: 1-8.

APPENDIX

This appendix presents the Mathematica computer programming codes of exponentially fitted concurrent Milne's device (EFCMD)

Mathematica Computer Programming Codes of Concurrent Predictor System

```
Given second order differential equations
Exact Solutions
h=given values
v[n]=given values
t=values generated from v[n]
w= given values

g[1]=g[0]+h(g'[0])+(h^2/2)g''[0]+(h^3/6)g'''[0]+(h^4/24)g''''[0]
g[2]=g[1]+h(g'[v[n]])+(h^2/2)g''[v[n]]+(h^3/6)g'''[v[n]]+(h^4/24)g''''[v[n]]
g[3]=g[2]+h(g'[v[n]+h])+(h^2/2)g''[v[n]+h]+(h^3/6)g'''[v[n]+h]+(h^4/24)g''''[v[n]+h]
g[4]=g[3]+h(g'[v[n]+2h])+(h^2/2)g''[v[n]+2h]+(h^3/6)g'''[v[n]+2h]+(h^4/24)g''''[v[n]+2h]

t=v[n]+h
g[3]=2g[2]-g[1]+h^2((-1/w^3-1/(2w^2)-1/6w+25/24)g''[t]+(-1/12+2/w^3+1/w^2+1/3w)g''[t-v[n]]+(-
    1/w^3-1/(2w^2)-1/(6w)+5/24)+1/w^2)g''[t-v[n]+h]
t=v[n]+3h
g[5]=3g[3]-2g[2]+h^2((55/12+15/w^4-2/w^3-2/w^2-4/(3w))g''[t]+(-13/6-
    30/w^4+4/w^3+4/w^2+8/(3w))g''[t-v[n]]+(23/12+15/w^4-2/w^3-2/w^2-4/(3w))g''[t-v[n]+h])
t=v[n]+5h
g[7]=4g[4]-3g[3]+h^2((80/w^4-3/w^3-9/(2w^2)-9/(2w)+97/8)g''[t]+(-33/4-
    160/w^4+6/w^3+9/w^2+9/w)g''[t-v[n]]+(80/w^4-3/w^3-9/(2w^2)-9/(2w)+53/8)g''[t-v[n]+h])

t=v[n]+4h
g[6]=2g[5]-g[4]+h^2((-1/w^3-1/(2w^2)-1/6w+25/24)g''[t]+(-1/12+2/w^3+1/w^2+1/3w)g''[t-v[n]]+(-
    1/w^3-1/(2w^2)-1/(6w)+5/24)+1/w^2)g''[t-v[n]+h]
t=v[n]+6h
g[8]=3g[6]-2g[5]+h^2((55/12+15/w^4-2/w^3-2/w^2-4/(3w))g''[t]+(-13/6-
    30/w^4+4/w^3+4/w^2+8/(3w))g''[t-v[n]]+(23/12+15/w^4-2/w^3-2/w^2-4/(3w))g''[t-v[n]+h])
t=v[n]+8h
g[10]=4g[7]-3g[6]+h^2((80/w^4-3/w^3-9/(2w^2)-9/(2w)+97/8)g''[t]+(-33/4-
    160/w^4+6/w^3+9/w^2+9/w)g''[t-v[n]]+(80/w^4-3/w^3-9/(2w^2)-9/(2w)+53/8)g''[t-v[n]+h])

t=v[n]+7h
g[9]=2g[8]-g[7]+h^2((-1/w^3-1/(2w^2)-1/6w+25/24)g''[t]+(-1/12+2/w^3+1/w^2+1/3w)g''[t-v[n]]+(-
    1/w^3-1/(2w^2)-1/(6w)+5/24)+1/w^2)g''[t-v[n]+h]
t=v[n]+9
g[11]=3g[9]-2g[8]+h^2((55/12+15/w^4-2/w^3-2/w^2-4/(3w))g''[t]+(-13/6-
    30/w^4+4/w^3+4/w^2+8/(3w))g''[t-v[n]]+(23/12+15/w^4-2/w^3-2/w^2-4/(3w))g''[t-v[n]+h])
t=v[n]+11h
g[13]=4g[10]-3g[9]+h^2((80/w^4-3/w^3-9/(2w^2)-9/(2w)+97/8)g''[t]+(-33/4-
    160/w^4+6/w^3+9/w^2+9/w)g''[t-v[n]]+(80/w^4-3/w^3-9/(2w^2)-9/(2w)+53/8)g''[t-v[n]+h])
```

Mathematica Computer Programming Codes of Concurrent Corrector System (EFCMD2)

Given second order differential equations

```
Exact Solution
h=given values
x[n]=given values
u= values generated from x[n]
w=given values

y[1]=y[0]+h(y'[0])+(h^2/2)y''[0]+(h^3/6)y'''[0]+(h^4/24)y''''[0]
y[2]=y[1]+h(y'[x[n]])+(h^2/2)y''[x[n]]+(h^3/6)y'''[x[n]]+(h^4/24)y''''[x[n]]
y[3]=y[2]+h(y'[x[n]+h])+(h^2/2)y''[x[n]+h]+(h^3/6)y'''[x[n]+h]+(h^4/24)y''''[x[n]+h]
y[4]=y[3]+h(y'[x[n]+2h])+(h^2/2)y''[x[n]+2h]+(h^3/6)y'''[x[n]+2h]+(h^4/24)y''''[x[n]+2h]

u=x[n]+h
y[3]=2y[2]-y[1]+h^2((-1/w^3-1/(2w^2)-1/(6w)+73/24)y''[u+x[n]]+(17/12-
    2/w^3+1/w^2+1/(3w))y''[u+x[n]+h]+(-1/w^3-1/(2w^2)-1/(6w)+25/24)y''[u+x[n]+2h])
u=x[n]+3h
y[5]=3y[3]-2y[2]+h^2((79/12+15/w^4-2/w^3-2/w^2-4/(3w))y''[u+x[n]]+(23/6-30/w^4-
    4/w^3+4/w^2+8/(3w))y''[u+x[n]+h]+(19/12+15/w^4-2/w^3-2/w^2-4/(3w))y''[u+x[n]+2h])
u=x[n]+5h
y[7]=4y[4]-3y[3]+h^2((80/w^4-3/w^3-9/(2w^2)-9/(2w)+65/8)y''[u+x[n]]+(45/4-160/w^4-
    6/w^3+9/w^2+9/w)y''[u+x[n]+h]+(80/w^4-3/w^3-9/(2w^2)-9/(2w)+1/8)y''[u+x[n]+2h])

u=x[n]+4h
y[6]=2y[5]-y[4]+h^2((-1/w^3-1/(2w^2)-1/(6w)+73/24)y''[u+x[n]]+(17/12-
    2/w^3+1/w^2+1/(3w))y''[u+x[n]+h]+(-1/w^3-1/(2w^2)-1/(6w)+25/24)y''[u+x[n]+2h])
u=x[n]+6h
y[8]=3y[6]-2y[5]+h^2((79/12+15/w^4-2/w^3-2/w^2-4/(3w))y''[u+x[n]]+(23/6-30/w^4-
    4/w^3+4/w^2+8/(3w))y''[u+x[n]+h]+(19/12+15/w^4-2/w^3-2/w^2-4/(3w))y''[u+x[n]+2h])
u=x[n]+8h
y[10]=4y[7]-3y[6]+h^2((80/w^4-3/w^3-9/(2w^2)-9/(2w)+65/8)y''[u+x[n]]+(45/4-160/w^4-
    6/w^3+9/w^2+9/w)y''[u+x[n]+h]+(80/w^4-3/w^3-9/(2w^2)-9/(2w)+1/8)y''[u+x[n]+2h])

u=x[n]+7h
y[9]=2y[8]-y[7]+h^2((-1/w^3-1/(2w^2)-1/(6w)+73/24)y''[u+x[n]]+(17/12-
    2/w^3+1/w^2+1/(3w))y''[u+x[n]+h]+(-1/w^3-1/(2w^2)-1/(6w)+25/24)y''[u+x[n]+2h])
u=x[n]+9h
y[11]=3y[9]-2y[8]+h^2((79/12+15/w^4-2/w^3-2/w^2-4/(3w))y''[u+x[n]]+(23/6-30/w^4-
    4/w^3+4/w^2+8/(3w))y''[u+x[n]+h]+(19/12+15/w^4-2/w^3-2/w^2-4/(3w))y''[u+x[n]+2h])
u=x[n]+11h
y[13]=4y[10]-3y[9]+h^2((80/w^4-3/w^3-9/(2w^2)-9/(2w)+65/8)y''[u+x[n]]+(45/4-160/w^4-
    6/w^3+9/w^2+9/w)y''[u+x[n]+h]+(80/w^4-3/w^3-9/(2w^2)-9/(2w)+1/8)y''[u+x[n]+2h])
```