Softcodes of Parallel Processing Milne's Device via Exponentially Fitted Method for Valuating Special ODEs

J. G. Oghonyon, Member, IAENG, M. R. Odekunle, A. O. Adesanya, O. F. Imaga

Abstract- **The idea of technological computing has immensely assisted to enhance accuracy and maximize computed errors involving computational math. Softcodes computer programme is guided towards supplying comfortable computation, proficiency and faster results at all times. The objective of this study will be to devise softcodes of parallel processing Milne's device (SPPMD) via exponentially fitted method for valuating special ordinary differential equations. This is established through collocation and interpolation of the exponentially fitted method. Dissecting (SPPMD) produces the principal local truncation error (PLTE) after expressing the order of SPPMD leading to the boundary of convergence. Some selected examples of special ODEs were tested to show the efficiency and accuracy of (SPPMD) at different boundary of convergence. The finished results exist with the aid of (SPPMD). Computed results show that the (SPPMD) is more proficient compare to subsisting methods in terms of the work out max errors at all levels.**

Index Terms- **Softcodes, exponentially fitted method, boundary of convergence, Principal local truncation errors**

I. INTRODUCTION

Several computational methods for the direct consolidation of (1) exist and subsequently, authors have not been able to utilize the peculiar info concerning special ODEs. For instance, scholars will not take into consideration the vibrating or decomposing behavioral attributes of the exact solution. Thus, SPPMD is consider as one of the most effective technic for valuating special ODEs. See [15]. This research study sees special ODEs owning an exceptional character of the approximative solution been situated ahead of time. Such special problem is of the class [10]-[11], [14], [19], [26]

$$g' = z(u,g), \qquad g(u_0) = g_0, \;\; u\epsilon[c,\; d] \tag{1}$$

with vibrating exact solutions, where $z: R \times R^k \rightarrow R^k$, k is the proportion of the physical organization and $g, g_0 \in R^k$. See [27], [34].

Equation (1) possesses majorly alternating, increasing and decomposing solutions that are widely known in numerous areas, these includes, ambiance, biological sciences, Newtonian mechanics, celestial bodies/universe, quanta theory, control theory and electrical circuit. Technological computing of distinct technics has been laid down by exponentially fitted method with recognized frequence and having solution beforehand abounds in literatures. Observe [4]-[5], [14], [19], [24] is particularly reserve. Scholarly persons have projected and implemented (1) to give the sought after result. See [27], [35]. However, [26]-[28] carried out all math calculations employing electronic computer programming code in Matlab format. On another occasion, [10]-[11], [17], [29]-[30] concluded math application using a composed encrypt in Mathematica 10. 0 to showcase the efficiency and accuracy of the technics. However, [9] executed all numeric computations on a PC computing machine device initiated by running PYTHON. Nevertheless, several shortcomings were noticed from the bookmen's contribution listed supra. This admits; applying a set step size and lack of suit step size, lack of boundary of convergence to insure convergence of the method and cumbersome computation without error control. The motivation of this study stems from the need to design an exceptional SPPMD with known frequence of one as seen in [23], [34] thereby yielding better computed max errors at the least boundary of convergence.

From the gaps enlisted earlier, this research study apart from demonstrating the use of SPPMD comes with some computational benefits such as designing suitable step size, varying the step size, deciding the boundary of convergence to ascertain convergence and maximize error control. Hence, this constitute the primary objective of SPPMD as computing technics which is suitable to valuate vibrating problems. See [7], [12]-[13], [19]-[20], [32]-[33].

Theorem (Weierstrass Approximation Theorem)
Let $z: R \rightarrow R$ be continuous and 2π $-$periodic. Then for each $\varepsilon > 0$, there exists a trigonometric polynomial $P(u) = \sum_{j=-n}^{k} c_j\, e^{iju}$ such that for all $u, |z(u) - P(u)| < \varepsilon$. Tantamountly, as for any such f, there must exist a successive

Manuscript submitted March 30, 2018; revised for submission April 7, 2018. This work was supported by Covenant University Centre for Research, Innovation and Discovery (CUCRID, Ota, Ogun State, Nigeria.

J. G. Oghonyon, Department of Mathematics, College of Science and Technology, Covenant University, P.M.B. 1023, Ota, Ogun State, Nigeria (corresponding author phone no.: +234-8139724200; godwin.oghonyon@covenantuniversity.edu.ng).

M. R. Odekunle, Department of Mathematics, Modibbo Adama University OF Technology, Yola, Nigeria (remi_odekunle@yahoo.com).

O. A. Adesanya, Department of Mathematics, Modibbo Adama University of Technology, Yola, Nigeria (torlar10@yahoo.com).

I. F. Imaga, Department of Mathematics, College of Science and Technology, Covenant University, P. M. B. 1023, Ota, Ogun State, Nigeria (Imaga.ogbu@covenantunivesity.edu.ng).

polynomial such that $P_n \to z$ in a uniform manner on R. See [8].

The residuary of this research study is as complies: in Subsection 2 SPPMD of Materials and Methods; in Subsection 3 Examples of Vibrating Problems; in Subsection 4 Computational Results and Discussion; in Subsection 5 Conclusion as cited [3], [32]-[33].

II. MATERIALS AND METHODS

Under this subsection, the objective to be attained will be to invent SPPMD. Parallel processing Milne's device is a conjugation of $v - step$ parallel processing predictor scheme (PPPS) and $v - 1 - step$ parallel processing corrector scheme (PPCS) of ilk order. This pair is defined as

$$g(u) = \sum_{i=0}^{k} \alpha_i\, g_{n-i} + h \sum_{i=0}^{k} \beta_i(u) z_{n-i}, \qquad (2)$$

$$g(u) = \sum_{i=0}^{k} \alpha_i\, g_{n-i} + \sum_{i=1}^{k} \boldsymbol{\beta}_i^*(u) z_{n+i}. \qquad (3)$$

Par (2) and (3) gives the SPPMD with $u = wh$, $\beta_i(u)$, $i = 0, 1, 2$ containing characteristics that bank on suited step size, changing step size and frequence. Mentioning that g_{n+j} is the numeric estimate to the precise solutions $g(u_{n+i})$ i.e. $g(u_{n+i}) \approx g_{n+i}$, and $z(u_{n+i}, g_{n+i}) \approx z_{n+i}$ owning $i = 0, 1, 2$. In order to arrive at par (2) and (3), the exponentially fitted method is employed in concert with the collocating /interpolating scheme to estimate the precise solution $g(u)$ on time interval of u_{n-1} for PPPS. Again, PPCS utilizes u_{n-2} for PPCS via the interpolating subprogram of the type (4)

$$g(u) = \sum_{i=0}^{k} a_i \left(\frac{u-u_n}{h}\right)^i + \sum_{i=0}^{1} \frac{e^{wu}}{i!}. \qquad (4)$$

Rewriting (4) give birth to the softcodes of exponentially fitted method (SEFM) of the form

$$g[u\_] = a[0] + a[1] + a[2]\frac{(u-u[n])^2}{h^2} + +a[3]\left(1 + \frac{w(u-u[n])}{h} + \frac{w^2(u-u[n])^2}{2h^2} + \frac{w^3(u-u[n])^3}{6h^3}\right), \qquad (5)$$

since w is always given, a_0, a_1, a_2 and a_3 are unchanging parameters needed to ascertain in a particular manner. Presuming the condition that equation (5) matches the precise solution at some picked out interval u_n, u_{n-2} to give the approximation as

$$g(u_n) \approx g_n, \qquad g(u_{n-2}) \approx g_{n-2}. \qquad (6)$$

Demanding that the interpolating function (6) gratifies par (1) at the points $u_{n+i}, i = 0, 1, 2, 3$ to bring forth the coming estimation of PPPS as

$$g'(u_{n-i}) \approx z_{n-i}, \qquad i = 0, 1, 2, \qquad (7)$$

while PPCS is stated as

$$g'(u_{n-2}) \approx z_{n-2},\ g'(u_{n+i}) \approx z_{n+i}, \quad i = 1, 2, 3. \ (8)$$

Uniting the estimation of (6), (7) and (8) will produce the four-fold systems of equation in Au=b pattern.

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

$$b = \{g[n], z[n-1], z[n-2], z[n-3]\};$$
$$\{x, t, l, q\} = Inverse[matrixa].b, \qquad (9)$$

$$matrixa = \left\{\begin{array}{c} \left\{1,-2,4,1 - 2w^2 - \dfrac{4w^3}{3}\right\}, \\ \left\{0,1,2, w + w^2 + \dfrac{w^3}{2}\right\}, \\ \{0,1,4, w + 2w^2 + 2w^3\}, \\ \left\{0,1,6, w + 3w^2 + \dfrac{9w^3}{2}\right\} \\ \\ \end{array}\right\};$$

$$b = \{g[n-2], z[n+1], z[n+2], z[n+3]\};$$
$$\{x, t, l, q\} = Inverse[matrixa].b, \quad (10)$$

Working out the systems of equation will develop the softcodes of PPPS and PPCS for solving the systems of equation. Finding $a_i, i = 0, 1, 2, 3$ and subbing the measures of $a_i{}'s$ into (5) will generate the continuous SPPMD for PPPS and PPCS as

$$g[u\_] = (1)g[n] + \left(-\frac{1}{w^3} - \frac{(2w-6w^3)}{(2w^3)}\frac{(u-u[n])^1}{h} - \frac{\left(w^2 - \frac{(5w^3)}{2}\right)}{(2w^3)}\frac{(u-u[n])^2}{h^2} + \left(\frac{1}{w^3}\right)\frac{(u-u[n])^3}{h^3}\right) f[n-1]h + \left(\frac{2}{w^3} - \frac{(-4w+6w^3)}{(2w^3)}\frac{(u-u[n])^1}{h} - \frac{(-2w^2+4w^3)}{(2w^3)}\frac{(u-u[n])^2}{h^2} - \left(\frac{2}{w^3}\right)\frac{(u-u[n])^3}{h^3}\right) f[n-2]h + \left(-\frac{1}{w^3} - \frac{(2w-2w^3)}{(2w^3)}\frac{(u-u[n])^1}{h} - \frac{\left(-w^2 - \frac{3w^3}{2}\right)}{(2w^3)}\frac{(u-u[n])^2}{h^2} + \left(\frac{1}{w^3}\right)\frac{(u-u[n])^3}{h^3}\right) f[n-3]h,$$
(11)

$$\begin{aligned} g[u\_] = (1)g[n-2] \\ &+ \left(\left(\frac{-2+\frac{74w^3}{3}}{2w^3}\right)\right. \\ &- \frac{(-2w+6w^3)}{(2w^3)}\frac{(u-u[n])^1}{h} \\ &- \frac{\left(w^2-\frac{(5w^3)}{2}\right)}{(2w^3)}\frac{(u-u[n])^2}{h^2} \\ &\left. + \left(\frac{1}{w^3}\right)\frac{(u-u[n])^3}{h^3}\right) f[n+1]h \\ &+ \left(\left(\frac{4-\frac{100w^3}{3}}{2w^3}\right)\right. \\ &+ \frac{(4w-6w^3)}{(2w^3)}\frac{(u-u[n])^1}{h} \\ &+ \frac{(2w^2+4w^3)}{(2w^3)}\frac{(u-u[n])^2}{h^2} \\ &\left. - \left(\frac{2}{w^3}\right)\frac{(u-u[n])^3}{h^3}\right) f[n+2]h \\ + \left(\left(\frac{-2+\frac{38w^3}{3}}{2w^3}\right) - \frac{(-2w+2w^3)}{(2w^3)}\frac{(u-u[n])^1}{h}\right. \\ &+ \frac{\left(-w^2-\frac{3w^3}{2}\right)}{(2w^3)}\frac{(u-u[n])^2}{h^2} \\ &\left. + \left(\frac{1}{w^3}\right)\frac{(u-u[n])^3}{h^3}\right) f[n+3]h \\ . \qquad \qquad (12) \end{aligned}$$

Valuating the continuous PPPS and PPCS (11) and (12) at some selected points of $x_{n+i},\ i = 1, 2, 3$ will bring forth the PPPS and PPPCS as

$g[u\_] = g[n] + h(\beta_0(w,u)z[n-1] + \beta_1(w,u)z[n-2] + \beta_2(w,x)z[n-3]),$ (13)

$g[u\_] = g[n-2] + h(\beta_0(w,u)z[n+1] + \beta_1(w,u)z[n+2] + \beta_2(w,u)z[n+3]),$ (14)

where w is the known frequence, $\beta_0(w,u), \beta_1(w,u)$ and $\beta_2(w,u)$ are unchanging constants. Find [1]-[2], [13], [26]-[30], [32]-[33] for more items.

Inventing the convergence boundary of SPPMD

To propel the SPPMD, the $v-step$ PPPS and $v-1-step$ PPCS are processed as PPPS-PPCS pair possessing the ilk order. A compendium of [7], [12]-[13], [19]-[20], [32]-[33] shows that it is executable to look for the approximate of principal local truncation error of the PPPS-PPPS pair in absence of calculating higher differential constants of $g(u)$. Make bold that $\tilde{p} = \bar{p}$, where $\bar{p}$ and $\tilde{p}$ establishes the order of the PPPS and PPCS. Straightaway, for a scheme of order $\tilde{p}$, the investigation of $v-step$ PPPS will give birth to the principal local truncation errors as

$\tilde{C}^{[1]}_{\tilde{p}+4}h^{\tilde{p}+4}g^{(\tilde{p}+4)}(\tilde{u}_n) =$
$g(u_{n+1}) - g^{[l_1]}_{n+1} - \left(\frac{-112+44w+47w^2}{24w^2}\right) + O(h^{\tilde{p}+5}),$
$\tilde{C}^{[2]}_{\tilde{p}+4}h^{\tilde{p}+4}g^{(\tilde{p}+4)}(\tilde{u}_n) = g(u_{n+2}) - g^{[l_2]}_{n+2} + \left(\frac{19}{3} + \frac{14}{w\text{^}3} - \frac{4}{w^2} -$
$\left.\frac{4}{w}\right) + O(h^{\tilde{p}+5}),$ (15)
$\tilde{C}^{[3]}_{\tilde{p}+4}h^{\tilde{p}+4}g^{(\tilde{p}+4)}(\tilde{u}_n) =$
$g(u_{n+3}) - g^{[l_3]}_{n+3} - \left(\frac{1924-222w-333w^2+567w\text{^}3}{36w^3}\right) + O(h^{\tilde{p}+5}).$

A like computing analysis of $v-1-step$ PPCS will generate the principal local truncation errors as

$\bar{C}^{[1]}_{\bar{p}+5}h^{\bar{p}+5}g^{(\bar{p}+5)}(\bar{x}_n) =$
$g(t_{n+1}) - g^{[q_1]}_{n+1} + \left(\frac{-1134+426w+375w^2-3115w^3}{36w^3}\right) + O(h^{\bar{p}+5}),$

$\bar{C}^{[2]}_{\bar{p}+5}h^{\bar{p}+5}g^{(\bar{p}+5)}(\bar{t}_n) =$
$g(t_{n+2}) - g^{[q_2]}_{n+2} + \left(\frac{-2(231-66w-66w^2+206w^3}{9w^3}\right) + O(h^{\bar{p}+5})$
(16)
$\bar{C}^{[3]}_{\bar{p}+5}h^{\bar{p}+5}g^{(\bar{p}+5)}(\bar{t}_n) = g(t_{n+3}) -\ g^{[q_3]}_{n+3} + \left(-\frac{1499}{72} - \frac{572}{4w^3} + \right.$
$\left.\frac{22}{w^2} + \frac{33}{w}\right) + O(h^{\bar{p}+5}),$

where $\tilde{C}^{[1]}_{\tilde{p}+4}, \tilde{C}^{[2]}_{\tilde{p}+4}, \tilde{C}^{[1]}_{\tilde{p}+4}\ , \bar{C}^{[1]}_{\bar{p}+4}, \bar{C}^{[2]}_{\bar{p}+4}$ and $\bar{C}^{[3]}_{\bar{p}+4}$ are in existent as a separate entity of the step-size h and $g(u)$ behave as the precise solution to the differential constant fulfiliing the starting pre-condition $g(u_n) \approx g_n$. See [7], [12]-[13], [19]-[20], [32]-[33] for more info.

To advance, the pre-condition is set small for h valuates to be attained as

$g^{(4)}(\tilde{u}_n) \approx g^{(4)}(\bar{u}_n),$ (17)

and executing the SPPMD relies immediately on this precondition put forward by (17).

Further simplification of the principal local truncation errors of (15) and (16) as well as dropping off terms of degree $O(h^{\bar{p}+5})$, it gets well to achieve the computation of the principal local truncation errors of SPPMD as

$\bar{C}^{[1]}_{\bar{p}+4}h^{\bar{p}+4}g^{(\bar{p}+4)}(\bar{u}_n) \approx \frac{31}{39}\left[g^{[l_1]}_{n+1} - g^{[q_1]}_{n+1}\right] < \sigma_1,$

$\bar{C}^{[2]}_{\bar{p}+4}h^{\bar{p}+4}g^{(\bar{p}+4)}(\bar{u}_n) \approx \frac{160}{321}\left[g^{[l_2]}_{n+2} - g^{[q_2]}_{n+2}\right] < \sigma_2,$ (18)
$\bar{C}^{[3]}_{\bar{p}+4}h^{\bar{p}+4}g^{(\bar{p}+4)}(\bar{u}_n) \approx \frac{1}{41}\left[g^{[l_3]}_{n+3} - g^{[q_3]}_{n+3}\right] < \sigma_3.$

Citing the statements that $g^{[l_1]}_{n+1} \neq g^{[q_1]}_{n+1}, g^{[l_2]}_{n+2} \neq g^{[q_2]}_{n+2}$ and $g^{[l_3]}_{n+3} \neq g^{[q_3]}_{n+3}$ are acknowledged as PPPS and PPCS estimates brought forth by the SPPMD of order p, while $\bar{C}^{[1]}_{\bar{p}+4}h^{\bar{p}+4}g^{(\bar{p}+4)}(\bar{u}_n)$, $\bar{C}^{[2]}_{\bar{p}+4}h^{\bar{p}+4}g^{(\bar{p}+4)}(\bar{u}_n)$ and $\bar{C}^{[3]}_{\bar{p}+4}h^{\bar{p}+4}g^{(\bar{p}+4)}(\bar{u}_n)$ are distinctly named the principal local truncation errors. σ_1, σ_2 and σ_3 are the boundaries of convergence of SPPMD.

Yet, the estimates of the principal local truncation error (18) is employed to make decision either to allow or discontinue the final results of the iterative process or repeat the successive process with a smaller varying step size. The procedure is truly acceptable on the basis of a try out test evaluation as defined by (18). Check [7], [12]-[13], [19]-[20], [32]-[33] for more details.

III. NUMERICAL EXAMPLES

Two problems are studied and solve employing SPPMD at distinct boundaries of convergence such as 10^{-4}, 10^{-6}, 10^{-8} and 10^{-10}. See [23], [34] for more details. A softcodes founded on SPPMD is composed applying computing software package. This SPPMD is carried out in a parallel processing style as prescribed by SPPMD. See appendix.

Problem 1: Consider the initial value ODE
$g' = -g,\ \ g(0) = 1,\ \ u \in [0, 20]$.
Exact Solution: $g(u) = e^{-u}$.
Problem 2: Consider the nonlinear Duffing equation:
$g' = 0.1(g - sinu) + cosu,\ \ g(0) = 0,\ \ u \in [0, 20]$.
Exact Solution: $g(u) = sinu$.

IV. RESULTS AND DISCUSSION

This subsection introduces the softcodes of the computational results executed employing the SPPMD. The finalized output supplied were achieved with the assistance of Mathematica 9 kernel to show-case the efficiency and preciseness. See [1]-[2].

Table 1 of problem 1

$\mathbf{T_{used}}$	$\mathbf{Max_{errs}}$	$\mathbf{BO_{convergence}}$
2BPC	$3.1515(-6)$	10^{-4}
1PVSO	$1.9434(-3)$	
3PS	$7.5511(-7)$	
4PS	$3.3456(-6)$	
SPPMD	$1.66769(-9)$	10^{-4}
SPPMD	$1.66819(-9)$	
SPPMD	$1.6727(-9)$	
2BPC	$2.8360(-8)$	10^{-6}
1PVSO	$1.2904(-7)$	
3PS	$4.9429(-9)$	
4PS	$2.0967(-8)$	
SPPMD	$1.55431(-15)$	10^{-6}
SPPMD	$1.55431(-15)$	
SPPMD	$1.66533(-3)$	
2BPC	$1.5336(-10)$	10^{-8}
1PVSO	$1.7610(-9)$	
3PS	$8.8436(-11)$	
4PS	$1.5070(-10)$	
SPPMD	$0.$	10^{-8}
SPPMD	$0.$	
SPPMD	$1.11022(-16)$	
2BPC	$1.4077(-12)$	10^{-10}
1PVSO	$1.2743(-11)$	
3PS	$1.8760(-12)$	
4PS	$6.1173(-13)$	
SPPMD	$0.$	10^{-10}
SPPMD	$0.$	
SPPMD	$0.$	

Table 2 of problem 2

$\mathbf{T_{used}}$	$\mathbf{Max_{errs}}$	$\mathbf{BO_{convergence}}$
2BPC	$1.2411(-4)$	10^{-4}
1PVSO	$9.4330(-4)$	
SPPMD	$1.66668(-9)$	10^{-4}
SPPMD	$1.66668(-9)$	
SPPMD	$1.66673(-9)$	
2BPC	$4.0421(-8)$	10^{-6}
1PVSO	$3.5795(-5)$	
SPPMD	$1.66668(-15)$	10^{-6}
SPPMD	$1.66671(-15)$	
SPPMD	$1.67168(-15)$	
2BPC	$8.1888(-9)$	10^{-8}
1PVSO	$2.9001(-7)$	
SPPMD	$1.64113(-21)$	10^{-8}
SPPMD	$1.64113(-21)$	
SPPMD	$1.69407(-21)$	

T_{used}	Max_{errs}	$BO_{convergence}$
2BPC	$6.4933(-11)$	10^{-10}
1PVSO	$9.7441(-10)$	
SPPMD	$8.27181(-25)$	10^{-10}
SPPMD	$8.27181(-25)$	
SPPMD	$8.27181(-25)$	

SPPMD: errors in SPPMD (Softcodes of parallel processing Milne's device) for tested problems 1 and 2.
T_{used}: technic used.
Max_{errs}: magnitude of the maximum errors of SPPMD.
C_{crit}: boundary of convergence.
2BPC: error in 2BPC (implementation of the two point predictor-corrector block method using variable step size) for tested problem 1 and 2. See [23].
1PVSO: errors in 1PVSO (implementation of the one point method variable step size and order using the integration coefficients. See [23].
3PS: errors in 3PS (the implementation of the three-step implicit block method for tested problem 1. See [34].
4PS: errors in 4PS (the implementation of the four-step implicit block method for tested problem 1. See [34].

A composed step by step technic that will enforce the SPPMD implementation and valuation of the maximum errors is been prescribed below as follows:

Stride 1: Choose the step size for h.
Stride 2: The order of the parallel processing predictor-corrector pair must be the alike.
Stride 3: The stride measure of the parallel processing Predictor method must be one stride above the parallel processing corrector method.
Stride 4: Define the boundary of convergence of the SPPMD.
Stride 5: Insert the SPPMD in any computing software package.
Stride 6: Adopt single stride technic to prime if necessary, if not, avoid stride 6 and move on to stride 7.
Stride 7: Carried out the implementation of SSPMD in computing software package.
Stride 8: If stride 7 is discontinued due to h value and boundary of convergence, adopt this new rule of generator stated below to find the true value of length h to achieve convergence, otherwise move on to stride 9.

$$qh = \left|\frac{\varepsilon_1}{2\left(\bar{C}^{[1]}_{p+6} - , \bar{C}^{[1]}_{p+6}\right)}\right|^{\frac{1}{4}}.$$

Stride 9: Valuate the magnitude of the maximum errors after the boundary of convergence is satisfied.
Stride 10: Put into writing the magnitude of the maximum errors. See [32].

V. CONCLUSION

The computed terminal outputs shown in Table 1 and Table 2 demonstrates the SPPMD is attained with the aid of the boundary of convergence, suited and varying stride size. This boundary of convergence helps to check whether the looping is allowed or disallowed. Therefore, establishes the efficiency of the SPPMD to obtain better maximum errors compare to 2BPC, 1PVSO, 3PS and 4PS in the least boundary of convergence of 10^{-4}, 10^{-6}, 10^{-8}, and 10^{-10} as cited [23], [34]. For this reason, it will satisfactory to resolve that the SPPMD is effective for computing special problem of frequence one as against [23], [34]. Further work will be to apply trigonometrically fitted parallel processing Milne's device on first order ODEs.

APPENDIX

The SPPMD for solving problem 1 and 2 is seen infra.

```
g[t_]=Exp[-t]
w=1
h= given value, x[n]=given starting value
t=given value
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'[x[n]])+(h^2/2)g"[x[n]]+(h^3/6)g"'[x[n]]+(h^4/24)g""[x[n]]
g[3]=g[2]+h(g'[x[n]+h])+(h^2/2)g"[x[n]+h]+(h^3/6)g"'[x[n]+h]+(h^4/24)g""[x[n]+h]
g[4]=g[3]+h(g'[x[n]+2h])+(h^2/2)g"[x[n]+2h]+(h^3/6)g"'[x[n]+2h]+(h^4/24)g""[x[n]+2h]
g[5]=g[4]+h(g'[x[n]+3h])+(h^2/2)g"[x[n]+3h]+(h^3/6)g"'[x[n]+3h]+(h^4/24)g""[x[n]+3h]

t=x[n]+2h
g[4]=g[3]+h((-(6+3w-37w^2)/(6w^2))g'[t-x[n]]+(-19/3+3/w)g'[t-x[n]+h]-(-(6+3w-13w^2)/(6w^2))g'[t-x[n]+2h])
t=x[n]+4h
g[6]=g[4]+h((155/12+7/w^3-2/w^2-2/w)g'[t-x[n]]+(-46/3-14/w^3+4/w^2+4/w)g'[t-x[n]+h]+(65/12+7/w^3-2/w^2-2/w)g'[t-x[n]+2h])
t=x[n]+6h
g[8]=g[5]+h((133/6+26/w^3-3/w^2-9/(2w))g'[t-x[n]]+(-85/3-52/w^3+6/w^2+9/w)g'[t-x[n]+h]+(61/6+26/w^3-3/w^2-9/(2w))g'[t-x[n]+2h])
t=x[n]+5h
g[7]=g[6]+h((-(6+3w-37w^2)/(6w^2))g'[t-x[n]]+(-19/3+3/w)g'[t-x[n]+h]-(-(6+3w-13w^2)/(6w^2))g'[t-x[n]+2h])
t=x[n]+7h
g[9]=g[7]+h((155/12+7/w^3-2/w^2-2/w)g'[t-x[n]]+(-46/3-14/w^3+4/w^2+4/w)g'[t-x[n]+h]+(65/12+7/w^3-2/w^2-2/w)g'[t-x[n]+2h])
t=x[n]+9h
g[11]=g[8]+h((133/6+26/w^3-3/w^2-9/(2w))g'[t-x[n]]+(-85/3-52/w^3+6/w^2+9/w)g'[t-x[n]+h]+(61/6+26/w^3-3/w^2-9/(2w))g'[t-x[n]+2h])

y[u_]=Exp[-u]
w=1
h=given value, x[n]=given starting value
u=given value
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]
y[5]=y[4]+h(y'[x[n]+3h])+(h^2/2)y"[x[n]+3h]+(h^3/6)y'"[x[n]+3h]+(
h^4/24)y""[x[n]+3h]

u=x[n]+2h
y[4]=y[1]+h((169/12-1/w^2-1/(2w))y'[u+x[n]]+(-
53/3+2/w^2+1/w)y'[u+x[n]+h]+(79/12-1/w^2-1/(2w))y'[u+x[n]+2h])
u=x[n]+4h
y[6]=y[2]+h((40/3+7/w^3-2/w^2-2/w)y'[u+x[n]]+(-44/3-
14/w^3+4/w^2+4/w)y'[u+x[n]+h]+(16/3+7/w^3-2/w^2-
2/w)y'[u+x[n]+2h])
u=x[n]+6h
y[8]=y[3]+h((121/12+26/w^3-3/w^2-9(2w))y'[u+x[n]]+(-23/3-
52/w^3+6/w^2+9/w)y'[u+x[n]+h]+(31/12+26/w^3-3/w^2-
9/(2w))y'[u+x[n]+2h])
u=x[n]+5h
y[7]=y[4]+h((169/12-1/w^2-1/(2w))y'[u+x[n]]+(-
53/3+2/w^2+1/w)y'[u+x[n]+h]+(79/12-1/w^2-1/(2w))y'[u+x[n]+2h])
u=x[n]+7h
y[9]=y[5]+h((40/3+7/w^3-2/w^2-2/w)y'[u+x[n]]+(-44/3-
14/w^3+4/w^2+4/w)y'[u+x[n]+h]+(16/3+7/w^3-2/w^2-
2/w)y'[u+x[n]+2h])
u=x[n]+9h
y[11]=y[6]+h((121/12+26/w^3-3/w^2-9(2w))y'[u+x[n]]+(-23/3-
52/w^3+6/w^2+9/w)y'[u+x[n]+h]+(31/12+26/w^3-3/w^2-
9/(2w))y'[u+x[n]+2h])
```

ACKNOWLEDGEMENTS

The authors would like to appreciate Covenant University for providing financial backing through grants throughout the study period of time.

REFERENCES

[1] M. L. Abell and J. P. Braselton, “Mathematica by Example,” 4th Edition, Elsevier, USA, 2009.

[2] G. Adejumo, M. T. Abioye and T. A. Anake, “Adoption of computer assisted language learning software among Nigerian secondary school,” EDULEARN14 Proceedings. 950-955.

[3] A. O. Akinfenwa, S. N. Jator and N. M. Yao, “Continuous block backward differentiation formula for solving stiff ordinary differential equations,” Computers and Mathematics with Applications. Vol. 65, No. 7, pp. 996-1005, April 2013.

[4] T. A. Anake, D. O. Awoyemi and A. O. Adesanya, “One-step implicit hybrid block method for the direct solution of general second order ODEs,” IAENG International Journal of Applied Mathematics. Vol 42, No. 4, pp. 224-228, 2012.

[5] T. A. Anake and L. O. Adoghe, “A four point integration method for the solutions of IVP in ODE,” Australian Journal of Basic and Applied Sciences. Vol 7, No. 10, pp. 467-473, 2013.

[6] T. A. Anake, S. A. Bishop, A. O. Adesanya and M. C. Agarana, “An A(α)-stable method for solving initial value problems of ordinary differential equations,” Advances in Differential Equations and Control Processes. Vol. 13, No. 1, pp. 21-35, May, 2014.

[7] U. M. Ascher and L. R. Petzoid, “Computer methods for ordinary differential equations and differential-algebraic equations,” SIAM, USA, 1984.

[8] M. Bond, “Convolutions and the Weierstrass approximation theorem, Department of Mathematics, Michigan State University, https://bondmatt.files.wordpress.com/2009/09/weierstrass2-01.pdf, 2009.

[9] M. Calvo, J. I. Montijano and M. Van Daele, “Exponentially fitted fifth-order two-step peer explicit methods,” AIP Conference Proceedings. Vol. 1648, No. 1, pp. 1-4, 2015.

[10] R. D’Ambrosio, E. Esposito, B. Paternoster, “Exponentially fitted two-step hybrid methods for y^"=f(x,y),” Journal of Computational and Applied Mathematics. Vol. 235, pp. 4888-4897, 2011.

[11] R. D’Ambrosio, E. Esposito and B. Paternoster, “Exponentially fitted two-step Runge-Kutta methods: construction and parameter selection,” Applied Mathematics and Computation. Vol. 218, pp. 7468-7480, 2012.

[12] J. R. Dormand, “Numerical methods for differential equations, A Computational Approach, London, 1996.

[13] J. D. Faires and R. L. Burden, “Initial-value problems for ODEs,”3rd Edition, Dublin City University, Brooks Cole, 2012.

[14] W. Gautschi, “Numerical integration of ordinary differential equations based on trigonometric polynomials,” Numerical Mathematics. Vol 3, pp. 381-397, 1961.

[15] D. Hollevoet, M. Van Daele, G. V. Berghe, “The optimal exponentially-fitted Numerov method for solving two-point boundary value problems,” Jounral of Computational and Applied Mathematics. Vol. 230, pp. 260-269, 2009.

[16] Z. B. Ibrahim, K. I. Othman and M. Suleiman, “Implicit r-point block backward differentiation formula for solving first-order stiff ODEs,” Applied Mathematics and Computation. Vol. 186, No. 1, pp. 558-565, March, 2007.

[17] S. N. Jator, “On a class of hybrid methods for y^"=(x,y,y^'),” International Journal of Pure and Applied Mathematics. Vol. 59, No. 4, pp. 381-395, January, 2010.

[18] Y. L. Ken, I. F. Ismail and M. Suleiman, “Block methods for special second order ODEs,” USA, 2011.

[19] J. D. Lambert, “Computational Methods in Ordinary Differential Equations,” New York, 1973.

[20] J. D. Lambert, “Numerical Methods for Ordinary Differential Systems, “New York, 1991.

[21] Z. A. Majid and M. B. Suleiman, “Implementation of four-point fully implicit block method for solving ordinary differential equations,” Applied Mathematics and Computation. Vol. 184, No. 2, pp. 514-522, January, 2007.

[22] Z. A. Majid, M. Suleiman, “Parallel direct integration variable step block method for solving large system of higher Order odes, World Academy of Science, Engineering and Technology. Vol 40, pp. 71-75, May, 2008.

[23] Z. A. Majid and M. Suleiman, “Predictor-corrector block iteration method for solving ordinary differential equations,” Sains Malaysiana. Vol. 40, pp. 659-664, June, 2011.

[24] J. Martin-Vaquero and J. Vigo-Aguiar, “Exponentially fitting BDF algorithms: Explicit and implicit 0-stable methods,” Journal of Computational and Applied Mathematics. Vol. 192, pp. 100-113, 2006.

[25] S. Mehrkanoon, Z. A. Majid and M. Suleiman, “A variable step implicit block multistep method for solving first-order ODEs,” Journal of Comp. and Appl. Mathematics. Vol. 233, No. 9, pp. 2387-2394, March, 2010.

[26] F. F. Ngwane and S. N. Jator, “Block hybrid method using trigonometric basis for initial value problems with oscillating solutions,” Numerical Algorithm. Vol. 63, pp. 713-725, October, 2013.

[27] F. F. Ngwane and S. N. Jator, “Solving oscillatory problems using a block hybrid trigonometrically fitted method with two off-step points, Electronic Journal of Differential Equations,” Conference on Differential Equations and Computational Simulations. Vol. 20, pp. 119-132, October, 2013.

[28] F. F. Ngwane and S. N. Jator, “Trigonometrically-fitted second derivative method for oscillatory problems,” SpringerPlus. Vol. 3, pp. 1-11, June, 2014.

[29] F. F. Ngwane and S. N. Jator, “Solving the telegraph and oscillatory differential equations by a block hybrid trigonometrically fitted algorithm,” Hindawi Publishing Corporation. Vol. 2015, pp. 1-15, October, 2015.

[30] F. F. Ngwane and S. N. Jator, “A trigonometrically fitted block method for solving oscillatory second-order initial value problems and Hamiltonian systems,” Hindawi Publishing Corporation. Vol. 2017, pp. 1-14, January, 2017.

[31] M. R. Odekunle, M. O. Egwurube, A. O. Adesanya and M. O. Udo, “Five steps block predictor-block corrector method for the solution of y^" f(x,〖 y,y〗 ^'),” Applied Mathematics. Vol. 5, pp. 1252-1266, 2014.

[32] J. G. Oghonyon, S. A. Okunuga and S. A. Iyase, “Milne’s implementation on block predictor-corrector methods,” Journal of Applied Sciences. Vol. 16, No. 5, pp. 236-241, April, 2016

[33] J. G. Oghonyon, S. A. Okunuga and S A Bishop, “A variable-step-size block predictor-corrector method for ODEs,” Asian Journal of Applied Sciences. Vol. 10, No. 2, pp. 96-101, March, 2017.

[34] H. M. Radzi, Z. A. Majid, F. Ismail and M. Suleiman, “Four step implicit block method of Runge-Kutta type for solving first order ODEs, IEEE. Vol. 2011, pp. 1-5, April, 2011.

[35] J. Vigo-Aguiar, and H. Ramos, “Dissipative Chebyshev exponential-fitted methods for numerical solution of second-order differential equations,” Journal of Computational and Applied Mathematics. Vol. 158 pp. 187-211, 2003.