International Journal of Theoretical and Applied Mathematics
2017; 3(2): 58-63
http://www.sciencepublishinggroup.com/j/ijtam
doi: 10.11648/j.ijtam.20170302.12

Development of a New One-Step Scheme for the Solution of Initial Value Problem (IVP) in Ordinary Differential Equations

Fadugba Sunday Emmanuel[1, *], Falodun Bidemi Olumide[2]

[1]Department of Mathematics, Ekiti State University, Ado Ekiti, Nigeria
[2]Department of Mathematics, University of Ilorin, Ilorin, Nigeria

Email address:
emmasfad2006@yahoo.com (F. S. Emmanuel)
[*]Corresponding author

To cite this article:
Fadugba Sunday Emmanuel, Falodun Bidemi Olumide. Development of a New One-Step Scheme for the Solution of Initial Value Problem (IVP) in Ordinary Differential Equations. *International Journal of Theoretical and Applied Mathematics.* Vol. 3, No. 2, 2017, pp. 58-63. doi: 10.11648/j.ijtam.20170302.12

Received: October 27, 2016; **Accepted**: January 16, 2017; **Published**: February 9, 2017

Abstract: In this paper, a new one-step scheme was developed for the solution of initial value problems of first order in ordinary differential equations. In its development a combination of interpolating function and Taylor series were used. The method was used for the solution of initial value problems emanated from real life situations. The numerical results showed that the new scheme is consistent, robust and efficient.

Keywords: Interpolating Function, Initial Value Problem, One-Step Method, Ordinary Differential Equation, Taylor Series

1. Introduction

In the past years, a large number of methods suitable for solving ordinary differential equations have been proposed. A major impetus to developing numerical procedures was the invention of calculus by Newton and Leibnitz, as this led to accurate mathematical models for physical reality, such as Sciences, Engineering, Medicine and Business. These mathematical models cannot be usually solved explicitly and numerical method to obtain approximate solutions is needed.

The approach for the solution of initial value problems in ordinary differential equations based on numerical approximations were developed before the existence of programmable computers.

A numerical method is a complete and unambiguous set of procedures for the solution of a problem, together with computable error estimates. The study and implementation of such methods is the province of numerical analysis [12].

Development of numerical integrator for the solution of initial value problems in ordinary differential equations has attracted the attention of many researchers in recent years. There are numerous methods that produce numerical approximations to solution of initial value problem in ordinary differential equation such as Euler's method which was the oldest and simplest method originated by Leonhard Euler in 1768, Improved Euler method, Runge Kutta methods described by Carl Runge and Martin Kutta in 1895 and 1905 respectively.

There are many excellent literature and exhaustive texts on this subject that may be consulted, such as [1], [3], [4], [5], [6], [7], [8], [9], [11], [12] just to mention a few.

In this paper we consider the initial value problem of the form

$$y' = f(x, y), y(a) = y_0, x \in [a, b], y \in \mathbb{R} \tag{1}$$

and develop an algorithm which can effectively solve initial value problems in ordinary differential equation. Many researchers have solved the problem in (1). However, if the solution to (1) posseses a singularity point, a numerical integration formulae will be more effective. In another development, [10] recently discussed one-step method of Euler-Maruyama type for the solution of stochastic differential equations using varying step sizes. [2] derived a continuous linear multistep method using Hermite polynomials as basis functions.

In this paper we develop a new accurate scheme for the solution of the initial value problems in ordinary equation. The rest of the paper is organized as follows: Section Two is the development of a new scheme. Section Three consists of some basic concepts vital to the development of the new scheme. Section Four consists of numerical experiment and discussion of results. Section Five concludes the paper.

2. Development of the New Scheme

We develop a new one-step scheme in solving an initial value problem in ordinary differential equations as follows. We intend to solve problem (1) with power series polynomial such that

$$y(x) = \sum_{i=0}^{1} \alpha_i x^i + \alpha_2 e^{-x} \tag{2}$$

With integration interval of [a,b] in the form $a = x_0 < x_1 < \cdots < x_n < x_{n+1} < \cdots < x_N = b$ with step size h given by $h = x_{n+1} - x_n$ such that $n = 0,1,\ldots,N-1$.

Expanding (2) above we obtain the interpolant of the form:

$$F(x) = \alpha_0 + \alpha_1 x + \alpha_2 e^{-x} \tag{3}$$

where α_0, α_1 and α_2 are real undetermined coefficients.

At $x = x_n$, (3) becomes

$$F(x_n) = \alpha_0 + \alpha_1 x_n + \alpha_2 e^{-x_n} \tag{4}$$

Also,

$$F(x_{n+1}) = \alpha_0 + \alpha_1 x_{n+1} + \alpha_2 e^{-x_{n+1}} \tag{5}$$

Differentiating (4) and let $F'(x_n) = f_n$ and $F''(x_n) = f_n'$ yields

$$F'(x_n) = \alpha_1 - \alpha_2 e^{-x_n} = f_n \tag{6}$$

Similarly,

$$F''(x_n) = \alpha_2 e^{-x_n} = f_n' \tag{7}$$

Simplifying (6) and (7) above we obtain

$$\alpha_1 = f_n + f_n' \tag{8}$$

and

$$\alpha_2 = \frac{f_n'}{e^{-(b+nh)}} \tag{9}$$

Since

$$x_{n+1} = b + (n+1)h \tag{10}$$

and

$$x_n = b + nh \tag{11}$$

It worth mentioning in the present paper that b varies which makes our derived scheme an avenue to solve any problem whose initial condition is not limited only to $y(0) = 1$.

Subtracting (4) from (5) and using (8) and (9) with $b = 0$ yields

$$F(x_{n+1}) - F(x_n) = hf_n + [h + (e^{-h} - 1)]f_n' \tag{12}$$

Using the fact that

$$y_{n+1} - y_n = F(x_{n+1}) - F(x_n) \tag{13}$$

Substituting (12) into (13) we obtain

$$y_{n+1} - y_n = hf_n + [h + (e^{-h} - 1)]f_n' \tag{14}$$

Equation (14) is the new one-step scheme.

3. Some Basic Concepts [9]

We consider the following basic concepts which are very vital to the development of the new scheme.

3.1. Stability

A numerical method is said to be stable if the difference between the numerical solution and the exact solution can be made as small as possible, that is if there exists two positive numbers e_0 and K such that the following holds.

$$\|y_n - y(x_n)\| \le K\|e_0\| \tag{15}$$

3.2. Consistency

A numerical method with an increment function $\varphi(x_n; y, h)$ is said to be consistent with the initial value problem (1) under consideration, if

$$\varphi(x_n; y, h) = f(x, y) \tag{16}$$

3.3. Convergence

A numerical method is said to be convergent if for all initial value problem satisfying the hypothesis of the Lipschitz condition given by

$$\left|f(x,y) - f(x,y^*)\right| \le L\left|y - y^*\right| \tag{17}$$

where the Lipschitz constant L is denoted by $L = \max\left|f_y'(x,y)\right|$. The necessary and sufficient conditions for convergence are the stability and consistency.

3.4. Round off Error

This can be defined as the error due to computing device. They arise because it is possible to represent all real numbers exactly on a finite-state machine. It can be represented mathematically as

$$R_{n+1} = y_{n+1} - p_{n+1} \tag{18}$$

where y_{n+1} is the approximate solution and p_{n+1} is the computer output. The magnitude depends in the storage and the arithmetic operation adopted. In such cases, double precision are employed to guarantee an adequate approximation.

4. Numerical Experiment and Discussion of Results

This section presents a comparative study of the new scheme and the theoretical solution with different step sizes.

4.1. Numerical Experiment

It is usually necessary to demonstrate the suitability and applicability of the newly developed scheme. In this course, we translated the algorithm of our scheme into MATLAB programming language and implemented with a problem emanated from real life situations. The performance of the method was checked by comparing its accuracy and efficiency. The efficiency was determined from the number of iterations counts and number of functions evaluations per step while the accuracy is determined by the size of the discretization error estimated from the difference between the exact solution and the numerical approximations.

4.1.1. Application of the New Scheme to Real Life Problem

Let us assume that a colony of 1000 bacteria are multiplying at the rate of $r = 0.8$ per hour per individual (i.e an individual produces an average of 0.8 off spring every hour, see (Hahn B. D (1997)). How many bacteria are there after 1 hour? If we assume that the colony grows continuously and without restriction.

Solution

It is possible to model this growth with a differential equation of first order of the form:

$$\frac{dN(t)}{dt} = rN(t) \tag{19}$$

with initial condition:

$$N(0) = 1000 \tag{20}$$

Equations (19) and (20) connote the IVP given by

$$\frac{dN(t)}{dt} = rN(t), N(0) = 1000, r = 0.8 \tag{21}$$

where $N(t)$ is the population size at time t. The exact solution to (21) is obtained as

$$N(t) = 1000e^{0.8t} \tag{22}$$

Take $h = 0.0125$ and $h = 0.0100, t \in [0,1]$. The results generated from our scheme for $h = 0.0125$ and $h = 0.01$ are shown in Tables 1 and 2 below respectively.

4.1.2. Table of Results

Table 1. *Comparative Analyses of the Results of the New Scheme with Theoretical Solution for h = 0.0125.*

t-value	Exact-solution	Computed-solution	Error
0.0125000	1010.050167084167900000	1010.049792316084100000	3.747681e-004
0.0250000	1020.201340026755800000	1020.200582957764600000	7.570690e-004
0.0375000	1030.454533953516800000	1030.453386937238100000	1.147016e-003
0.0500000	1040.810774192388200000	1040.809229467362700000	1.544725e-003
0.0625000	1051.271096376024200000	1051.269146064173400000	1.950312e-003
0.0750000	1061.836546545359600000	1061.834182650425400000	2.363895e-003
0.0875000	1072.508181254216700000	1072.505395660181200000	2.785594e-003
0.1000000	1083.287067674958700000	1083.283852144445700000	3.215531e-003
0.1125000	1094.174283705210400000	1094.170629877864900000	3.653827e-003
0.1250000	1105.170918075647700000	1105.166817466496500000	4.100609e-003
0.1375000	1116.278070458871300000	1116.273514456662300000	4.556002e-003
0.1500000	1127.496851579375700000	1127.491831444896900000	5.020134e-003
0.1625000	1138.828383324621900000	1138.822890188999500000	5.493136e-003
0.1750000	1150.273798857227300000	1150.267823720201700000	5.975137e-003
0.1875000	1161.834242728283000000	1161.827776456463700000	6.466272e-003
0.2000000	1173.510870991810200000	1173.503904316909000000	6.966675e-003
0.2125000	1185.304851320365700000	1185.297374837407900000	7.476483e-003
0.2250000	1197.217363121810200000	1197.209367287323600000	7.995834e-003
0.2375000	1209.249597657251600000	1209.241072787431700000	8.524870e-003
0.2500000	1221.402758160169900000	1221.393694429024000000	9.063731e-003
0.2625000	1233.678059956743500000	1233.668447394210500000	9.612563e-003
0.2750000	1246.076730587381000000	1246.066559077428300000	1.017151e-002
0.2875000	1258.600009929478100000	1258.589269208173800000	1.074072e-002
0.3000000	1271.249150321404800000	1271.237829974968000000	1.132035e-002
0.3125000	1284.025416687741700000	1284.013506150565900000	1.191054e-002
0.3250000	1296.930086665772000000	1296.917575218426000000	1.251145e-002
0.3375000	1309.964450733247500000	1309.951327500450600000	1.312323e-002
0.3500000	1323.129812337437000000	1323.116066286008800000	1.374605e-002
0.3625000	1336.427488025472300000	1336.413107962257300000	1.438006e-002
0.3750000	1349.858807576003100000	1349.843782145770500000	1.502543e-002
0.3875000	1363.425114132178100000	1363.409431815493000000	1.568232e-002
0.4000000	1377.127764335957400000	1377.111413447029000000	1.635089e-002
0.4125000	1390.968128463780400000	1390.951097148280700000	1.703132e-002
0.4250000	1404.947590563594100000	1404.929866796450400000	1.772377e-002
0.4375000	1419.067548593257500000	1419.049120176418500000	1.842842e-002
0.4500000	1433.329414560340600000	1433.310269120513300000	1.914544e-002

t-value	Exact-solution	Computed-solution	Error
0.4625000	1447.734614663324700000	1447.714739649685000000	1.987501e-002
0.4750000	1462.284589434224900000	1462.263972116098100000	2.061732e-002
0.4875000	1476.980793882643000000	1476.959421347157100000	2.137254e-002
0.5000000	1491.824697641270600000	1491.802556790979700000	2.214085e-002
0.5125000	1506.817785112853700000	1506.794862663332400000	2.292245e-002
0.5250000	1521.961555618633800000	1521.937838096041300000	2.371752e-002
0.5375000	1537.257523548281600000	1537.232997286896800000	2.452626e-002
0.5500000	1552.707218511335900000	1552.681869651061600000	2.534886e-002
0.5625000	1568.312185490168800000	1568.285999974004000000	2.618552e-002
0.5750000	1584.073984994481600000	1584.046948565965100000	2.703643e-002
0.5875000	1599.994193217360400000	1599.966291417979800000	2.790180e-002
0.6000000	1616.074402192893300000	1616.045620359466100000	2.878183e-002
0.6125000	1632.316219955378800000	1632.286543217396200000	2.967674e-002
0.6250000	1648.721270700127700000	1648.690683977070000000	3.058672e-002
0.6375000	1665.291194945886000000	1665.259682944502200000	3.151200e-002
0.6500000	1682.027649698885900000	1681.995196910442500000	3.245279e-002
0.6625000	1698.932308618550200000	1698.898899316043500000	3.340930e-002
0.6750000	1716.006862184857900000	1715.972480420193700000	3.438176e-002
0.6875000	1733.253017867394600000	1733.217647468532300000	3.537040e-002
0.7000000	1750.672500296100200000	1750.636124864163100000	3.637543e-002
0.7125000	1768.267051433734400000	1768.229654340082300000	3.739709e-002
0.7250000	1786.038430750072600000	1785.999995133341400000	3.843562e-002
0.7375000	1803.988415397856000000	1803.948924160958800000	3.949124e-002
0.7500000	1822.118800390508100000	1822.078236197599800000	4.056419e-002
0.7625000	1840.431398781636300000	1840.389744055042700000	4.165473e-002
0.7750000	1858.928041846340900000	1858.885278763447100000	4.276308e-002
0.7875000	1877.610579264342100000	1877.566689754445900000	4.388951e-002
0.8000000	1896.480879304950100000	1896.435845046075400000	4.503426e-002
0.8125000	1915.540829013894800000	1915.494631429566000000	4.619758e-002
0.8250000	1934.792334402030100000	1934.744954658007100000	4.737974e-002
0.8375000	1954.237320635938000000	1954.188739636911600000	4.858100e-002
0.8500000	1973.877732230446100000	1973.827930616692600000	4.980161e-002
0.8625000	1993.715533243080700000	1993.664491387076400000	5.104186e-002
0.8750000	2013.752707470474900000	2013.700405473468000000	5.230200e-002
0.8875000	2033.991258646748700000	2033.937676335290700000	5.358231e-002
0.9000000	2054.433210643886000000	2054.378327566319100000	5.488308e-002
0.9125000	2075.080607674120500000	2075.024403097024800000	5.620458e-002
0.9250000	2095.935514494362500000	2095.877967398956600000	5.754710e-002
0.9375000	2117.000016612672400000	2116.941105691172700000	5.891092e-002
0.9500000	2138.276220496816200000	2138.215924148750700000	6.029635e-002
0.9625000	2159.766253784912400000	2159.704550113389400000	6.170367e-002
0.9750000	2181.472265498198800000	2181.409132306130700000	6.313319e-002
0.9875000	2203.396426255934300000	2203.331841042216600000	6.458521e-002
1.0000000	2225.540928492464900000	2225.474868448106000000	6.606004e-002

Table 2. *Comparative Analyses of the Results of the New Scheme and Theoretical Solution for h = 0.01.*

t-value	Exact-solution	Computed-solution	Error
0.0100000	1008.032085504273500000	1008.031893599467600000	1.919048e-04
0.0200000	1016.128685406094900000	1016.128298513728300000	3.868924e-04
0.0300000	1024.290317890621500000	1024.289732890798600000	5.849998e-04
0.0400000	1032.517505305118200000	1032.516719040404500000	7.862647e-04
0.0500000	1040.810774192388200000	1040.809783467408600000	9.907250e-04
0.0600000	1049.170655324470500000	1049.169456905503700000	1.198419e-03
0.0700000	1057.597683736611300000	1057.596274351180000000	1.409385e-03
0.0800000	1066.092398761505100000	1066.090775097962000000	1.623664e-03
0.0900000	1074.655344063813600000	1074.653502770922800000	1.841293e-03
0.1000000	1083.287067674958700000	1083.285005361474100000	2.062313e-03
0.1100000	1091.988122028197500000	1091.985835262436000000	2.286766e-03
0.1200000	1100.759063993978800000	1100.756549303389600000	2.514691e-03
0.1300000	1109.600454915582500000	1109.597708786311600000	2.746129e-03
0.1400000	1118.512860645045300000	1118.509879521496300000	2.981124e-03
0.1500000	1127.496851579375700000	1127.493631863766300000	3.219716e-03
0.1600000	1136.553002697060300000	1136.549540748973200000	3.461948e-03
0.1700000	1145.681893594861800000	1145.678185730792600000	3.707864e-03
0.1800000	1154.884108524913700000	1154.880151017813400000	3.957507e-03
0.1900000	1164.160236432112500000	1164.156025510925500000	4.210921e-03

t-value	Exact-solution	Computed-solution	Error
0.2000000	1173.510870991810200000	1173.506402841008400000	4.468151e-03
0.2100000	1182.936610647811000000	1182.931881406921300000	4.729241e-03
0.2200000	1192.438058650669500000	1192.433064413799700000	4.994237e-03
0.2300000	1202.015823096301600000	1202.010559911658400000	5.263185e-03
0.2400000	1211.670516964900800000	1211.664980834305400000	5.536131e-03
0.2500000	1221.402758160169900000	1221.396945038567500000	5.813122e-03
0.2600000	1231.213169548867700000	1231.207075343832200000	6.094205e-03
0.2700000	1241.102379000671800000	1241.095999571905600000	6.379429e-03
0.2800000	1251.071019428362400000	1251.064350587192200000	6.668841e-03
0.2900000	1261.119728828329500000	1261.112766337195600000	6.962491e-03
0.3000000	1271.249150321404800000	1271.241889893346100000	7.260428e-03
0.3100000	1281.459932194021300000	1281.452369492155400000	7.562702e-03
0.3200000	1291.752727939704100000	1291.744858576702000000	7.869363e-03
0.3300000	1302.128196300894400000	1302.120015838449300000	8.180462e-03
0.3400000	1312.587001311108500000	1312.578505259400800000	8.496052e-03
0.3500000	1323.129812337437000000	1323.120996154592600000	8.816183e-03
0.3600000	1333.757304123384500000	1333.748163214927900000	9.140908e-03
0.3700000	1344.470156832052900000	1344.460686550355400000	9.470282e-03
0.3800000	1355.269056089671900000	1355.259251733394900000	9.804356e-03
0.3900000	1366.154693029480100000	1366.144549843011600000	1.014319e-02
0.4000000	1377.127764335957400000	1377.117277508843200000	1.048683e-02
0.4100000	1388.188972289412500000	1388.178136955782700000	1.083533e-02
0.4200000	1399.339024810930800000	1399.327836048918700000	1.118876e-02
0.4300000	1410.578635507678700000	1410.567088338836800000	1.154717e-02
0.4400000	1421.908523718577500000	1421.896613107285200000	1.191061e-02
0.4500000	1433.329414560340600000	1433.317135413206400000	1.227915e-02
0.4600000	1444.842038973879400000	1444.829386139138900000	1.265283e-02
0.4700000	1456.447133771086600000	1456.434102037992500000	1.303173e-02
0.4800000	1468.145441681989700000	1468.132025780197900000	1.341590e-02
0.4900000	1479.937711402288600000	1479.923906001235100000	1.380540e-02
0.5000000	1491.824697641270600000	1491.810497349545400000	1.420029e-02
0.5100000	1503.807161170112100000	1503.792560534825700000	1.460064e-02
0.5200000	1515.885868870569100000	1515.870862376712500000	1.500649e-02
0.5300000	1528.061593784057300000	1528.046175853855400000	1.541793e-02
0.5400000	1540.335115161127300000	1540.319280153386900000	1.583501e-02
0.5500000	1552.707218511336400000	1552.690960720787400000	1.625779e-02
0.5600000	1565.178695653522000000	1565.162009310151900000	1.668634e-02
0.5700000	1577.750344766478300000	1577.733224034859900000	1.712073e-02
0.5800000	1590.422970440039800000	1590.405409418652900000	1.756102e-02
0.5900000	1603.197383726574600000	1603.179376447121100000	1.800728e-02
0.6000000	1616.074402192893800000	1616.055942619605000000	1.845957e-02
0.6100000	1629.054849972574900000	1629.035932001512900000	1.891797e-02
0.6200000	1642.139557818705700000	1642.120175277058500000	1.938254e-02
0.6300000	1655.329363157055700000	1655.309509802422800000	1.985335e-02
0.6400000	1668.625110139667400000	1668.604779659342700000	2.033048e-02
0.6500000	1682.027649698886800000	1682.006835709129700000	2.081399e-02
0.6600000	1695.537839601820500000	1695.516535647122700000	2.130395e-02
0.6700000	1709.156544505233600000	1709.134744057578500000	2.180045e-02
0.6800000	1722.884636010888000000	1722.862332469002000000	2.230354e-02
0.6900000	1736.722992721326400000	1736.700179409923500000	2.281331e-02
0.7000000	1750.672500296101600000	1750.649170465120300000	2.332983e-02
0.7100000	1764.734051508460200000	1764.710198332292300000	2.385318e-02
0.7200000	1778.908546302479000000	1778.884162879192700000	2.438342e-02
0.7300000	1793.196891850663400000	1793.171971201216400000	2.492065e-02
0.7400000	1807.600002612005300000	1807.574537679452100000	2.546493e-02
0.7500000	1822.118800390509700000	1822.092784039200100000	2.601635e-02
0.7600000	1836.754214394190500000	1836.727639408960600000	2.657499e-02
0.7700000	1851.507181294539100000	1851.480040379894700000	2.714091e-02
0.7800000	1866.378645286473300000	1866.350931065763900000	2.771422e-02
0.7900000	1881.369558148764100000	1881.341263163351400000	2.829499e-02
0.8000000	1896.480879304952200000	1896.451996013367500000	2.888329e-02
0.8100000	1911.713575884749200000	1911.684096661844700000	2.947922e-02
0.8200000	1927.068622785936000000	1927.038539922027000000	3.008286e-02
0.8300000	1942.547002736755100000	1942.516308436754100000	3.069430e-02
0.8400000	1958.149706358806700000	1958.118392741348800000	3.131362e-02
0.8500000	1973.877732230448600000	1973.845791327007600000	3.194090e-02
0.8600000	1989.732086950705000000	1989.699510704703000000	3.257625e-02

t-value	Exact-solution	Computed-solution	Error
0.8700000	2005.713785203689200000	2005.680565469595900000	3.321973e-02
0.8800000	2021.823849823545300000	2021.789978365967500000	3.387146e-02
0.8900000	2038.063311859907300000	2038.028780352672800000	3.453151e-02
0.9000000	2054.433210643888700000	2054.398010669118300000	3.519997e-02
0.9100000	2070.934593854599400000	2070.898716901770500000	3.587695e-02
0.9200000	2087.568517586197700000	2087.531955051199500000	3.656253e-02
0.9300000	2104.336046415478900000	2104.298789599659400000	3.725682e-02
0.9400000	2121.238253470012800000	2121.200293579212500000	3.795989e-02
0.9500000	2138.276220496819900000	2138.237548640399800000	3.867186e-02
0.9600000	2155.451037931604800000	2155.411645121465900000	3.939281e-02
0.9700000	2172.763804968546500000	2172.723682118135000000	4.012285e-02
0.9800000	2190.215629630644300000	2190.174767553951400000	4.086208e-02
0.9900000	2207.807628840633700000	2207.766018251183600000	4.161059e-02
1.0000000	2225.540928492468500000	2225.498560002297400000	4.236849e-02

4.2. Discussion of Results

The numerical experiment shows that the new scheme provides comparable results. The differences between our scheme and the theoretical solution are negligible from a practical point of view as we can see from Tables 1 and 2 above. Also we deduce that the smaller the step size, the more accurate is the new scheme. The numbers of bacteria present after 1 one hour for $h = 0.0125$ and $h = 0.01$ were obtained as 2225.474868448106000000 and 2225.498560002297400000 respectively.

5. Conclusion

In this paper, we have developed a new scheme for the solution of initial value problems in ordinary differential equations. The new scheme was used to obtain numerical solution to real life problem. The comparative analyses of the results were also presented. The numerical results of our scheme were compared favorably with the theoretical solution. Hence the new scheme is computational efficient, robust and easy to implement.

References

[1] Areo E. A. and Adeniyi R. B. (2014). Block implicit one-step method for the numerical integration of initial value problems in ordinary differential equations. International Journal of Mathematics and Statistics studies, 2 (3), 4-13.

[2] Aboiyar T., Luga T. and Ivorter B. V. (2015). Derivation of continuous linear multistep methods using Hermite polynomials as Basis functions. American Journal of Applied Mathematics and Statistics, 3 (6), 220-225.

[3] Boyce, W. E. and DiPrima, R. C. (2001). Elementary differential equation and boundary value problems. John Wiley and Sons.

[4] Collatz, L. (1960). Numerical treatment of differential equations. Springer Verlag Berlin.

[5] Erwin, K. (2003). Advanced engineering mathematics. Eighth Edition, Wiley Publisher.

[6] Fatunla S. O. (1980): "Numerical integrators for stiff and highly oscillatory differential equations", Mathematics of Computation 34, 373-390.

[7] Gilat, A. (2004). Matlab: An introduction with application. John Wiley and Sons. Gautschi W. (1961): "Numerical integration of ordinary differential equations based on trigonometric polynomials", NumerischeMathematik 3, 381-397.

[8] Kayode S. J., Ganiyu A. A., and Ajiboye A. S. (2016). On one-step method of Euler-Maruyana type for solution of stochastic differential equations using varying stepsizes. Open Access Library Journal.

[9] Ogunrinde R. B. and Fadugba S. E. (2012). Development of a new scheme for the solution of initial value problems in ordinary differential equations. IOSR Journal of Mathematics, 2, 24-29.

[10] Wallace C. S and Gupta G. K. (1973). General Linear multistep methods to solve ordinary differential equations. The Australian Computer Journal, 5, 62-69.

[11] Ying T. Y., Zurni O. and Kamarun H. M. (2014). Modified exponential rational methods for the numerical solution of first order initial value problems. Sains Malaysiana, 43(12), 1951-1959.

[12] http://www.encyclopedia.com/computing/dictionaries-thesauruses-pictures-and-press-releases/numerical-methods.