International Electronic Journal of Pure and Applied Mathematics
Volume 3 No. 3 2011, 235-243

THE NUMERICAL SOLUTIONS OF STIFF ORDINARY DIFFERENTIAL EQUATIONS USING SEMI IMPLICIT EXTRAPOLATION METHOD

A.O. Binuyo[1], S.A. Odejide[2 §], Y.A.S. Aregbesola[3]

[1]Department of Physical Sciences
Ajayi Crowther University
Oyo, NIGERIA

[2]Department of Mathematics
Obafemi Awolowo University
Ile-Ife, NIGERIA

[3]Department of Pure and Applied Mathematics
Ladoke Akintola University of Technology
Ogbomosho, NIGERIA

Abstract: In this paper, numerical method based on the implicit differencing scheme was used to obtain the approximate solutions of first order initial value problems of stiff ordinary differential equations. Some stiff ordinary differential equations were numerically solved with the semi-implicit extrapolation method and the obtained results were compared with the exact solutions, the fourth order Runge-Kutta method (RK4) and the implicit trapezoid method. The results show that the semi-implicit extrapolation method is convergent, accurate and works very well for the first order stiff ordinary differential equations considered as the test problems.

AMS Subject Classification: 65L05, 65L06, 65L07
Key Words: stiff differential equations, semi-implicit extrapolation method, fourth order Runge-Kutta method, implicit trapezoid method, linear equations, non-linear equations

1. Introduction

The initial value problems of stiff ordinary differential equations do occur in many fields of engineering sciences, particularly in the studies of the chemical reactions,

Received: March 3, 2011

§Correspondence author

the turbulent flow, chaos, vibrations, network analysis, simulation problems and so on [10]. In this article, we consider the ordinary differential equations with the initial conditions given in the form;

$$Y' = F(t, Y), t \in [0, T] \tag{1}$$

$$Y(0) = Y_0, \tag{2}$$

with the theoretical or exact solution of the form $Y_E(t)$ [10].

Stiff ordinary differential equations correspond to physical process whose components have disparate time scales or whose time scale is small compared to interval over which it is studied [7]. When the general solution of first order ordinary differential equations involves the sums or differences of terms of the form $a_i e^{\lambda_i(t)}, (i = 1, 2, \cdots, n)$ where $\lambda_1, \lambda_2, \cdots, \lambda_n$ are eigenvalues in case of systems of stiff ordinary differential equations usually with negative values of large or small magnitudes [3], such equations are called the stiff ordinary differential equations. The stiff problems usually involve the $Re(\lambda_i) < 0$ for $(i = 1, 2, \cdots, n)$ to decay exponentially fast as the time scale increases [2].

Stiff ordinary differential equations can be solved analytically where feasible and approximate numerical solutions can also be determined. When the equations are linear, the theoretical solutions can be easily determined with all the known analytical methods of solving ordinary differential equations [6]. But in the case of the non-linear equations of stiff ordinary differential equations, the theoretical solutions could not be easily obtained except by finding their approximate solutions numerically.

The difficulty that arises in attempting to obtain a numerical approximation to the solution $Y(t)$ of a stiff equation is that of its numerical stability [8]. In order to obtain the numerical solutions of such equations, it is necessary to use a step size h such that every one of the values $h_i = h\lambda_i(i = 1, 2, \cdots, s)$ where λ_i are the eigenvalues $J(x)$ or $\frac{\partial F}{\partial Y}$ (Jacobian matrix) lies within the region of stability of the numerical method [1].

Most realistic stiff equations do not have analytical solutions and so only the numerical procedures have to be used. The conventional methods like explicit Euler's method, explicit fourth order Runge Kutta method, Taylor's series, Picard iterative scheme and so on are usually restricted to a very small step size in order that the solution be stable [7]. This means that a great deal of computer time shall be required.

To overcome this stability limitation on the step size, we shall use the semi - implicit extrapolation method to obtain the numerical solutions of some linear and nonlinear problems considered in this work [4]. This method ensures that the approximate solutions to the equations decay exponentially as the number of iterations increase [3].

2. Description of the Method

In this article, we consider the interval $[0, T]$ in equation (1) which is sub-divided into m individual subintervals as;

$$0 = t_0 < t_1 < t_2 < \cdots < t_{m-1} < \cdots < t_m = T. \tag{3}$$

Let equation (1) be a stiff ordinary differential equation given in the form;

$$Y^{'}(t) = F(t, Y) \quad and \quad Y(0) = Y_0. \tag{4}$$

Using the backward Euler's scheme given in the form;

$$Y_{k+1} = Y_k + hF(t_{k+1}, Y_{k+1}) \tag{5}$$

Linearizing equation (5) as in Newton's method about $F(Y_k)$ [4], we obtain

$$Y_{k+1} = Y_k + h[F(Y_k) + (Y_{k+1} - Y_k)\frac{\partial F}{\partial Y}]. \tag{6}$$

Expanding, we have

$$Y_{k+1} = Y_k + h[F(Y_k)] + hY_{k+1}\frac{\partial F}{\partial Y} - hY_k\frac{\partial F}{\partial Y}. \tag{7}$$

Rearranging, we obtain

$$Y_{k+1} = Y_k + h(1 - h\frac{\partial F}{\partial Y})^{-1}F(Y_k) \tag{8}$$

where $\frac{\partial F}{\partial Y}$ is called the Jacobian matrix and h is the step size. Equation (8) is called the semi-implicit Euler's method [9].

From the implicit form of the midpoint rule given in the form [4];

$$\frac{Y_{k+1} - Y_{k-1}}{2} = hF(\frac{Y_{k+1} + Y_{k-1}}{2}). \tag{9}$$

By linearizing the right-hand of equation (9) about $F(Y_k)$ as in Newton's method, we obtain;

$$\frac{Y_{k+1} - Y_{k-1}}{2} = h[F(Y_k) + (\frac{Y_{k+1} + Y_{k-1}}{2} - Y_k)\frac{\partial F}{\partial Y}]. \tag{10}$$

Expanding, we have,

$$Y_{k+1} - Y_{k-1} = 2hF(Y_k) + hY_{k+1}\frac{\partial F}{\partial Y} + hY_{k-1}\frac{\partial F}{\partial Y} - 2hY_k\frac{\partial F}{\partial Y}. \tag{11}$$

Rearranging equation (11), we obtain,

$$Y_{k+1}(1 - h\frac{\partial F}{\partial Y}) = Y_{k-1}(1 + h\frac{\partial F}{\partial Y}) + 2h[F(Y_k) - Y_k\frac{\partial F}{\partial Y}]. \tag{12}$$

Equation (12) is called the semi-implicit midpoint rule.

For computational efficiency, the equations of the semi-implicit extrapolation method shall be written in terms of the change in the solutions.

Let

$$D_k = Y_{k+1} - Y_k, D_{k-1} = Y_k - Y_{k-1} \quad and \quad Y_{k+1} = Y_k + D_k \tag{13}$$

such that with $k = 0$, we have,

$$D_0 = Y_1 - Y_0 \quad or \quad Y_1 = Y_0 + D_0 \tag{14}$$

From equation (8), we have,

$$Y_{k+1} - Y_k = h(1 - h\frac{\partial F}{\partial Y})^{-1} F(Y_k). \tag{15}$$

With $k = 0$, we obtain

$$Y_1 - Y_0 = h(1 - h\frac{\partial F}{\partial Y})^{-1} F(Y_0). \tag{16}$$

Then using equation (14) in equation (16), we have

$$D_0 = h(1 - h\frac{\partial F}{\partial Y})^{-1} F(Y_0). \tag{17}$$

such that

$$Y_1 = Y_0 + D_0 \tag{18}$$

From equation (12), i.e.,

$$Y_{k+1}(1 - h\frac{\partial F}{\partial Y}) = Y_{k-1}(1 + h\frac{\partial F}{\partial Y}) + 2h[F(Y_k) - Y_k\frac{\partial F}{\partial Y}] \tag{19}$$

By manipulating expression (19) to obtain,

$$Y_{k+1}(1 - h\frac{\partial F}{\partial Y}) = 2Y_{k-1} - Y_{k-1} + 2Y_k - 2Y_k + hY_{k-1}\frac{\partial F}{\partial Y} + 2hF(Y_k) - 2hY_k\frac{\partial F}{\partial Y} \tag{20}$$

On rearranging equation (20) and replacing with (13), we obtain

$$D_k(1 - h\frac{\partial F}{\partial Y}) = D_{k-1}(1 - h\frac{\partial F}{\partial Y}) + 2[hF(Y_k) - D_{k-1}]. \tag{21}$$

Rearranging equation (21), we have

$$D_k = D_{k-1} + 2(1 - h\frac{\partial F}{\partial Y})^{-1}[hF(Y_k) - D_{k-1}] \tag{22}$$

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 3, No. 3 (2011)

and

$$Y_{k+1} = Y_k + D_k. \tag{23}$$

Finally, to smooth the last step of the computation, we compute using

$$D_m = (1 - h\frac{\partial F}{\partial Y})^{-1}[hF(Y_m) - D_{m-1}] \tag{24}$$

such that

$$\overline{Y_m} = Y_m + D_m, \tag{25}$$

where $\overline{Y_m}$ gives the approximate solution to the equation (1) and m is the number of steps to be used in going across the interval from t_0 to t_1 and $h = \frac{(t_1 - t_0)}{m}$ is the step size.

The semi implicit form of the implicit midpoint rule together with a special first step of the semi implicit Euler step and a smoothing last step forms the basic algorithm of the semi implicit extrapolation method.

Equations (17) and (14) are used to compute the approximate solution for equation (1) at the initial points of the subinterval while equations (22) and (23) are used to compute the extrapolated values at the points between the subintervals i.e. for $k = 1, \cdot, m-1$. Equations (24) and (25) are used to give the approximate solutions to equation (1) at the end points of the sub-interval i.e. for $k = m$. The sub-interval is divided into $m-1$ equal intervals where point m is the end point of the sub-intervals.

3. Numerical Examples

In this section, we illustrate the efficiency of this method as a novel solver for the stiff ordinary differential equations. In order to do so, three different problems were selected as test problems. Numerical results were presented in tabular forms and all the numerical calculations were done with the aid of Maple 13 software [5, 10].

Example 1. We consider the following linear equation of stiff ordinary differential equations of the form;

$$y'(t) = 30\sin t - 30y(t) + 3\cos t; y(0) = 0 \tag{26}$$

The exact solution is given as:

$$y(t) = \frac{60}{901}\cos t + \frac{903}{901}\sin t - \frac{60}{901}e^{-30t}. \tag{27}$$

The numerical results are presented in the Tables 1 and 2.

n	t	$Y_E(t)$	$Y_{SE}(t)$	$Y_{RK4}(t)$	$Y_{TR}(t)$
1	0	0.0	0.0	0.0	0.0
2	0.01	0.02727830886	0.02723826303	0.02591625000	0.02739063044
3	0.02	0.05007558076	0.05001620236	0.04770626587	0.05024179216
4	0.03	0.06955023211	0.06948419180	0.06643656212	0.06973469476
5	0.04	0.08656018029	0.08649488210	0.08289699396	0.08674214593
6	0.05	0.10174072790	0.10168018690	0.09767245735	0.10190900440
7	0.06	0.11556226080	0.11550836140	0.11119600690	0.11571164520
8	0.07	0.12837299170	0.12832632210	0.12378820840	0.12850191270
9	0.08	0.14043062580	0.14039102250	0.13568629100	0.14053960660
10	0.09	0.15192581920	0.15189271700	0.14706574670	0.15201649450
11	0.10	0.16299955720	0.16297220790	0.15805632960	0.16307405970

Table 1: Exact solution ($Y_E(t)$), Semi-Implicit Extrapolation method ($Y_{SE}(t)$), the fourth order Runge-Kutta method ($Y_{RK4}(t)$) and the implicit trapezoid method ($Y_{TR}(t)$) for the example 1 with $h = 0.01$

n	t	$\lvert Y_E(t) - Y_{SE}(t)\rvert$	$\lvert Y_E(t) - Y_{RK4}(t)\rvert$	$\lvert Y_E(t) - Y_{TR}(t)\rvert$
1	0	0.0	0.0	0.0
2	0.01	0.00004004583	0.00136205886	0.00011232158
3	0.02	0.00005937840	0.00236931489	0.00016621140
4	0.03	0.00006604031	0.00311366999	0.00018446265
5	0.04	0.00006529819	0.00366318633	0.00018196564
6	0.05	0.00006054100	0.00406827055	0.00016827650
7	0.06	0.00005389940	0.00436625390	0.00014938440
8	0.07	0.00004666960	0.00458478330	0.00012892100
9	0.08	0.00003960330	0.00474433480	0.00010898080
10	0.09	0.00003310220	0.00486007250	0.00009067530
11	0.10	0.00002734930	0.00494322760	0.00007450250

Table 2: The associated errors for the example 1 with $h = 0.01$

Example 2. Next, we consider the following linear equation of stiff ordinary differential equations of the form;

$$y'(t) = -20(y(t) - t^2) + 2t; y(0) = \frac{1}{3} \tag{28}$$

Here the exact solution is given as;

$$y(t) = t^2 + \frac{e^{-20t}}{3} \tag{29}$$

n	t	$Y_E(t)$	$Y_{SE}(t)$	$Y_{RK4}(t)$	$Y_{TR}(t)$
1	0	0.33333333333	0.33333333333	0.333333333333	0.33333333333
2	0.1	0.05511176107	0.05575217192	0.111111111110	0.01000000000
3	0.2	0.04610521296	0.04630345293	0.050370370330	0.04000000000
4	0.3	0.09082625073	0.09089210603	0.056790123430	0.09000000000
5	0.4	0.16011182090	0.16014980880	0.098930041150	0.16000000000
6	0.5	0.25001513330	0.25004798470	0.166310013700	0.25000000000
7	0.6	0.36000204810	0.36003401710	0.255436671200	0.36000000000
8	0.7	0.49000027720	0.49003210110	0.365145557000	0.49000000000
9	0.8	0.64000003750	0.64003183830	0.495048519000	0.64000000000
10	0.9	0.81000000510	0.81003180230	0.645016173000	0.81000000000
11	1.0	1.00000000100	1.00003179700	0.815005391000	1.00000000000

Table 3: Exact solution ($Y_E(t)$), Semi-Implicit Extrapolation method ($Y_{SE}(t)$), the fourth order Runge-Kutta method ($Y_{RK4}(t)$) and the implicit trapezoid method ($Y_{TR}(t)$) for the example 2 with $h = 0.1$

n	t	$\lvert Y_E(t) - Y_{SE}(t)\rvert$	$\lvert Y_E(t) - Y_{RK4}(t)\rvert$	$\lvert Y_E(t) - Y_{TR}(t)\rvert$
1	0	0.0	0.0	0.0
2	0.1	0.00064041085	0.05599934993	0.04511176107
3	0.2	0.00019823997	0.00426515737	0.00610521296
4	0.3	0.00006585530	0.03403612730	0.00082625073
5	0.4	0.00003798790	0.06118177975	0.00011182090
6	0.5	0.00003285140	0.08370511960	0.000015133300
7	0.6	0.00003196900	0.10456537690	0.00002048100
8	0.7	0.00003182390	0.12485472020	0.00000277200
9	0.8	0.00003180080	0.14495151850	0.00000037500
10	0.9	0.00003179720	0.16498383210	0.00000005100
11	1.0	0.00003179600	0.18499461000	0.00000000100

Table 4: The associated errors for the example 2 with $h = 0.1$

The numerical results are presented in the Tables 3 and 4.

Example 3. Also, we consider the stiff ordinary differential equations of the form;

$$y'(t) = \frac{50}{y(t)} - 50y(t); y(0) = \sqrt{2}, \tag{30}$$

whose exact solution is given by

$$y(t) = \sqrt{1 + e^{-100t}}. \tag{31}$$

n	t	$Y_E(t)$	$Y_{SE}(t)$	$Y_{RK4}(t)$	$Y_{TR}(t)$
1	0	1.414213562	1.414213562	1.414213562	1.414213562
2	0.1	1.000022700	1.000000000	-15.85259706	19.50578492
3	0.2	1.000000001	1.000000000	-215.7458704	395.9537526

Table 5: Exact solution ($Y_E(t)$), Semi-Implicit Extrapolation method ($Y_{SE}(t)$), the fourth order Runge-Kutta method ($Y_{RK4}(t)$) and the implicit trapezoid method ($Y_{TR}(t)$) for the example 3 with $h = 0.1$

n	t	$Y_E(t)$	$Y_{SE}(t)$	$Y_{RK4}(t)$	$Y_{TR}(t)$
1	0	1.414213562	1.414213562	1.414213562	1.414213562
2	0.01	1.169563782	1.170464296	1.171055650	3.631431091
3	0.02	1.065521132	1.066280815	1.067032289	47.80671916

Table 6: For $h = 0.01$

n	t	$\|Y_E(t) - Y_{SE}(t)\|$	$\|Y_E(t) - Y_{RK4}(t)\|$	$\|Y_E(t) - Y_{TR}(t)\|$
1	0	0.0	0.0	0.0
2	0.1	0.000022700	16.85261976	18.50576222
3	0.2	0.000000001	216.7458704	394.9537526

Table 7: The associated errors with $h = 0.1$

n	t	$\|Y_E(t) - Y_{SE}(t)\|$	$\|Y_E(t) - Y_{RK4}(t)\|$	$\|Y_E(t) - Y_{TR}(t)\|$
1	0	0.0	0.0	0.0
2	0.01	0.000900514	0.001491876	2.461867309
3	0.02	0.000759683	0.001511157	46.74119803

Table 8: The associated errors with $h = 0.01$

The numerical results are presented in the Tables 5-8 with different values of h.

4. Results and Conclusion

We have used the semi-implicit extrapolation method, the fourth order Runge-Kutta method and implicit trapezoid method to obtain the approximate solutions of some sets of equations of stiff ordinary differential equations. It can be observed that the semi-implicit extrapolation method compares favourably well with the exact. This method is efficient, direct and second order accurate which can be easily implemented on computers.

It can be concluded that the results obtained with the use semi-implicit extrapolation method are accurate which shows high capability of the method compare to other explicit methods like the fourth order Runge-Kutta method. This method helps the approximate solutions not to diverge from the exact solutions, it works very well and it does not require extremely small value of the step size as can be seen in Tables 5-8.

References

[1] K.E. Atkinson, *An Introduction to Numerical Analysis*, John Wiley and Sons, New York (1989).

[2] C.W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, Englewood Cliffs, NJ, Prentice Hall, Chapter 9 (1971).

[3] N. Guzel, M. Bayram, On the numerical solution of stiff systems, *Appl. Maths. Comp.*, **170** (2005), 230-236.

[4] V.F. Laureen, *Applied Numerical Analysis using MATLAB*, Second Edition, Pearson, Prentice Hall, Upper Saddle River, NJ (2008), 538-553.

[5] *Maple 13*, Software Package, Waterloo Maple Inc., Waterloo, Ontario, Canada.

[6] P. Dawkins, *Online Lecture Notes on Differential Equations*, http://www.tutorial.math.com (2007).

[7] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes in Fortran 77. The Art of Scientific Computing*, Second Edition, Cambridge University Press, Cambridge, UK (1992).

[8] J. Stoer, R. Bulirsch, *Introduction to Numerical Analysis*, Springer Verlag New York, Chapter 7 (1980).

[9] A. Tahmasbi, Numerical solutions for stiff ordinary differential equations systems, *International Mathematical Forum*, **3**, No. 15 (2008), 703-711.

[10] Y. Tetsu, *Mathematical Models with Maple. Applied System*, Second Edition, Cybernet System, Otsuka 2-9-3, Bunkyo-Ku, Tokyo 112-0012, Japan (2005), 151-155.