World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:10, No:11, 2016

Development of Extended Trapezoidal Method for Numerical Solution of Volterra Integro-Differential Equations

Fuziyah Ishak, Siti Norazura Ahmad

***Abstract*—**Volterra integro-differential equations appear in many models for real life phenomena. Since analytical solutions for this type of differential equations are hard and at times impossible to attain, engineers and scientists resort to numerical solutions that can be made as accurately as possible. Conventionally, numerical methods for ordinary differential equations are adapted to solve Volterra integro-differential equations. In this paper, numerical solution for solving Volterra integro-differential equation using extended trapezoidal method is described. Formulae for the integral and differential parts of the equation are presented. Numerical results show that the extended method is suitable for solving first order Volterra integro-differential equations.

***Keywords*—**Accuracy, extended trapezoidal method, numerical solution, Volterra integro-differential equations.

I. Introduction

VOLTERRA integro-differential equations (VIDEs) play important roles in modeling real life phenomena with various disciplines including natural science, engineering, physics, economics and biology. Applications of such equations as stated in [1] include heat transfer, diffusion process in general, neutron diffusion and many more. Since the importance of VIDEs in modeling is increasing, finding solutions for the equations has attracted many researchers including scientists and engineers for decades. It is known that analytical solutions for VIDEs are very hard and at times impossible to obtain. As an alternative, scientists and engineers seek numerical solutions which can be made as accurately as possible.

Numerical methods for solving ordinary differential equations (ODEs) are adapted to solve VIDEs where the integral part of the equations is approximated using quadrature formulas. Runge-Kutta type of methods and linear multistep methods are the most common methods for solving VIDEs. See, for example the implementations proposed by [2]-[10].

In this paper, we propose solving VIDEs using extended one-step trapezoidal method. The extended method with higher order of convergence and improved stability conditions is suitable for solving many types of differential equations. Previous literature have shown extensive implementations of extended one-step methods in solving ODEs, delay differential equations and combination scheme of solving stiff and non-stiff equations. Readers are advised to refer published work on various implementations of extended one-step methods for numerical solutions of differential equations, see for example [11]–[15]. The motivation for this research is due to existing limitation in the implementation of extended one-step trapezoidal method for solving VIDEs. The extended trapezoidal method has been first developed by [11] to solve ODEs with third order convergence while preserving the property of *A*-stability of the classical trapezoidal method. As cited in [11], Dahlquist stated in 1963 that a method is to be *A*-stable if the numerical solution of differential equation $y' = \lambda y$ where $\mathrm{Re}(\lambda) < 0$ approaches zero as stepsize approaches zero.

We focus on the development of numerical solution for solving initial value VIDEs of the form:

$$y'(x) = f(x, y(x)) + \int_{x_0}^{x} F(x, s, y(s))ds, \quad x_0 \le x \le x_N, \qquad (1)$$
$$y(x_0) = y_0,$$

where the given functions $f(x, y(x))$ and $F(x, s, y(s))$ satisfy Lipschitz conditions in their arguments such that the solution $y(x)$ exists. The value y_0 is the given initial condition.

The organization of this paper is as follows. In Section II, we discuss the development of the proposed method. Numerical results and related discussions are presented in Section III. Section IV highlights the conclusions.

II. Method Development

We consider (1) where the interval $[x_0, x_N]$ is divided into N subintervals with stepsize $h = \frac{x_N - x_0}{N}$. The notation y_n refers to the approximate solution for $y(x_n)$ where y is the solution of (1). The grids $x_n = x_0 + ih, \ i = 0,1,2,\ldots,N$ represent N equal subintervals. It is assumed that approximate solutions have been obtained up to x_n. The immediate task is to evaluate y_{n+1}. The formulae pair y_{n+1}, approximate solution for $y(x_{n+1})$ and $\hat{y}_{n+1}$, predicted value for y_{n+1} are implicit and being implemented in *PECE* mode where *P*

F. Ishak is with Universiti Teknologi MARA, 40450 Shah Alam, Selangor Darul Ehsan, Malaysia (phone: +603-5543-5348; e-mail: fuziyah@tmsk.uitm.edu.my).

S. N. Ahmad is with Kolej Universiti Poly-Tech MARA, Jln 7/91. Taman Shamelin Perkasa, 56100 Cheras, Kuala Lumpur, Malaysia (e-mail: sitinorazura@gapps.kptm.edu.my).

stands for predict, E stands for function evaluation and C is for correct.

A. Formulae Derivation

We derive the formulae by integrating (1) on both sides with limit of integration from x_n to x_{n+1} to obtain:

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} f(x, y(x))\,dx + \int_{x_n}^{x_{n+1}} \int_{x_0}^{x} F(x, s, y(s))\,ds\,dx$$
$$= y_n + I_1 + I_2$$

where

$$I_1 = \int_{x_n}^{x_{n+1}} f(x, y(x))\,dx \text{ and } I_2 = \int_{x_n}^{x_{n+1}} \int_{x_0}^{x} F(x, s, y(s))\,ds\,dx.$$

The integral I_1 is solved by first interpolating $f(x, y(x))$ by $P(x)$ using interpolation points (x_n, f_n), (x_{n+1}, f_{n+1}) and (x_n, f_{n+2}). The polynomial $P(x)$ is given by:

$$P(x) = \sum_{k=n}^{n+2} f_k L_k(x)$$

and for each $k = n, n+1, n+2$,

$$L_k(x) = \prod_{\substack{i=n \\ i \neq k}}^{n+2} \frac{(x - x_i)}{(x_k - x_i)}.$$

Here, we denote $f_n = f(x_n, y_n)$, $f_{n+1} = f(x_{n+1}, \hat{y}_{n+1})$ and $f_{n+2} = f(x_{n+2}, \hat{y}_{n+2})$. The notations $\hat{y}_{n+1}$ and $\hat{y}_{n+2}$ refer to predicted values for y_{n+1} and y_{n+2} respectively. Using $x = x_n + Sh$, we have:

$$I_1 = h\int_0^1 P(S)\,dS$$
$$= h\int_0^1 \left(\frac{S^2 - 3S + 2}{2} f_n + (-S^2 + 2S) f_{n+1} + \frac{S^2 - S}{2} f_{n+2} \right) dS$$
$$= \frac{h}{12}(5 f_n + 8 f_{n+1} - f_{n+2})$$

which the coefficients are as given in [11], [16]. Since the formula for I_1 is implicit, we predict the values for y_{n+1} and y_{n+2} using the formulae suggested in [11] as:

$$\hat{y}_{n+1}^{(0)} = y_n + hf_n,$$
$$\hat{y}_{n+1}^{(1)} = y_n + \frac{h}{2}\left[f_n + f(x_{n+1}, \hat{y}_{n+1}^{(0)})\right],$$
$$\hat{y}_{n+1} = \hat{y}_{n+1}^{(1)},$$
$$\hat{y}_{n+2} = 5y_n - 4y_{n+1} + h(2 f_n + 4 f_{n+1}).$$

In similar manner we obtain the integral I_2 using trapezoidal method. Thus,

$$I_2 = \frac{h}{2}\left(\int_{x_0}^{x_n} F(x_n, s, y(s))ds + \int_{x_0}^{x_{n+1}} F(x_{n+1}, s, y(s))ds \right)$$
$$= I_{21} + I_{22}$$

where

$$I_{21} = \frac{h}{2}\int_{x_0}^{x_n} F(x_n, s, y(s))\,ds$$
$$= \frac{h}{2}[\frac{h}{2} F(x_n, x_0, y_0) + hF(x_n, x_1, y_1) + \cdots + hF(x_n, x_{n-1}, y_{n-1}) + \frac{h}{2} F(x_n, x_n, y_n)]$$
$$= \frac{h^2}{4}[F(x_n, x_0, y_0) + 2F(x_n, x_1, y_1) + \cdots + 2F(x_n, x_{n-1}, y_{n-1}) + F(x_n, x_n, y_n)]$$

and

$$I_{22} = \frac{h}{2}\int_{x_0}^{x_{n+1}} F(x_{n+1}, s, y(s))\,ds$$
$$= \frac{h}{2}[\frac{h}{2} F(x_{n+1}, x_0, y_0) + hF(x_{n+1}, x_1, y_1) + \cdots + hF(x_{n+1}, x_n, y_n) + \frac{h}{2} F(x_{n+1}, x_{n+1}, \hat{y}_{n+1})]$$
$$= \frac{h^2}{4}[F(x_{n+1}, x_0, y_0) + 2F(x_{n+1}, x_1, y_1) + \cdots + 2F(x_{n+1}, x_n, y_n) + F(x_{n+1}, x_{n+1}, \hat{y}_{n+1})].$$

The derivatives are evaluated as follows,

$$y'_0 = f(x_0, y_0),$$
$$y'_{n+1} = f_{n+1} + \frac{h}{2}[F(x_{n+1}, x_0, y_0) + 2F(x_{n+1}, x_1, y_1) + \cdots + 2F(x_{n+1}, x_n, y_n) + F(x_{n+1}, x_{n+1}, y_{n+1})],$$
$$y'_{n+2} = f_{n+2} + \frac{h}{2}[F(x_{n+2}, x_0, y_0) + 2F(x_{n+2}, x_1, y_1) + \cdots + 2F(x_{n+2}, x_{n+1}, y_{n+1}) + F(x_{n+2}, x_{n+2}, y_{n+2})].$$

B. Algorithm

The development of the algorithm is shown in the following

Fig. 1.

Step 1: Begin. Initialize all given values.
Step 2: Calculate $\hat{y}_{n+1}$ and $\hat{y}_{n+2}$.
Step 3: Solve for I_1.
Step 4: Solve for I_2.
Step 5: Calculate y_{n+1}.
Step 6: Calculate error.
Step 7: Calculate y'_{n+1} and y'_{n+2} for the next iteration.
Step 8: Update *x*.
Step 9: Go to Step 2 if the endpoint is not reached. Else go to Step 10.
Step 10: Stop.

Fig. 1 Algorithm for solving VIDEs

C. Test Problems

In order to analyze the accuracy of the method, we test the algorithm on several test problems with exact solutions. The test problems are taken from [2], [17].

– **Test Problem 1**

$$y'(x) = 1 - \int_0^x y(s)\,ds, \quad y(0) = 0, \quad 0 \le x \le 1.$$

Exact solution is $y(x) = \sin x$.

– **Test Problem 2**

$$y'(x) = 1 + 2x - y + \int_0^x x(1+2x)e^{s(x-s)}y(s)\,ds,$$
$$y(0) = 1, \quad 0 \le x \le 1.$$

Exact solution is $y(x) = e^{x^2}$.

– **Test Problem 3**

$$y'(x) = 2 - x + \frac{1}{6}x^3 - \int_0^x (x-s)y(s)\,ds,$$
$$y(0) = -1, \quad 0 \le x \le 1.$$

Exact solution is $y(x) = x - e^{-x}$.

– **Test Problem 4**

$$y'(x) = 1 + x + \int_0^x (x-s)y(s)\,ds,$$
$$y(0) = 1, \quad 0 \le x \le 1.$$

Exact solution is $y(x) = e^x$.

III. Numerical Results and Discussions

In this section, numerical results are displayed in terms or absolute error at grid points using various stepsizes for Test Problem 1 – Test Problem 4. For illustrative purposes, errors at specified grids with stepsizes $h = 0.1$, $h = 0.025$ and $h = 0.01$ are tabulated. We compare the results of using extended trapezoidal method for solving VIDEs with that of the Euler method. The errors in the numerical solutions for Test Problem 1 – Test Problem 4 are given in Tables I-IV respectively. The abbreviations *h* refers to stepsize, e. trap means extended trapezoidal method and 4.98e-2 is equivalent to 4.98×10^{-2}.

TABLE I
Errors in the Solutions for Test Problem 1

x	h = 0.1		h = 0.025		h = 0.01	
	euler	e. trap	euler	e. trap	euler	e.trap
0.1	4.98e-2	8.28e-5	4.99e-2	5.18e-6	4.99e-2	8.29e-7
0.2	9.90e-2	1.63e-4	9.93e-2	1.02e-5	9.93e-2	1.63e-5
0.3	1.47e-1	2.38e-4	1.48e-1	1.49e-5	1.48e-1	2.39e-6
0.4	1.94e-1	3.07e-4	1.94e-1	1.92e-5	1.95e-1	3.07e-6
0.5	2.38e-1	3.65e-4	2.39e-1	2.29e-5	2.40e-1	3.66e-6
0.6	2.80e-1	4.12e-4	2.82e-1	2.58e-5	2.82e-1	4.13e-6
0.7	3.19e-1	4.46e-4	3.21e-1	2.79e-5	3.22e-1	4.46e-6
0.8	3.55e-1	4.64e-4	3.58e-1	2.90e-5	3.58e-1	4.64e-6
0.9	3.87e-1	4.66e-4	3.91e-1	2.91e-5	3.91e-1	4.66e-6
1.0	4.15e-1	4.50e-4	4.19e-1	2.81e-5	4.20e-1	4.50e-6

TABLE II
Errors in the Solutions for Test Problem 2

x	h = 0.1		h = 0.025		h = 0.01	
	euler	e. trap	euler	e. trap	euler	e.trap
0.1	9.75e-3	2.00e-4	5.93e-3	1.31e-5	5.24e-3	2.10e-6
0.2	2.78e-2	5.06e-4	2.04e-2	3.20e-5	1.90e-2	5.11e-6
0.3	5.29e-2	9.23e-4	4.20e-2	5.79e-5	3.99e-2	9.23e-6
0.4	8.37e-2	1.49e-3	6.95e-2	9.29e-5	6.66e-2	1.48e-5
0.5	1.19e-1	2.26e-3	1.01e-1	1.41e-4	9.80e-2	2.25e-5
0.6	1.59e-1	3.34e-3	1.37e-1	2.08e-4	1.33e-1	3.33e-5
0.7	2.01e-1	4.88e-3	1.76e-1	3.04e-4	1.70e-1	4.86e-5
0.8	2.47e-1	7012e-3	2.16e-1	4.44e-4	2.10e-1	7.09e-5
0.9	2.96e-1	1.04e-2	2.58e-1	6.51e-4	2.50e-1	1.04e-4
1.0	3.48e-1	1.55e-2	3.01e-1	9.63e-4	2.91e-1	1.54e-4

TABLE III
Errors in the Solutions for Test Problem 3

x	h = 0.1		h = 0.025		h = 0.01	
	euler	e. trap	euler	e. trap	euler	e.trap
0.1	9.49e-2	8.74e-5	9.69e-2	5.66e-6	9.72e-2	9.12e-7
0.2	1.85e-1	1.91e-4	1.89e-1	1.24e-5	1.89e-1	1.99e-6
0.3	2.70e-1	3.12e-4	2.75e-1	2.01e-5	2.77e-1	3.23e-6
0.4	3.50e-1	4.49e-4	3.57e-1	2.88e-5	3.59e-1	4.64e-6
0.5	4.24e-1	6.01e-4	4.33e-1	3.86e-5	4.35e-1	6.20e-6
0.6	4.93e-1	7.69e-4	5.04e-1	4.93e-5	5.06e-1	7.92e-6
0.7	5.56e-1	9.52e-4	5.69e-1	6.09e-5	5.72e-1	9.78e-6
0.8	6.14e-1	1.15e-3	6.28e-1	7.33e-5	6.31e-1	1.18e-5
0.9	6.66e-1	1.35e-3	6.82e-1	8.66e-5	6.84e-1	1.39e-5
1.0	7.12e-1	1.58e-3	7.28e-1	1.01e-4	7.31e-1	1.61e-5

From the tabulated errors in the numerical results, it is clearly seen that extended trapezoidal method is suitable for solving VIDEs as compared to the existing Euler method. For various values of *h*, the errors in using extended trapezoidal method improve as the stepsizes get smaller. Compared to

Euler method, using extended trapezoidal method to solve VIDEs improves the overall accuracy of the numerical results. Thus it can be concluded that extended trapezoidal method is suitable for solving VIDEs.

TABLE IV
ERRORS IN THE SOLUTIONS FOR TEST PROBLEM 4

x	$h = 0.1$		$h = 0.025$		$h = 0.01$	
	euler	e. trap	euler	e. trap	euler	e.trap
0.1	5.49e-2	7.91e-5	5.31e-2	4.95e-6	5.28e-2	7.92e-7
0.2	1.15e-1	1.50e-4	1.11e-1	9.38e-6	1.11e-1	1.50e-6
0.3	1.80e-1	2.12e-4	1.75e-1	1.33e-5	1.73e-1	2.13e-6
0.4	2.50e-1	2.67e-4	2.43e-1	1.67e-5	2.42e-1	2.67e-6
0.5	3.26e-1	3.14e-4	3.17e-1	1.97e-5	3.15e-1	3.14e-6
0.6	4.08e-1	3.53e-4	3.97e-1	2.21e-5	3.95e-1	3.54e-6
0.7	4.95e-1	3.85e-4	4.83e-1	2.41e-5	4.80e-1	3.86e-6
0.8	5.90e-1	4.11e-4	5.75e-1	2.57e-5	5.72e-1	4.12e-6
0.9	6.91e-1	4.30e-4	6.74e-1	2.69e-5	6.71e-1	4.31e-6
1.0	8.00e-1	4.43e-4	7.82e-1	2.78e-5	7.78e-1	4.45e-6

IV. CONCLUSIONS

We have presented the development of extended trapezoidal one-step method for the numerical solution of VIDEs. The extended trapezoidal method has been and adapted to solve VIDEs and further implemented in *PECE* scheme. From the numerical results, it is clearly shown that the extended trapezoidal method is suitable to solve VIDEs.

REFERENCES

[1] A. M. Wazwaz, *A First Course in Integral Equations*. Singapore: World Scientific Publishing Company, 1997.
[2] J. T. Day, "Note on the numerical solution of integro-differential equations," *Computer Journal*, vol. 9 no. 4, pp. 394–395, 1967.
[3] P. Linz, "Linear multistep methods for Volterra integro-differential equations," *Journal of the Association for Comp. Machinery*, vol. 16 no. 2, pp. 295–301, 1969.
[4] P. J. van der Houwen, and H. J. J. Riele, "Linear multistep methods for Volterra integral and integro-differential equations," *Mathematics of Computation*, vol. 45 no. 172, pp. 439–461, 1985.
[5] K. Maleknejad, and M. Shahrezaee, "Using Runge-Kutta method for numerical solution of the system of Volterra integral equations,"*Applied Mathematics and Computation*, vol. 149 no.2, pp. 399–410, 2004.
[6] M. Gachpazan, "Numerical scheme to solve integro-differential equations system," *Journal of Advanced Research in Scientific Computing*, vol. 1 no. 1, pp. 11–21, 2009.
[7] S. K. Vanani, and A. Aminataei, "Numerical solution of Volterra integro-differential equations," *J. of Comp. Analysis and Applications*, vol. 13, pp. 654–662, 2011.
[8] F. Mazzia, and A. M. Nagy, "Solving Volterra integro-differential equations by variable stepsize bock BS methods: properties and implementation techniques," *Applied Mathematics and Computation*, vol. 239, pp. 198–210, 2014.
[9] A. Filiz, "Numerical solution of linear Volterra integro-differential equations using Runge-Kutta-Fehlberg method," *Applied and Computational Mathematics*, vol. 3 no. 1, pp. 9–14, 2014.
[10] L. Zhang, and F. Ma, "Pouzet-Runge-Kutta-Chebyshev method for Volterra equations of the second kind," *Journal of Computational and Applied Mathematics*, vol. 288, pp. 323–331, 2015.
[11] R. A. Usmani, and R. P. Agarwal, "An A-stable extended trapezoidal rule for the integration of ordinary differential equations," *Computer and Mathematics with Applications*, vol. 11 no. 12, pp. 1183–1191, 1985.
[12] M. M. Chawla, and M. A. Al-Zanaidi, and M. G. Al-Aslab, "A class of stabilized extended one-step methods for the numerical solution of ordinary differential equations," *Computers and Mathematics with Applications*, vol. 29 no. 10, pp. 79–84, 1995.
[13] F. Ibrahim, A. A. Salama, A. Quazzi, and S. Turek, "Extended one-step methods for solving delay differential equations," *Applied Mathematics*, vol. 8 no. 3, pp. 941–948, 2014.
[14] F. Ibrahim, A. A. Salama, and S. Turek, "A class of extended one-step methods for solving delay differential equations," *Applied Mathematics*, vol. 9 no. 2, pp. 593–602, 2015.
[15] F. Ibrahim, F. A. Rihan, and S. Turek, "Extended one-step schemes for stiff and non-stiff delay differential equations," extracted from http://www.mathematik.tu-dortmund.de/papers/IbrahimRihanTurek2015.pdf
[16] I. B. Jacques, "Extended one-step methods for the numerical solution of ordinary differential equations," *Int. Journal of Computer Mathematics*, vol. 29 no. 2-4, pp. 247–255, 1989.
[17] A. M. Wazwaz, *Linear and Nonlinear Integral Equations*. London: Springer Publishing Company, 2011.