

New Development of Extended Trapezoidal Method for Solving First Order Linear Volterra Integro-Differential Equations

Fuziyah Ishak* and Muhammad Nur Firdaus Selamat

Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia

Modelling real life phenomena in the area of fluid dynamics, biological models and chemical kinetics give rise to Volterra integro-differential equations. As most of these equations cannot be solved analytically, efficient numerical methods are sought in approximating the solutions accurately. Trapezoidal, as well as extended trapezoidal methods are known to be efficient and accurate methods in solving many types of differential equations. These methods can be efficiently adapted to solve Volterra integro-differential equations to achieve required accuracy. In this research, new development on the extended trapezoidal method for solving first order linear Volterra integro-differential equations is considered. Previous numerical results are compared with the new development and it is shown that the extended trapezoidal method is efficient and suitable in solving the integro-differential equations.

Keywords: Volterra integro-differential equations; mathematical models; numerical methods; extended Trapezoidal method; accuracy

I. INTRODUCTION

In this paper, we focus on numerical solutions for Volterra integro-differential equations (VIDEs) of the form:

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

$$y(x_0) = y_0,$$

where the given functions $f(x, y(x))$ and $F(x, s, y(s))$ in (1) satisfy Lipschitz conditions in their arguments such that the solution $y(x)$ exists. The value y_0 is the given initial condition. It is well known that VIDEs play important roles in modelling real life phenomena in various areas of applications such as heat transfer, diffusion process in general, neutron diffusion and many more, see (Wazwaz, 1997). Finding solutions for these models is becoming very important and for decades it has attracted scientists and engineers in looking for various solutions approaches.

Because of the complexity in VIDEs, analytical solutions are very hard and at times impossible to obtain. As an alternative, scientists and engineers seek numerical solutions that can be made as accurately as possible.

Conventionally, numerical methods for solving ordinary differential equations (ODEs) are adapted to solve VIDEs where an 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 Day (1967); Linz (1969); Van der Houwen & Riele (1985); Maleknejad & Shahrezaee (2004); Gachpazan (2009); Vanani & Aminataei (2011); Mazzia & Nagy (2014); Filiz (2014); and Agbolae & Anake (2017).

In this paper, we extend the work that was proposed by (Ishak & Ahmad, 2016) in solving VIDEs using

*Corresponding author's email: fuziyah@tmsk.uitm.edu.my

extended one-step trapezoidal method. In (Ishak & Ahmad, 2016), an integral part of (1) was solved using a lower order method than the method used for differential part. Solving both the integral and differential parts using the same method and of the same order will produce better results in terms of its accuracy. The extended method with higher order of convergence and improved stability conditions is suitable for solving many types of differential equations. Previous research has shown extensive implementations of extended one-step methods in solving various types of differential equations including ODEs and delay differential equations. The extended methods are also implemented in combination scheme for solving stiff and non-stiff differential equations. Please refer to published work on various implementations of extended one-step methods for numerical solutions of differential equations, see for example (Usmani & Agarwal, 1985); (Chawla, Al-Zanaidi & Al-Aslab, 1995); (Ibrahim et al., 2014); (Ibrahin, Salama & Turek, 2015); (Ibrahim, Rihan & Turek, 2014). 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 (Usmani & Agarwal, 1985) to solve ODEs with third order convergence while preserving the property of A -stability of the classical trapezoidal method. As cited in (Usmani & Agarwal, 1985), Dahlquist stated in 1963 that a method is to be A -stable if the numerical solution of differential equation $y' = \lambda y$ where $\text{Re}(\lambda) < 0$ approaches zero as step size approaches zero.

The organization of this paper is as follows. In section 2, we discuss the development of the proposed method. Numerical results and related discussions are presented in section 3. Section 4 highlights the summary.

II. MATERIALS AND METHODS

We consider the initial value problem (IVP) for VIDEs as the following:

$$y'(x) = f(x, y(x)) + \int_{x_0}^x F(x, s, y(s)) ds, \quad x_0 \leq x \leq x_N, \quad (2)$$

$$y(x_0) = y_0,$$

where the interval $[x_0, x_N]$ is divided into N subintervals

with step size $h = \frac{x_N - x_0}{N}$. The notation y_n refers to

the approximate solution for $y(x_n)$ where y is the solution for the IVP in (2). The grids $x_n = x_0 + ih, i = 0, 1, 2, \dots, N$ 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} and \hat{y}_{n+1} , where y_{n+1} is the approximate solution for $y(x_{n+1})$ and \hat{y}_{n+1} is the predicted value for y_{n+1} are implicit and being implemented in *PECE* mode where *P* stands for predict, *E* stands for function evaluation and *C* is for correct.

We derive the formulae by integrating the differential equation in (2) 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

$$I_1 = \int_{x_n}^{x_{n+1}} f(x, y(x)) dx \quad (3)$$

and

$$I_2 = \int_{x_n}^{x_{n+1}} \int_{x_0}^x F(x, s, y(s)) ds dx. \quad (4)$$

The integral I_1 in (3) 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

$$I_1 = h \int_0^1 P(S) dS$$

$$= h \int_0^1 \left(\frac{S^2 - 3S + 2}{2} f_n + (-S^2 + 2S) f_{n+1} + \frac{S^2 - S}{2} f_{n+2} \right) dS$$

$$= \frac{h}{12} (5f_n + 8f_{n+1} - f_{n+2})$$

which the coefficients are as given in (Usmani & Agarwal, 1985) and (Jacques, 1989). 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 (Usmani & Agarwal, 1985) as follows:

$$\hat{y}_{n+1}^{(0)} = y_n + hf_n,$$

$$\hat{y}_{n+1}^{(1)} = y_n + \frac{h}{2} [f_n + f(x_{n+1}, \hat{y}_{n+1}^{(0)})],$$

$$\hat{y}_{n+1} = \hat{y}_{n+1}^{(1)},$$

$$\hat{y}_{n+2} = 5y_n - 4y_{n+1} + h(2f_n + 4f_{n+1}).$$

In similar manner we obtain the integral I_2 . Let

$$z(x) = \int_{x_0}^x F(x, s, y(s)) ds. \tag{5}$$

The notation z_n means $z_n = \int_{x_0}^{x_n} F(x, s, y(s)) ds$. The integral I_2 in (4) is solved by first interpolating $z(x)$ in (5) by $\bar{P}(x)$ using interpolation points (x_n, z_n) , (x_{n+1}, z_{n+1}) and (x_{n+2}, z_{n+2}) . The polynomial $\bar{P}(x)$ is given by

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

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

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

Using $x = x_n + Sh$, we have

$$I_2 = h \int_0^1 \bar{P}(S) dS$$

$$= h \int_0^1 \left(\frac{S^2 - 3S + 2}{2} z_n + (-S^2 + 2S) z_{n+1} + \frac{S^2 - S}{2} z_{n+2} \right) dS$$

$$= \frac{h}{12} (5z_n + 8z_{n+1} - z_{n+2})$$

For simplicity, we write

$$I_2 = \frac{5}{12} h z_n + \frac{8}{12} h z_{n+1} - \frac{1}{12} h z_{n+2}$$

$$= I_{21} + I_{22} + I_{23}$$

where

$$I_{21} = \frac{5}{12} h \int_{x_0}^{x_n} F(x_n, s, y(s)) ds$$

$$= \frac{5}{12} h \left[\frac{h}{2} F(x_n, x_0, y_0) + h F(x_n, x_1, y_1) + \dots \right. \\ \left. + h F(x_n, x_{n-1}, y_{n-1}) + \frac{h}{2} F(x_n, x_n, y_n) \right]$$

$$= \frac{5h^2}{24} [F(x_n, x_0, y_0) + 2F(x_n, x_1, y_1) + \dots \\ + 2F(x_n, x_{n-1}, y_{n-1}) + F(x_n, x_n, y_n)]$$

and

$$I_{22} = \frac{8}{12} h \int_{x_0}^{x_{n+1}} F(x_{n+1}, s, y(s)) ds$$

$$= \frac{8}{12} h \left[\frac{h}{2} F(x_{n+1}, x_0, y_0) + h F(x_{n+1}, x_1, y_1) + \dots \right. \\ \left. + h F(x_{n+1}, x_n, y_n) + \frac{h}{2} F(x_{n+1}, x_{n+1}, \hat{y}_{n+1}) \right]$$

$$= \frac{8h^2}{24} [F(x_{n+1}, x_0, y_0) + 2F(x_{n+1}, x_1, y_1) + \dots \\ + 2F(x_{n+1}, x_n, y_n) + F(x_{n+1}, x_{n+1}, \hat{y}_{n+1})]$$

and

$$I_{23} = -\frac{1}{12} h \int_{x_0}^{x_{n+2}} F(x_{n+2}, s, y(s)) ds$$

$$= -\frac{1}{12} h \left[\frac{h}{2} F(x_{n+2}, x_0, y_0) + h F(x_{n+2}, x_1, y_1) + \dots \right. \\ \left. + h F(x_{n+2}, x_{n+1}, y_{n+1}) + \frac{h}{2} F(x_{n+2}, x_{n+2}, \hat{y}_{n+2}) \right]$$

$$= \frac{8h^2}{24} [F(x_{n+2}, x_0, y_0) + 2F(x_{n+1}, x_1, y_1) + \dots \\ + 2F(x_{n+2}, x_{n+1}, y_{n+1}) + F(x_{n+2}, x_{n+2}, \hat{y}_{n+2})]$$

The derivatives are evaluated as follows,

$$y'_0 = f(x_0, y_0),$$

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

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

The development of the algorithm is shown in the following Figure 1.

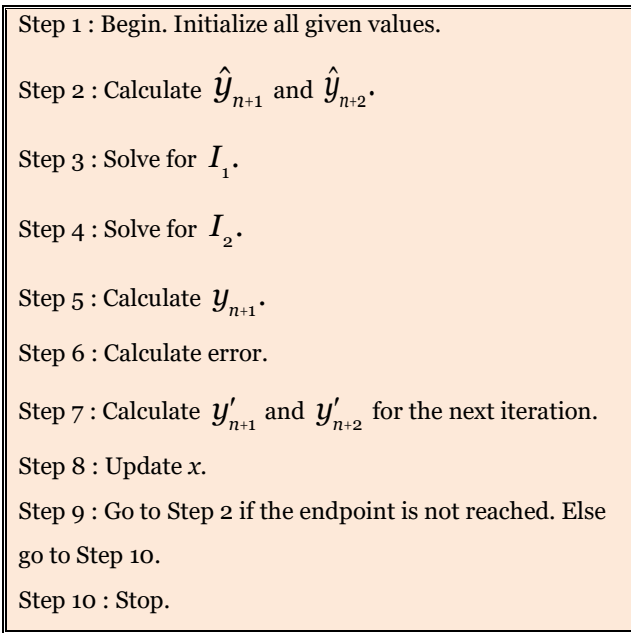


Figure 1. Algorithm for solving VIDEs

To analyse the accuracy of the method, the above algorithm is tested with test problems having exact solutions. For illustrative purposes, the results of Test Problem 1 and Test Problem 2 are given in the next section.

Test Problem 1 (Source, Day (1967))

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

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

Test Problem 2 (Source, Day (1967))

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

$$y(0) = 1, \quad 0 \leq x \leq 1.$$

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

III. RESULTS AND DISCUSSIONS

In this section, numerical results for Test Problem 1 and Test Problem 2 are given in Table 1 – Table 6. The results are displayed in terms of absolute errors between approximate and exact solutions at different grid points using various stepsizes. We also compare the numerical results of the method described in the previous section with Trapezoidal method and Extended Trapezoidal in (Ishak & Ahmad, 2016). Results for Test Problem 1 for stepsizes $h = 0.1$, $h = 0.025$, and $h = 0.01$ are given in Table 1, Table 2 and Table 3 respectively. Moreover, results for Test Problem 2 for stepsizes $h = 0.1$, $h = 0.025$, and $h = 0.01$ are given in Table 4, Table 5 and Table 6 respectively. The abbreviations TM refers to Trapezoidal method, ETM refers to extended Trapezoidal method implemented in (Ishak & Ahmad, 2016), NETM refers to the implementation method described in the previous section, and 5.4983×10^{-4} is the same as

Table 1. Results for Test Problem 1, $h = 0.1$

x	TM	ETM	NETM
0.1	8.4317e-05	8.2792e-05	1.2502e-07
0.2	1.6558e-04	1.6310e-04	7.3015e-07
0.3	2.4402e-04	2.3848e-04	2.5669e-06
0.4	3.1661e-04	3.0657e-04	6.4301e-06
0.5	3.8012e-04	3.6514e-04	1.3074e-05
0.6	4.3328e-04	4.1210e-04	2.3195e-05
0.7	4.7376e-04	4.4558e-04	3.7422e-05
0.8	4.9978e-04	4.6392e-04	5.6296e-05
0.9	5.0976e-04	4.6571e-04	8.0264e-05
1.0	5.0243e-04	4.4985e-04	1.0966e-04

Table 2. Results for Test Problem 1, $h = 0.025$

x	TM	ETM	NETM
0.1	5.1916e-06	5.1818e-06	6.3815e-09
0.2	1.0247e-05	1.0208e-05	5.8198e-08
0.3	1.5012e-05	1.4926e-05	2.0634e-07
0.4	1.9339e-05	1.9187e-05	4.9983e-07
0.5	2.3086e-05	2.2852e-05	9.8478e-07
0.6	2.6120e-05	2.5790e-05	1.7034e-06
0.7	2.8323e-05	2.7883e-05	2.6932e-06
0.8	2.9588e-05	2.9027e-05	3.9860e-06
0.9	2.9825e-05	2.9136e-05	5.6073e-06
1.0	2.8961e-05	2.8139e-05	7.5758e-06

Table 3. Results for Test Problem 1, $h = 0.01$

x	TM	ETM	NETM
0.1	8.2978e-07	8.2916e-07	1.2051e-09
0.2	1.6359e-06	1.6334e-06	1.0290e-08
0.3	2.3938e-06	2.3883e-06	3.5392e-08
0.4	3.0799e-06	3.0702e-06	8.4328e-08
0.5	3.6715e-06	3.6565e-06	1.6443e-07
0.6	4.1478e-06	4.1266e-06	2.8242e-07
0.7	4.4897e-06	4.4615e-06	4.4421e-07
0.8	4.6805e-06	4.6447e-06	6.5484e-07
0.9	4.7061e-06	4.6620e-06	9.1831e-07
1.0	4.5551e-06	4.5025e-06	1.2375e-06

Table 4. Results for Test Problem 2, $h = 0.1$

x	TM	ETM	NETM
0.1	5.4983e-04	1.9968e-04	4.7572e-04
0.2	1.1757e-03	5.0620e-04	5.0519e-04
0.3	1.9220e-03	9.2292e-04	5.6263e-04
0.4	2.8533e-03	1.4871e-03	6.5502e-04
0.5	4.0648e-03	2.2602e-03	7.9663e-04
0.6	5.6984e-03	3.3398e-03	1.0067e-03
0.7	7.9682e-03	4.8800e-03	1.3124e-03
0.8	1.1201e-02	7.1225e-03	1.7533e-03
0.9	1.5901e-02	1.0450e-02	2.3883e-03
1.0	2.2855e-02	1.5471e-02	3.3067e-03

Table 5. Results for Test Problem 2, $h = 0.025$

x	TM	ETM	NETM
0.1	3.3444e-05	1.3122e-05	5.9860e-06
0.2	7.1899e-05	3.2014e-05	6.5272e-06
0.3	1.1815e-04	5.7857e-05	7.3872e-06
0.4	1.7631e-04	9.2945e-05	8.6529e-06
0.5	2.5246e-04	1.4108e-04	1.0380e-05
0.6	3.5577e-04	2.0834e-04	1.2552e-05
0.7	5.0017e-04	3.0426e-04	1.5012e-05
0.8	7.0700e-04	4.4386e-04	1.7348e-05
0.9	1.0094e-03	6.5081e-04	1.8701e-05
1.0	1.4593e-03	9.6284e-04	1.7435e-05

Table 6. Results for Test Problem 2, $h = 0.01$

x	TM	ETM	NETM
0.1	5.3182e-06	2.0968e-06	3.6613e-07
0.2	1.1445e-05	5.1090e-06	4.0079e-07
0.3	1.8827e-05	9.2335e-06	4.5246e-07
0.4	2.8121e-05	1.4837e-05	5.1900e-07
0.5	4.0308e-05	2.2526e-05	5.8592e-07
0.6	5.6862e-05	3.3270e-05	6.1545e-07
0.7	8.0026e-05	4.8595e-05	5.2893e-07

0.8	1.1324e-04	7.0898e-05	1.7754e-07
0.9	1.6186e-04	1.0396e-04	7.0676e-07
1.0	2.3428e-04	1.5382e-04	2.5949e-06

From the tabulated results, it is clearly seen that the extended Trapezoidal method as described in the previous sections achieves more than 100% of the desired accuracy. As the stepsize gets smaller, the errors at the respective grid points become smaller as well. Thus, it can be concluded that NETM is suitable for solving VIDEs. Moreover, the methods give better accuracy as compared with Trapezoidal and extended Trapezoidal as in (Ishak & Ahmad, 2016).

IV. SUMMARY

We have presented the new development of extended Trapezoidal method for solving VIDEs. The implicit formulae adapted from extended Trapezoidal method for solving ODEs are implemented in PECE mode for both differential and integral parts.

Absolute errors between exact and approximate solutions are presented at selected grid points with different stepsizes. The errors indicate that the new implementation of the method achieves the desired accuracy as the errors become smaller when the stepsize becomes smaller. The extended formulae used in solving an integral part of the VIDEs also improve the performance of the method in terms of its accuracy.

From the numerical results and comparison between the existing methods, the new development of the extended Trapezoidal method is suitable for solving VIDEs numerically since the errors become smaller when the stepsize is getting smaller. Thus, the extended implementation of the method produces accurate approximate solutions as the approximate solution approaches exact solution as stepsize tends to zero.

V. REFERENCES

- 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.
- A. M. Wazwaz, *A First Course in Integral Equations*. Singapore: World Scientific Publishing Company, 1997.
- 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.
- 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.
- 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>
- 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.
- F. Ishak, and S. N. Ahmad, "Development of extended Trapezoidal method for numerical solution of Volterra integro-differential equations," *Int. Jour. of Mathematical, Computational, Physical, Electrical and Computer Engineering*, vol. 10, no. 11, pp. 524–527, 2016.
- 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.
- J. T. Day, "Note on the numerical solution of integro-differential equations," *Computer Journal*, vol. 9 no. 4, pp. 394–395, 1967.
- 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.
- 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.
- 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.
- 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.
- O. A. Agbolae, and T. A. Anake, "Solutions of first-order Volterra type linear integrodifferential equations by collocation method," *Journal of Applied Mathematics*, 2017, Article ID 1510267
<https://doi.org/10.1155/2017/1510267>
- 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.
- 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.
- 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.
- 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.