A Fifth-order Numerical Convergence for Linear Volterra Integro-differential Equation

Ali FILIZ[1], Ali ISIK[1], Mehmet EKICI[2]

[1.] Department of Mathematics, Adnan Menderes University, 09010 AYDIN-Turkey
[2.] Department of Mathematics, Bozok University, 66100 Yozgat-Turkey
E-mail: afiliz@adu.edu.tr

Abstract: In this paper a new fifth-order numerical solution of linear Volterra integro-differential equation is discussed. Example of this question has been solved numerically using the Runge-Kutta-Verner method for Ordinary Differential Equation (ODE) part and Newton-Cotes formulae (quadrature rules) for integral parts. Finally, a new fifth-order routine is devised for numerical solution of the linear Volterra integro-differential equation.
[FILIZ A, ISIK A, EKICI M. **A Fifth-order Numerical Convergence for Linear Volterra Integro-differential Equation**. *Life Sci J* 2013;10(4):302-309] (ISSN:1097-8135). http://www.lifesciencesite.com. 40

Keywords: A fifth-order accuracy; Lagrange polynomial interpolating; quadrature formulae; Runge-Kutta methods; Volterra integro-differential equation.

1. Introduction

A functional equation in which the unknown function appears in the form of it is a derivative as well as under the integral sign is called an integro-differential equation (see, filiz(2000a; 2013) and Volterra(1931; 1959; 1957)). In this paper we will consider the linear Volterra integro-differential equation of the form (see, Asanov, 1978; Baker, 1978; Bellman, 1949; Cooke, 1966).

$$u'(t) = F\Big(t, u(t), \int_{t_0}^{t} K(t,s)\,u(s)\,ds\Big), \tag{1}$$
$$u(t_0) = u_0, \quad t \ge t_0,$$

with the kernel $K(t,s)$ of equation (1) assumed to be continuous on $[t_0, T]$ ($t > t_0$, T a finite) and $S = \{(t,s) \in R \times R : t \le s \le t \le T\}$. In this paper we consider in detail only one case, the question

$$u'(t) = \mu + \beta u(t) + \lambda \int_{t_0}^{t} K(t,s)\,u(s)\,ds), \quad t \ge t_0, \tag{2}$$

with initial condition

$$u(t_0) = u_0. \tag{3}$$

Equation (1) can be solved numerically using various methods (see, Baker et al., 1998; Filiz, 2000b; Baker et al., 2006; Filiz, 2000a; Filiz, 2013; Linz, 1985). In this paper $u(t_n)$ will denote the exact value of u at $t_n = t_0 + nh$. We shall use $\tilde{u}(t_n)$ or $\tilde{u}_n$ to denote a numerical solution u of at t_n. However, in this paper we will construct fifth-order numerical method for equation (1). Since the integral cannot be determined explicitly, it may be approximated using familiar numerical integration methods. The Newton-Cotes integration formulae, which include the 2-point closed Newton-Cotes formula is called the trapezoidal rule, the 3-point rule is known as Simpson's 1/3 rule, the 4-point closed rule is Simpson's 3/8 rule, the 5-point closed rule is Boole's rule (Bode's rule), Weddle's rule, higher rules include the 6-point, 7-point and 8-point are well suited here since they use nodes which were given in (e.g., Wolfram, 2013; Filiz, 2000a; Abramowitz and Stegun, 1972; Ueberhuber, 1972) and (see, Baker, 1978; Linz, 1985). In (Filiz, 2000b; Filiz, 2000a; Filiz, 2013) we consider an elementary class of formulae for the numerical solution of integro-differential equation of first-order and second-order, based upon the $\theta-$method (see Table 1).

2. The Numerical of Integro-differential Equations

The functional equation (1) is a first-order Volterra integro-differential equation; here, one usually looks for a solution which satisfies the initial condition $u(t_0) = u_0$.

Definition 1. (Linear kernel) A Volterra integro-differential equation is said to be linear if its kernel has the form $K(t,s,u(s)) = K(t,s)u(s)$.

Definition 2. (Convolution kernel) If the kernel of (1) is a function of $(t-s)$ only, that is $K(t,s) = k(t-s)$, then K is said to be a difference (convolution) kernel.

Since a nonlinear Volterra integro-differential equations is characterized by two functions, namely, $F(t,u(t),z(t))$ (differential part) and $K(t,s,u(s))$ (integral part) a corresponding existence and uniqueness theorem is an extension of the analogous for initial value problem (IVP) for first-order ordinary differential equations and for Volterra integral equation of the second kind.

Theorem 1. (Existence and uniqueness) In equation (1) suppose that $F(t,u(t),z(t))$ and $K(t,s,u(s))$ are, respectively, continuous for $t \in [t_0, T]$ and $(t,s) \in S$, and let the following (uniform) Lipschitz condition hold:

(i) $|F(t,u_1(t),z(t)) - F(t,u_2(t),z(t))| \le L_1 |u_1(t) - u_2(t)|$,

(ii) $|F(t,u(t),z_1(t)) - F(t,u(t),z_2(t))| \le L_2 |z_1(t) - z_2(t)|$,

(iii) $|K(t,s,u_1(t)) - K(t,s,u_2(t))| \le L_3 |u_1(t) - u_2(t)|$,

for all $t \in [t_0, T]$, $(t,s) \in S$, and $|u_i(t)| < \infty$, $|z_i(t)| < \infty$ $(i = 1,2)$. Then each u_0 there exist exactly one solution $u(t) \in C^1([t_0, T])$ of equation (1) satisfying $u(t_0) = u_0$.

Proof. (See (Linz, 1985)).

In general formulae for the numerical solution of integro-differential equations rely upon formulae for the underlying Ordinary Differential Equation (ODE), combined with auxiliary quadrature rules approximation of

(4) $$\tilde{z}(t_n) := h \sum_{j=0}^{n} \omega_{n,j} k(t_n - t_j)\, \tilde{u}(t_j) \approx \int_{t_0}^{t_n} k(t-s)\, u(s)\, ds.$$

For equation (1), we adapted the θ-method in filiz (2000a; 2013) and method convergence O(h) and $O(h^2)$ respectively.

Of course, whereas we have defined approximations $\tilde{z}(t_n)$ in terms of quadrature rules that reflect the underlying ODE method, it is in principle possible to “mix and match”. The combinations of formulae can be chosen on the basis of order of convergence. There are two directions in which $\theta-$ method can be generalized. The first involves adapting Linear Multistep Method (LMM) for ODE’s and second involves adapting Runge-Kutta methods. In each case, we will require to approximate integral terms (4) at selected values at t. Equation (1) can be solved various methods. In this paper we shall focus on fifth-order numerical method for equation (1). The integral term cannot be determined explicitly; it may be approximated using familiar numerical integration methods. The Newton-Cotes integration formulae, which include left and right rectangle rules, the trapezoidal rule, Simpson’s 1/3 rule and Simpson’s 3/8 rule are well suited here since they used nodes which were previously calculated:

$$\tilde{z}(t_n) := h \sum_{j=0}^{n} \omega_{n,j} k(t_n - t_j)\, \tilde{u}(t_j) \approx \int_{t_0}^{t_n} k(t-s)\, u(s)\, ds,$$

where $\omega_{n,j}$ are the appropriate coefficients for the composite integration schemes chosen. A combination of integration method may be used. Simpson’s 1/3 rule requires that n , the number of subintervals dividing $[t_0, t_n]$, be even. Therefore, Simpson’s 1/3 rule cannot be used at each step. When n is odd, one method is to use Simpson’s 1/3 rule on $[t_0, t_{n-1}]$, and trapezoidal rule on $[t_{n-1}, t_n]$, , adding the results to approximate the integral on $[t_0, t_n]$. Another method is to use the trapezoidal rule $[t_0, t_1]$, and Simpson’s 1/3 rule thereafter.

3. Convergence and Order of Convergence

If we use time discretization in (2), we get

(5) $$\tilde{u}(t_{n+1}) = \tilde{u}(t_n) + h(\mu + \beta\, \tilde{u}(t_n) + \lambda h \sum_{j=0}^{n} \omega_{n,j}\, k(t_n - t_j)\, \tilde{u}(t_j)).$$

Given an interval $[t_0, T]$ introduce $h = (T - t_0)/n$ for some $n \in N$ over mesh-points

$t_n = t_0 + nh$, n=0, 1, 2, 3, ….,n.

For the integro-differential equation with the unique solution $u(t)$ suppose $\tilde{u}(t_i)$, i = 0, 1, 2, 3, · · ·, n are computed by some approximation scheme. We have convergence of order p for mesh-points in $[t_0, T]$ using step-size h

$$\sup_{t_0 \le t_i \le T} |\tilde{u}(t_i) - u(t_i)|,$$

as $h \to 0$. Consider the set of values $e_i = \tilde{u}(t_i) - u(t_i),\ t_i = t_0 + ih,\ i = 1,2,3,....,n,$ which is called discretization error of the approximate solution $\tilde{u}(t_i)$ at the mesh-points $\{t_i\}$.

Definition 3. (Convergence) A method of the form (5) is said to be convergent on $[t_0, T]$ if

$$\lim_{h \to 0} \left(\max_{t_i \in [t_0, t]} |\tilde{u}(t_i) - u(t_i)| \right) = 0 .$$

Definition 4. (Order of convergence) If, for all h, there exists a number M < ∞, independent of h, such that if

$$\max_{0 \le i \le n} |\tilde{u}(t_i) - u(t_i)| \le M\, h^p ,$$

and if p is the largest number for which such an inequality holds, then p is called the order of convergence of the method.

4. The Fifth-order Numerical Routine for Linear Volterra Integro-differential Equation

Now consider the non-dimensional problem (1). In order to solve (1) numerically, we purpose the use of two methods familiar to most mathematicians. We consider methods which approximate the solution the initial value problem (IVP)

$$u'(t) = f(t, u(t)\ ),\ \ u(t_0) = u_0 ,$$

at time $t_n = t_0 + nh$, n=0, 1, 2, 3, …., where $h = t_n - t_{n-1}$ is the constant nodal step-size and, in the Example 1,

$$F(t, u(t), \int_{t_0}^{t} k(t-s)\ u(s)\ ds) =$$

$$\mu + \beta u(t) + \lambda \int_{0}^{t} k(t-s)\ u(s)\ ds.$$

For example, the explicit Euler method approximates the solution to Example 1 at t_{n+1}

$$\tilde{u}_{n+1} = \tilde{u}_n + h\left(\mu + \beta \tilde{u}_n + \lambda \int_{0}^{t_n} e^{-\delta(t-s)}\ u(s)\ ds\right).$$

The explicit finite difference method given in (Filiz,2013) as applied to equation (1) easily extended to more accurate predictor-corrector method. The predictor step uses $(\tilde{u}_{n+1} = \tilde{u}_n + h(\ F(t_n, \tilde{u}_n, \tilde{z}(t_n))\ ))$ to obtain $\tilde{u}^{\kappa}_{n+1}$, which is followed by the corrector step, which uses higher order trapezoidal method

$$(6)\quad \tilde{u}_{n+1} = \tilde{u}_n + h\left(\frac{1}{2}F\begin{pmatrix} t_n, \tilde{u}_n, \tilde{z}(t_n)) \\ + \frac{1}{2}F(t_{n+1}, \tilde{u}^{\kappa}_{n+1}, \tilde{z}(t_{n+1})\end{pmatrix}\right),$$

where

$$\tilde{z}_0 = 0,$$

$$(7)\quad \tilde{z}_n = \frac{h}{2}k(t_n - t_0)\tilde{u}_0 + h\sum_{j=1}^{n-1} k(t_n - t_j)\tilde{u}_j$$

$$+ \frac{h}{2}k(t_n - t_n)\tilde{u}_n, n = 1,2,3,....$$

This procedure is sometimes referred to as modified Euler method (second order Runge-Kutta-RK2) and is one order magnitude more accurate than the explicit Euler method.

Theorem 2. (Second order convergence) If conditions (3), Theorem 1-(i) and Theorem 1-(iii) are satisfied, and if in addition F and K are twice continuously differentiable with respect to all arguments, then approximate solution defined by (6) and (7) converges to the true solution of (7) with order two.

Proof. (See (Linz, 1985)).

At each step the equation (6) was solved by the trapezoidal method. The results are shown in Table 3. The apparent order of convergence is two, which is not surprising because of the use trapezoidal method.

Higher order methods can be constructed along similar lines. The 5-point extended closed rule is Boole's method may be devised on $[t_0, t_n]$ as following:

If n=0, $\tilde{z}_0 = 0$,

If n=1, use

$$\tilde{z}_1 = \frac{h}{2}k(t_1 - t_0)\tilde{u}_0 + \frac{h}{2}k(t_1 - t_1)\tilde{u}_1,$$

If n=2, use

$$\tilde{z}_2 = \frac{h}{3}(k(t_2 - t_0)\tilde{u}_0 + 4k(t_2 - t_1)\tilde{u}_1 + k(t_2 - t_2)\tilde{u}_2),$$

If n=3

$$\tilde{z}_3 = \frac{h}{3}(k(t_3 - t_0)\tilde{u}_0 + 3k(t_3 - t_1)\tilde{u}_1 + 3k(t_3 - t_2)\tilde{u}_2$$
$$+ k(t_3 - t_3)\tilde{u}_3\ ),$$

If n=4

$$\tilde{z}_4 = \frac{2h}{45}(7k(t_4 - t_0)\tilde{u}_0 + 32k(t_4 - t_1)\tilde{u}_1 + 12k(t_4 - t_2)\tilde{u}_2$$
$$+ 32k(t_4 - t_3)\tilde{u}_3 + 7k(t_4 - t_4)\tilde{u}_4),$$

If $n \geq 4$, then

$$\tilde{z}_n = \tilde{z}_{n-4} + \frac{2h}{45}(7k(t_{n-4} - t_0)\tilde{u}_0 + 32k(t_{n-3} - t_1)\tilde{u}_1$$
$$(8)\quad + 12k(t_{n-2} - t_2)\tilde{u}_2 + 32k(t_{n-1} - t_3)\tilde{u}_3$$
$$+ 7k(t_n - t_4)\tilde{u}_4)\quad, n = 4, 5,\ 6, 7,\ ....$$

The fourth order classical Runge-Kutta method (RK4) can also be adapted to the numerical solution of equation (1) (see Table 3). Stepping from $\tilde{u}_n$ with step-size h to obtain $\tilde{u}_{n+1}$, the RK4 method as applied to this problem in filiz(2000a; 2013).

The **fifth-order** Runge-Kutta-Fehlberg (see Table 2) and **sixth order** Runge-Kutta-Verner methods (see Table 1) may be used but not readily (see, (Burden and Faires, 1997)), since the intranodal evaluation points are uniformly spaced. Consequently, the integrals needed during the intermediate calculations to step from t_n to t_{n+1} may require the trapezoidal rule or Lagrange polynomial interpolating integration on a non-uniform partition $[t_n, t_{n+1}]$.

Other high-order finite difference methods which may be used here include the Adam-Basforth multistep methods. One such fourth order method is described in (see, (Burden and Faires, 1997)). It uses the RK4 method to obtain the starting values u_0, u_1, u_2. Thereafter, the method uses the fourth order explicit Adam-Basforth method as a predictor and fourth-order implicit Adam-Moulton method corrector to step from t_n to t_{n+1}.

Runge-Kutta-Verner method (RKV) can also be adapted to the numerical solution of (1). Stepping from $\tilde{u}_n$ with step-size h to obtain $\tilde{u}_{n+1}$, the RKV method as applied to this problem may be written as:

$$k_1 = hF(t_n, \tilde{u}_n, \tilde{z}(t_n)),$$

$$\tilde{u}^a_{n+1/6} = \tilde{u}_n + \frac{k_1}{6},$$

$$k_2 = hF(t_{n+1/6}, \tilde{u}^a_{n+1/6}, \tilde{z}_{n+1/6}),$$

$$k_2 = hF\left(t_{n+1/6}, \tilde{u}^a_{n+1/6}, \tilde{z}_n + \frac{h}{12}\left[\tilde{u}_n + \tilde{u}^a_{n+1/6}\right]\right),$$

(9) $$\tilde{u}^b_{n+4/15} = \tilde{u}_n + \frac{4k_1}{75} + \frac{16k_2}{75},$$

$$k_3 = hF(t_{n+4/15}, \tilde{u}^b_{n+4/15}, \tilde{z}_{n+4/15}),$$

$$k_3 = hF\left(t_{n+4/15}, \tilde{u}^b_{n+4/15}, \tilde{z}_n + \frac{4h}{30}\left[\tilde{u}_n + \tilde{u}^b_{n+4/15}\right]\right),$$

$$\tilde{u}^c_{n+2/3} = \tilde{u}_n + \frac{5k_1}{6} - \frac{8k_2}{3} + \frac{5k_3}{2},$$

$$k_4 = hF(t_{n+2/3}, \tilde{u}^c_{n+2/3}, \tilde{z}_{n+2/3}),$$

$$k_4 = hF\left(t_{n+2/3}, \tilde{u}^c_{n+2/3}, \tilde{z}_n + \frac{2h}{6}\left[\tilde{u}_n + \tilde{u}^c_{n+2/3}\right]\right),$$

$$\tilde{u}^d_{n+5/6} = \tilde{u}_n + \frac{165k_1}{64} + \frac{55k_2}{6} - \frac{425k_3}{64} + \frac{85k_4}{96},$$

$$k_5 = hF(t_{n+5/6}, \tilde{u}^d_{n+5/6}, \tilde{z}_{n+5/6}),$$

$$k_5 = hF\left(t_{n+5/6}, \tilde{u}^d_{n+5/6}, \tilde{z}_n + \frac{5h}{12}\left[\tilde{u}_n + \tilde{u}^d_{n+5/6}\right]\right),$$

$$\tilde{u}^e_{n+1} = \tilde{u}_n + \frac{12k_1}{15} - 8k_2 + \frac{4015k_3}{612} - \frac{11k_4}{36} + \frac{88k_5}{255},$$

$$k_6 = hF(t_{n+1}, \tilde{u}^e_{n+1}, \tilde{z}_{n+1}),$$

$$k_6 = hF\left(t_{n+1}, \tilde{u}^e_{n+1}, \tilde{z}_n + \frac{h}{2}\left[\tilde{u}_n + \tilde{u}^e_{n+1}\right]\right),$$

$$\tilde{u}^f_{n+1/15} = \tilde{u}_n + \frac{8263k_1}{15000} + \frac{124k_2}{75} - \frac{643k_3}{680} - \frac{81k_4}{250} + \frac{2484k_5}{10625},$$

$$k_7 = hF(t_{n+1/15}, \tilde{u}^f_{n+1/15}, \tilde{z}_{n+1/15}),$$

$$k_7 = hF\left(t_{n+1/15}, \tilde{u}^f_{n+1/15}, \tilde{z}_n + \frac{h}{30}\left[\tilde{u}_n + \tilde{u}^f_{n+1/15}\right]\right),$$

$$\tilde{u}^g_{n+1} = \tilde{u}_n + \frac{3501k_1}{1720} - \frac{300}{43}k_2 + \frac{297275k_3}{52632} - \frac{319k_4}{2322} + \frac{24068k_5}{84065} + \frac{3850k_7}{26703},$$

$$k_8 = hF(t_{n+1}, \tilde{u}^g_{n+1}, \tilde{z}_{n+1}),$$

$$k_8 = hF\left(t_{n+1}, \tilde{u}^g_{n+1}, \tilde{z}_n + \frac{h}{2}\left[\tilde{u}_n + \tilde{u}^g_{n+1}\right]\right),$$

(10) $$\tilde{u}_{n+1} = \tilde{u}_n + \frac{13k_1}{160} + \frac{2375}{5984}k_3 + \frac{5k_4}{16} + \frac{12k_5}{85} + \frac{3k_6}{44},$$

and

(11) $$\tilde{u}_{n+1} = \tilde{u}_n + \frac{3k_1}{40} + \frac{875}{2244}k_3 + \frac{23k_4}{72} + \frac{264k_5}{1955} + \frac{125k_7}{11592} + \frac{43k_8}{616}$$

In this example, the trapezoidal rule is used to approximate $\tilde{z}(t_n) \approx \int_{t_0}^{t_n} k(t-s)u(s)\,ds$ on $[t_n, t_{n+1/6}]$, $[t_n, t_{n+4/15}]$, $[t_n, t_{n+2/3}]$, $[t_n, t_{n+5/6}]$, $[t_n, t_{n+1}]$, $[t_n, t_{n+1/15}]$, $[t_n, t_{n+1}]$ in calculating, k_2, k_3, k_4, k_5, k_6, k_7 and k_8 respectively. If desired, the trapezoidal rule may be used on $[t_0, t_n]$ (gives second order accuracy); the trapezoidal rule and Simpson's 1/3 rule (giving third order accuracy, see Table 3) may be used on $[t_0, t_n]$.

In order to get **fifth** order accuracy the integral term must be evaluated more accurately on $[t_n, t_{n+1/6}]$, $[t_n, t_{n+4/15}]$, $[t_n, t_{n+2/3}]$, $[t_n, t_{n+5/6}]$, $[t_n, t_{n+1}]$, $[t_n, t_{n+1/15}]$, $[t_n, t_{n+1}]$ in calculating, k_2, k_3, k_4, k_5, k_6, k_7 and k_8, as shown in (12) --(18) below.

If we interpolating on $\tilde{u}_{n-2}$, $\tilde{u}_{n-1}$, $\tilde{u}_n$, $\tilde{u}_{n+1/6}$ (special formulae required for the first three steps, for example we can use (9)) Lagrange's formula for points t=-2, -1, 0, 1/6 gives

$$u(t) = \frac{1}{h^3}\left(-\frac{3}{13}t\,(t-\frac{h}{6})\,(t+h)u_{-2} + \frac{6}{7}t\,(t-\frac{h}{6})(t+2h)u_{-1} - 3(t+h)(t+2h)(t-\frac{h}{6})u_0 + \frac{216}{91}t(t+h)(t+2h)u_{1/6}\right)$$

If we integrate the expression between 0 and h/6, we get

(12) $$\int_0^{h/6} u(s)ds \approx h\left(\frac{13}{168}u_{1/6} + \frac{1}{5184}u_{-2} - \frac{25}{18144}u_{-1} + \frac{469}{5184}u_0\right).$$

Similarly, we can find t= t=-2, -1, 0, 4/15

(13) $$\int_0^{4h/15} u(s)ds \approx h\left(\frac{34}{285}u_{4/15} + \frac{8}{10125}u_{-2} - \frac{1024}{192375}u_{-1} + \frac{1538}{10125}u_0\right),$$

and find t= t=-2, -1, 0, 2/3

(14) $$\int_0^{2h/3} u(s)ds \approx h\left(\frac{4}{15}u_{2/3} + \frac{1}{81}u_{-2} - \frac{28}{405}u_{-1} + \frac{37}{81}u_0\right),$$

and find t= t=-2, -1, 0, 5/6

(15) $$\int_0^{5h/6} u(s)ds \approx h\left(\frac{85}{264}u_{5/6} + \frac{125}{5184}u_{-2} - \frac{3625}{28512}u_{-1} + \frac{3185}{5184}u_0\right),$$

and find t= t=-2, -1, 0, 1

(16) $$\int_0^{h} u(s)ds \approx h\left(\frac{3}{8}u_1 + \frac{1}{24}u_{-2} - \frac{5}{24}u_{-1} + \frac{19}{24}u_0\right),$$

and find t= t=-2, -1, 0, 1/15

(17) $$\int_0^{h/15} u(s)ds \approx h\left(\frac{31}{960}u_1 + \frac{1}{81000}u_{-2} - \frac{61}{648000}u_{-1} + \frac{2791}{81000}u_0\right),$$

and finally find t= t=-2, -1, 0, 1

(18) $$\int_0^{h} u(s)ds \approx h\left(\frac{3}{8}u_1 + \frac{1}{24}u_{-2} - \frac{5}{24}u_{-1} + \frac{19}{24}u_0\right).$$

Therefore the Runge-Kutta-Verner formulae become $n \geq 4$ (for starting values we can use equation (9))

$$k_1 = hF(t_n, \tilde{u}_n, \tilde{z}(t_n)),$$

$$\tilde{u}^a_{n+1/6} = \tilde{u}_n + \frac{k_1}{6},$$

$$k_2 = hF(t_{n+1/6}, \tilde{u}^a_{n+1/6}, \tilde{z}_{n+1/6}),$$

$$k_2 = hF(t_{n+1/6}, \tilde{u}^a_{n+1/6}, \tilde{z}_n + h(\frac{13}{168}\tilde{u}^a_{n+1/6} + \frac{1}{5184}\tilde{u}_{n-2} - \frac{25}{18144}\tilde{u}_{n-1} + \frac{469}{5184}\tilde{u}_n)),$$

$$\tilde{u}^b_{n+4/15} = \tilde{u}_n + \frac{4k_1}{75} + \frac{16k_2}{75},$$

(19) $$k_3 = hF(t_{n+4/15}, \tilde{u}^b_{n+4/15}, \tilde{z}_{n+4/15}),$$

$$k_3 = hF(t_{n+4/15}, \tilde{u}^b_{n+4/15}, \tilde{z}_n + h(\frac{34}{285}\tilde{u}^b_{n+4/15} + \frac{8}{10125}\tilde{u}_{n-2} - \frac{1024}{192375}\tilde{u}_{n-1} + \frac{1538}{10125}\tilde{u}_n)),$$

$$\tilde{u}^c_{n+2/3} = \tilde{u}_n + \frac{5k_1}{6} - \frac{8k_2}{3} + \frac{5k_3}{2},$$

$$k_4 = hF(t_{n+2/3}, \tilde{u}^c_{n+2/3}, \tilde{z}_{n+2/3}),$$

$$k_4 = hF(t_{n+2/3}, \tilde{u}^c_{n+2/3}, \tilde{z}_n + h(\frac{4}{15}\tilde{u}^c_{n+2/3} + \frac{1}{81}\tilde{u}_{n-2} - \frac{28}{405}\tilde{u}_{n-1} + \frac{37}{81}\tilde{u}_n)),$$

$$\tilde{u}^d_{n+5/6} = \tilde{u}_n + \frac{165k_1}{64} + \frac{55k_2}{6} - \frac{425k_3}{64} + \frac{85k_4}{96},$$

$$k_5 = hF(t_{n+5/6}, \tilde{u}^d_{n+5/6}, \tilde{z}_{n+5/6}),$$

$$k_5 = hF(_{n+5/6}, \tilde{u}^d_{n+5/6}, \tilde{z}_n + h(\frac{85}{264}\tilde{u}^d_{n+5/6} + \frac{125}{5184}\tilde{u}_{n-2} - \frac{3625}{28512}\tilde{u}_{n-1} + \frac{3185}{5184}\tilde{u}_n)),$$

$$\tilde{u}^e_{n+1} = \tilde{u}_n + \frac{12k_1}{15} - 8k_2 + \frac{4015k_3}{612} - \frac{11k_4}{36} + \frac{88k_5}{255},$$

$$k_6 = hF(t_{n+1}, \tilde{u}^e_{n+1}, \tilde{z}_{n+1}),$$

$$k_6 = hF(t_{n+1}, \tilde{u}^e_{n+1}, \tilde{z}_n + h(\frac{3}{8}\tilde{u}^e_{n+1} + \frac{1}{24}\tilde{u}_{n-2} - \frac{5}{24}\tilde{u}_{n-1} + \frac{19}{24}\tilde{u}_n)),$$

$$\tilde{u}^f_{n+1/15} = \tilde{u}_n + \frac{8263k_1}{15000} + \frac{124k_2}{75} - \frac{643k_3}{680} - \frac{81k_4}{250} + \frac{2484k_5}{10625},$$

$$k_7 = hF(t_{n+1/15}, \tilde{u}^f_{n+1/15}, \tilde{z}_{n+1/15}),$$

$$k_7 = hF(t_{n+1/15}, \tilde{u}^f_{n+1/15}, \tilde{z}_n + h(\frac{31}{960}\tilde{u}^f_{n+1/15} + \frac{1}{81000}\tilde{u}_{n-2} - \frac{61}{648000}\tilde{u}_{n-1} + \frac{2791}{81000}\tilde{u}_n)),$$

$$\tilde{u}^g_{n+1} = \tilde{u}_n + \frac{3501k_1}{1720} - \frac{300}{43}k_2 + \frac{297275k_3}{52632} - \frac{319k_4}{2322} + \frac{24068k_5}{84065} + \frac{3850k_7}{26703},$$

$$k_8 = hF(t_{n+1}, \tilde{u}^g_{n+1}, \tilde{z}_{n+1}),$$

$$k_8 = hF(t_{n+1}, \tilde{u}^g_{n+1}, \tilde{z}_n + h(\frac{3}{8}\tilde{u}^g_{n+1} + \frac{1}{24}\tilde{u}_{n-2} - \frac{5}{24}\tilde{u}_{n-1} + \frac{19}{24}\tilde{u}_n)),$$

and the sixth-order method

(20) $$\tilde{u}_{n+1} = \tilde{u}_n + \frac{3k_1}{40} + \frac{875}{2244}k_3 + \frac{23k_4}{72} + \frac{264k_5}{1955} + \frac{125k_7}{11592} + \frac{43k_8}{616}$$ is used to estimate the error in the fifth-order method

(21) $$\tilde{u}_{n+1} = \tilde{u}_n + \frac{13k_1}{160} + \frac{2375}{5984}k_3 + \frac{5k_4}{16} + \frac{12k_5}{85} + \frac{3k_6}{44}.$$

We can construct an algorithm similar to the Runge-Kutta-Fehlberg method and we can repeat Example 1 using this new method (see Table 2).

Table 1 shows the fifth order accuracy obtained with this formula. In Example 1, we have used Runge-Kutta-Verner methods and numerical quadrature, trapezoidal rule, the 3-point rule is known as Simpson's 1/3 rule, the 4-point closed rule is Simpson's 3/8 rule, the 5-point closed rule is Boole's rule (Bode's rule), Weddle's rule, higher rules include the 6-point, 7-point and 8-point and their combinations.

Example 1: Consider a first order Linear Volterra integro-differential equation of the form

(22) $$u'(t) = \mu + \beta u(t) + \lambda \int_0^t e^{-\delta(t-s)} u(s)\, ds, \quad t \ge 0; \qquad u(0) = u_0.$$

If the kernel of convolution type ($K(t,s) = k(t-s)$), and $\int_0^\infty |k(\sigma)|\, d\sigma$, we can solve (22), with suitable initial conditions, by Laplace transforms (see (Cooke, 1966)).

Equation (22) has the analytical solution.

$$u(t) = (e^{-\frac{t}{2}(\delta+\sqrt{(\beta+\delta)^2+4\lambda})}(-2e^{\frac{t}{2}(\delta+\sqrt{(\beta+\delta)^2+4\lambda})}\delta\mu + \sqrt{(\beta+\delta)^2+4\lambda} + e^{\frac{t\beta}{2}}(u_0(\beta\delta+\lambda)(-\beta-\delta+\sqrt{(\beta+\delta)^2+4\lambda}) + (-\delta(\beta+\delta)-2\lambda+\delta\sqrt{(\beta+\delta)^2+4\lambda})\mu) + e^{\frac{t}{2}(\beta+2\sqrt{(\beta+\delta)^2+4\lambda})}(u_0(\beta\delta+\lambda)(\beta+\delta+\sqrt{(\beta+\delta)^2+4\lambda}) + (2\lambda+\delta(\beta+\delta+\sqrt{(\beta+\delta)^2+4\lambda}))\mu))) / (2(\beta\delta+\lambda)+\sqrt{(\beta+\delta)^2+4\lambda}). \tag{23}$$

In solution (23):

Case (i): If we choose $\beta = 0,\ \lambda = -1,\ \delta = 0,\ \mu = 0$ and $u(0) = u_0$, we obtain $u(t) = u_0 \cos(t)$.

Case (ii): If we choose $\beta = -1,\ \lambda = -1,\ \delta = 1,\ \mu = 0$ and $u(0) = u_0$, we obtain $u(t) = u_0\, e^{-t} \cos(t)$.

Case (iii): If we choose $\beta = 0,\ \lambda = -1,\ \delta = 0,\ \mu = 1$ and $u(0) = u_0$, we obtain $u(t) = u_0 \cos(t) + \sin(t)$.

Case (iv): If we choose $\beta = 0,\ \lambda = 1,\ \delta = 0,\ \mu = 1$ and $u(0) = u_0$, we obtain $u(t) = u_0 \cosh(t) + \sinh(t)$.

Case (v): If we choose $\beta = 0,\ \lambda = -1,\ \delta = 1,\ \mu = 0$ and $u(0) = u_0$, we obtain

$$u(t) = u_0 \frac{e^{-\frac{t}{2}}}{3}\left(\cos\left(\frac{\sqrt{3}}{2}t\right) + \sin\left(\frac{\sqrt{3}}{2}t\right)\right).$$

Case (vi): If we choose $\beta = -1/2,\ \lambda = -1,\ \delta = -1,\ \mu = -1/2$ and $u(0) = u_0$, we obtain

$$u(t) = \frac{1}{7}(e^{\frac{t}{4}}(7(u_0-1)\cos\left(\frac{\sqrt{7}}{4}t\right) - \sqrt{7}(3u_0+1)\sin\left(\frac{\sqrt{3}}{2}t\right)) + 7).$$

Case (vii): If we choose $\beta = -1/2,\ \lambda = 1,\ \delta = -1,$ $\mu = -1/2$ and $u(0) = u_0$, we obtain

$$u(t) = \frac{e^{-t}}{15}\left(e^{\frac{5t}{2}}(3u_0-1) - 5e^{t} + 12u_0 + 6\right).$$

The errors found are given Table 1, where ***error=|true value – approximate value|***. Unless otherwise indicated, in this paper, error means absolute error. Table 1 is consistent with the property that the order of the error is $O(h^5)$.

Table 1: Errors in the Solutions (22) for RKV method $\beta = 0$, $\lambda = -1$, $\delta = 0$, $\mu = 1$, $u_0 = 0$, $t_{\max} = 1$ gives $O(h^5)$.

t	*Error1 with h=0.0250*	*Error2 with h=0.0125*	*Error3 with h=0.00625*
0.1	7.7134e-10	2.4116e-11	7.5501e-13
0.2	7.6308e-10	2.3839e-11	7.4746e-13
0.3	7.4661e-10	2.3305e-11	7.3169e-13
0.4	7.2210e-10	2.2518e-11	7.0810e-13
0.5	6.8980e-10	2.1488e-11	6.7685e-13
0.6	6.5006e-10	2.0224e-11	6.3793e-13
0.7	6.0328e-10	1.8740e-11	5.9230e-13
0.8	5.4997e-10	1.7050e-11	5.3990e-13
0.9	4.9069e-10	1.5174e-11	4.8139e-13
1.0	4.2606e-10	1.3129e-11	4.1744e-13

The fifth order Runge-Kutta-Verner method (RKV) and numerical quadrature rules (gives error $O(h^5)$).

Table 2: Errors in the Solutions (22) for RKF Method

A: ($\mu = 0$, $\beta = 0$, $\lambda = -1$, $\delta = 0$, $u_0 = 0$, $t_{\max} = 1$ **gives** $O(h^4)$ **.**

B: ($\mu = 1$, $\beta = 0$, $\lambda = -1$, $\delta = 0$, $u_0 = 0$, $t_{\max} = 1$ **gives** $O(h^5)$ **.**

t	*Error1 with h=0.0250*		*Error2 with h=0.0125*	
	Method A	Method B	Method A	Method B
0.1	4.8696e-08	7.7084e-10	3.0410e-09	2.4070e-11
0.2	4.8142e-08	7.6159e-10	3.0009e-09	2.3731e-11
0.3	4.7108e-08	7.4415e-10	2.9308e-09	2.3137e-11
0.4	4.5603e-08	7.1872e-10	2.8315e-09	2.2295e-11
0.5	4.3643e-08	6.8557e-10	2.7039e-09	2.1213e-11
0.6	4.1247e-08	6.4505e-10	2.5492e-09	1.9903e-11
0.7	3.8439e-08	5.9760e-10	2.3691e-09	1.8379e-11
0.8	3.5248e-08	5.4373e-10	2.1653e-09	1.6657e-11
0.9	3.1705e-08	4.8402e-10	1.9400e-09	1.4756e-11
1.0	2.7845e-08	4.1910e-10	1.6952e-09	1.2697e-11

The fifth order (RKF) and numerical quadrature rules (gives error $O(h^4)$ and $O(h^5)$).

5. Conclusion

The results are shown in Table 1. The apparent order of convergence is five, which is not surprising because of the use equation (22). After above calculation we are expecting order of $O(h^5)$. In view, it seems to be true because of the truncation error for Runge-Kutta-Verner and Boole's rule are $O(h^5)$. Numerical order of convergence is also calculated:

Ord= (ln($Error_1$) - ln($Error_2$)) / ln(2).

We expected that Ord=5. Obtained theoretical results are confirmed by numerical experiment. The seventh-order Runge-Kutta and eighth-order Runge-Kutta methods can also be adapted to the numerical solution of equation of equation (22).

Corresponding Author:
Dr. Ali FILIZ
Department of Mathematics, Adnan Menderes University, 09010 AYDIN-Turkey
E-mail: afiliz@adu.edu.tr

References

1. Abramowitz, M. and Stegun, I. A. Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, ninth dover printing, tenth gpo printing edn, Dover, New York, 1972.
2. Asanov, A. Uniqueness of the solution of systems of convolution-type Volterra integral equations of the first kind, In: Inverse problems for differential equations of mathematical physics (Russian), Novosibirsk: Akad. Nauk SSSR sibirsk. Otdel. Vychil. Tsentr., 1978:Vol. 155, pp. 2–34.
3. Baker, C. T. H. Numerical Treatment of Integral Equations, Clarendon Press, Oxford, New York, 1978.
4. Baker, C. T. H., Bochorov, G. A., Filiz, A., Ford, N. J., Paul, C. A. H., Rihan, F. A., Tang, A., Thomas, R. M., Tian, H. and Wille, D. R. Numerical modeling by retarded functional differential equations, Technical Report 335, Manchester Centre for Computational Mathematics, Numerical analysis report, ISSN 1360-1725, 1998.
5. Baker, C. T. H., Bochorov, G. A., Filiz, A., Ford, N. J., Paul, C. A. H., Rihan, F. A., Tang, A., Thomas, R. M., Tian, H. and Wille, D. R. Numerical modeling by delay and Volterra functional differential equations, In: Computer Mathematics and its Applications- Advances and Developments (1994-2005), Editor: Elias A. Lipitakis, LEA Publishers, Athens, Greece, 2006:pp. 233–256.
6. Bellman, R. A survey of the theory of the boundedness stability and asymptotic behaviour of solutions of linear and nonlinear differential and difference equations, Washington, D. C., 1949.
7. Burden, R. L. and Faires, J. D. Numerical Analysis, Brooks/Cole publishing Company, USA, New York, 1997.
8. Cooke, K. L. Functional differential equations close to differential equation, Amer. Math. Soc. 1966:72: 285–288.
9. Filiz, A. Numerical Solution of Some Volterra Equations, PhD thesis, The Manchester University, Manchester, UK, 2000a.
10. Filiz, A. On the solution of Volterra and Lotka-Volterra type equations, LMS supported One Day Meeting in Delayed Differential Equation. Liverpool, UK, 2000b.

11. Filiz, A. (2013). A fourth-order robust numerical method for integro-differential equations, Asian Journal of Fuzzy and Applied Mathematics 2013:1(1): 28–33.
12. Linz, P. Analytical and Numerical Methods for Volterra Equations, SIAM, Philadelphia, 1985.
13. Ueberhuber, C. W. Numerical Computation 2: Methods, Software, and Analysis, Springer-Verlag, Berlin, 1972.
14. Volterra, V. Lec¸ons sur la Th´ eorie Math ´ ematique de la Lutte Pour la Vie, Gauthier-Villars, Paris, 1931.
15. Volterra, V. Sulle equazioni integro-differenziali della teoria dell'elastica, Atti della Reale Accademia dei Lincei 18 (1909), Reprinted in Vito Volterra, Opera Mathematiche; Memorie e Note Vol. 3., Accademia dei Lincei, Rome, 1957.
16. Volterra, V. Theory of Functionals and of Integro Differential Equations, Dover, New York, 1959.
17. Wolfram. Newton-cotes Formulas, http://mathworld.wolfram.com/ newtoncotesformulas.html, 2013.

6. Appendix

Various numerical solution of equation (22).

Table 3: Errors in the solution of (22) with;

Errors in the Solutions (22) for Various Methods $\beta = 0,\ \lambda = -1,\ \delta = 0,\ \mu = 1,\ u_0 = 0,\ t_{\max} = 1$									
	Exact	(A) Explicit		(B) Implicit		(C) RK4 and SimpTrap		(D) RK4 and Simp II	
t	Solution	h=0.025	h=0.0125	h = 0.025	h = 0.0125	h=0.025	h=0.0125	h = 0.025	h = 0.0125
0.1	0.0998334	5.7228e-05	2.9907e-05	6.7538e-05	3.2485e-05	3.9424e-09	5.5109e-10	7.3484e-10	3.3681e-11
0.2	0.1986693	2.3829e-04	1.2166e-04	2.5828e-04	1.2666e-04	1.7562e-08	2.3103e-09	1.0650e-09	5.4422e-11
0.3	0.2955202	5.3976e-04	2.7348e-04	5.6822e-04	2.8059e-04	4.0594e-08	5.2430e-09	1.3730e-09	7.3910e-11
0.4	0.3894183	9.5579e-04	4.8236e-04	9.9091e-04	4.9114e-04	7.2586e-08	9.2908e-09	1.6503e-09	9.1611e-11
0.5	0.4794255	1.4782e-03	7.4414e-04	1.5176e-03	7.5399e-04	1.1290e-07	1.4373e-08	1.8888e-09	1.0702e-10
0.6	0.5646425	2.0965e-03	1.0535e-03	2.1375e-03	1.0638e-03	1.6073e-07	2.0385e-08	2.0813e-09	1.1967e-10
0.7	0.6442177	2.7983e-03	1.4043e-03	2.8375e-03	1.4141e-03	2.1509e-07	2.7207e-08	2.2209e-09	1.2913e-10
0.8	0.7173561	3.5692e-03	3.6030e-03	4.0738e-06	1.7975e-03	2.7488e-07	3.4696e-08	2.3018e-09	1.3503e-10
0.9	0.7833269	4.3930e-03	2.1999e-03	4.4178e-03	2.2061e-03	3.3884e-07	4.2695e-08	2.3191e-09	1.3705e-10
1.0	0.8414710	5.2524e-03	2.6280e-03	5.2641e-03	2.6309e-03	4.0562e-07	5.1034e-08	2.2691e-09	1.3496e-10
	Exact	(E) RK2 and Trap		(F) RK4 and Trap		(G) RK4 and TrapSimp		(H) RK4 and Simp I	
t	Solution	h=0.025	h=0.0125	h = 0.025	h = 0.0125	h=0.025	h=0.0125	h = 0.025	h = 0.0125
0.1	0.0998334	5.1818e-06	1.2955e-06	7.1956e-09	1.9741e-09	2.3161e-09	2.4629e-10	7.3484e-10	3.3686e-11
0.2	0.1986693	1.0208e-05	2.5522e-06	6.2993e-08	1.6502e-08	7.8435e-09	8.9431e-10	1.0657e-09	5.4461e-11
0.3	0.2955202	1.4926e-05	3.7317e-06	2.1827e-07	5.6291e-08	1.6477e-08	1.9315e-09	1.3753e-09	7.4015e-11
0.4	0.3894183	1.9187e-05	4.7971e-06	5.2191e-07	1.3354e-07	2.8050e-08	3.3373e-09	1.6552e-09	9.1811e-11
0.5	0.4794255	2.2852e-05	5.7133e-06	1.0198e-06	2.5971e-07	4.2334e-08	5.0837e-09	1.8973e-09	1.0734e-10
0.6	0.5646425	2.5790e-05	6.4478e-06	1.7539e-06	4.4524e-07	5.9044e-08	7.1356e-09	2.0941e-09	1.2014e-10
0.7	0.6442177	2.7883e-05	6.9711e-06	2.7614e-06	6.9939e-07	7.7842e-08	9.4512e-09	2.2388e-09	1.2976e-10
0.8	0.7173561	2.9027e-05	7.2572e-06	4.0738e-06	1.0300e-06	9.8344e-08	1.1983e-08	2.3255e-09	1.3585e-10
0.9	0.7833269	2.9136e-05	7.2844e-06	5.7162e-06	1.4432e-06	1.2012e-07	1.4677e-08	2.3491e-09	1.3808e-10
1.0	0.8414710	2.8139e-05	7.0351e-06	7.7067e-06	1.9437e-06	1.4271e-07	1.7477e-08	2.3057e-09	1.3619e-10

(A) The explicit ($\theta = 0$) method and the trapezoidal rule (gives error O(h));

(B) The implicit ($\theta = 1$) method and the trapezoidal rule (gives error O(h));

(C) The RK4 and Simpson's 1/3 rule and the trapezoidal rule (gives error $O(h^3)$);

(D) The fourth order Runge-Kutta methodu(RK4) and Simpson's method II (gives error $O(h^4)$);

(E) The second order Runge-Kutta methodu(RK2) and the trapezoidal rule (gives error $O(h^2)$);

(F) The fourth order Runge-Kutta method (RK4) and the trapezoidal rule (gives error $O(h^2)$);

(G) The RK4 and the trapezoidal rule and Simpson's 1/3 rule (gives error $O(h^3)$);

(H) The fourth order Runge-Kutta method (RK4) and Simpson's method I (gives error $O(h^4)$

9/11/2013