VOL. 30 NO. 9
ISSN: 2878-1894
SEPTEMBER, 2023

African Scholar Publications & Research International

African Scholars Journal of Science Innovation & Tech. Research (JSITR-9)

Linear Volterra Integro-Differential Equations Numerical Solution Using Finite Difference Methods

Albert Ayuba Shalangwa[1]; Ajiya Yahaya[2]; & Aminu Audu[2] and David John[3]

[(1)]Mathematics Unit School of Basic and Remedial Studies (SBRS), Gombe State University, Gombe, Nigeria.[(2)]Department Mathematics, Gombe State University, Gombe, Nigeria. [(3)]Mathematics Unit School of Basic and Remedial Studies (SBRS), Gombe State University, Gombe, Nigeria.

Abstract

This research focuses on using finite difference techniques to numerically solve linear Volterra integro differential equations. In order to solve these classes of equations, we contrasted the Simpson's-Trapezoid approach to the fourth-order Runge-Kutta method. Simpson's rule and the trapezoidal rule were used in the Simpson's-Trapezoid method to approximate the integral in the linear Volterra integro-differential equations. Finite differentiation was then used to determine the derivatives, which were then converted into systems of equations. The unknown functions in the matrix are thus determined using the Gaussian elimination approach, leading to a "n by n" matrix equation. Higher order linear Volterra integro-differential equations are reduced to first order or systems of first order linear Volterra integro differential equations in the fourth-order Runge-Kutta method, and the unknown functions, such as yi(x), i = 1(1)n, in the system of first order linear integro-differential equations, are obtained. Examples are provided to help illustrate how effective, straightforward, and accurate the methods are. Fourth order Runge-Kutta results are compared to those from the finite difference method.

Keywords: Integro-differential equation, Volterra integro-differential equation, Finite differentiation, Simpson's integration rule and Trapezoid integration rule.

Introduction

Engineering, mechanics, physics, chemistry, astronomy, biology, economics, potential theory, and electrostatics are just a few of the many practical fields where the integral-differential equation (IDE) is used often (Ali et al., 2013). The unknown function appears in an equation under the sign of integration in an integro-differential equation, which also contains the unknown function's derivatives. It falls within the categories of Volterra equations and Fredholm equations. For the integral part of the Volterra type, the region's upper bound is a variable while for the Fredholm type, it is a fixed value.

Functional equations, such as partial differential equations, integral and integro-differential equations, stochastic equations, and others are frequently produced when real-world issues are mathematically modelled. Since it is frequently challenging to solve integral-differential equations analytically, it is necessary to find an accurate approximation. There are several numerical approaches that can be used to approximate the Fredholm or Volterra integro-diffrential equations. Although integral-differential equations are significant, even numerically solving them is difficult, making the process time-consuming. Every phenomenon in real life has numerous elements and variables connected to one another that are governed by the law pertaining to that phenomenon. When the relationship between the parameters and the variables is expressed mathematically, we typically derive a mathematical model of the problem. This model may take the form of an equation, a diferential equation, an integral equation, an integro-differential equation, a system of integro-differential equations, etc. (Behrouz Raftari, 2006).

An integro-differential equation is a type of functional equation where the unknown function is represented by both its derivative and the integral sign (Miranda et al., 2002; Jain et al., 2014).

We provide a finite difference method to solve linear Volterra integro-differential equations of the following type in this paper:

$$a_0 y^n(x) + a_1 y^{n-1}(x) + \cdots + a_n y(x) = f(x) + \lambda \int_a^x \mathrm{k(x,t)y(t)dt} \qquad (1)$$

With initial conditions

$$y^j(a) = A_j; \quad j = 0,1,2,3, \dots, (r-1)$$

Where $a_{0,}a_1.a_{2,} \dots, a_n$ are constants, $y_n(x)$ indicates the nth order derivative of $y(x)$ and $y(t)$ is a linear function. In addition the kernel $k(x,t)$ are assumed real, differential for $x \in [a,b]$ and A_j ; $0 \leq j \leq (r-1)$ are finite constant. The functions $f(x)$, the kernel $k(x,t)$ are known and $y(x)$ is the solution to be determined.

SIMPSON'S-TRAPEZOID METHOD

The differential problem is directly translated into a discrete domain using the finite difference method, allowing for a numerical solution. It accurately depicts the governing equation $\frac{\partial f}{\partial x} = \frac{(f_{i+1} - f_i)}{(x_{i+1} - x_i)}$. The governing equation is defined throughout the interval using discontinuous but connected areas. The Taylors-Series expansion can be used to obtain the differential equation, which is the fundamental finite difference form, in addition to direct interpretation.

Consider the general finite difference approximation to the dth order derivative:

$$F^d(x) = \frac{d!}{h^d}\sum_{n=0}^{d+p-1}\left(\sum_{i=imin}^{imax} i^n\, C_i\right)\frac{h^n}{n!}F_n(x) + O(h^p) \qquad (2)$$

In order for equation (2) to be satisfied, it is necessary that

$$\sum_{i=imin}^{imax} i^n\, C_i = \begin{cases} 1, & if\ 0 \le n \le d+p-1\ and\ d \ne p \\ 0, & if\ n = d \end{cases} \qquad (3)$$

This is a set of $d+p$ linear equations in $i_{max} - i_{min} + 1$ unknowns. If we constrain the number of unknowns to be $d+p$, the linear system has a unique solution. A forward difference approximation occurs if we set $i_{min} = 0$ and $i_{max} = d+p-1$. A backward difference approximation occurs if we set $i_{max} = 0$ and $i_{min} = -(d+p-1)$. A centered difference approximation occurs if we set $i_{max} = -i_{min} = [(d+p-1)/2]$ where it appears that $d+p$ is necessarily an odd number. As it turns out, p can be chosen to be even regardless of the parity of d and $i_{max} = [(d+p-1)/2]$.

We take the Volterra Integro-Differential Equation in (1) into consideration and use numerical integration and numerical differentiation to estimate the solution. We divide the integration time interval (a,x) into N=2M equal width subintervals.

$$h = \frac{X_N - a}{N}; \quad N \ge 1 \qquad (4)$$

Where x_n is the end point we choose for x.We shall set $t_0 = a$ and $t_j = t_0 + jh$

Since we will be using either t or x as the independent variable for the solution y. We will call $x_0 = t_0 = a,\ x = x_N = t_N$ and $x_i = a + ih = t_i$. We will refer to the value of the functions $f(x)$ and $a(x)$ at x_i as $f(x_i) = f_i$ and $a(x_i) = a_i$, the value of the kernel $k(x,t)$ at (x_i, t_j) as $k(x_i, t_j) = k_{ij}$ and the approximate value of the solution $y(x)$ at x_i or t_i as $y(t_i) = y(x_i) = y_i$ and $y^n(x_i) = y_i^n$. $k(x_i, t_j)$ clearly varnishes for $t_j > x_i$ as the integration ends at $t_j \le x_i$. Note that the particular value $y(x_0) = y_0$ according to (1). Therefore, if we estimate the integral in the Volterra integro-differential equation (1) using the Trapezoidal rule and Simpson rule with n subinterval, we have;

$$\int_a^{x_1} k(x,t)y(t)dt = h\left[\frac{1}{2}k(x_1,t_0)y(t_0) + \frac{1}{2}k(x_1,t_1)y(t_1)\right] \qquad T\ Rule$$

$$\int_a^{x_2} k(x,t)y(t)dt = \frac{h}{3}[k(x_2,t_0)y(t_0) + 4k(x_2,t_1)y(t_1) + k(x_2,t_2)y(t_2)]\ S\ Rule$$

$$\begin{aligned}\int_a^{x_3} k(x,t)y(t)dt \\ &= h\left[\frac{1}{2}k(x_3,t_0)y(t_0) + k(x_3,t_1)y(t_1) + k(x_3,t_2)y(t_2)\right. \\ &\left. + \frac{1}{2}k(x_3,t_3)y(t_3)\right]\ T\ Rule\end{aligned}$$

$$\int_a^{x_4} k(x,t)y(t)dt$$
$$= \frac{h}{3}[k(x_4,t_0)y(t_0) + 4k(x_4,t_1)y(t_1) + 2k(x_4,t_2)y(t_2) + 4k(x_4,t_3)y(t_3) + k(x_4,t_4)y(t_4)]\ S\ Rule$$

.

.

.

$$\int_a^{x_{N-1}} k(x,t)y(t)dt$$
$$= h\left[\frac{1}{2}k(x_{N-1},t_0)y(t_0) + k(x_{N-1},t_1)y(t_1) + k(x_{N-1},t_2)y(t_2) + \cdots + k(x_{N-1},t_{N-2})y(t_{N-2}) + \frac{1}{2}k(x_{N-1},t_{N-1})y(t_{N-1})\right]\ T\ Rule$$

$$\int_a^{x_N} k(x,t)y(t)dt$$
$$= \frac{h}{3}[k(x_N,t_0)y(t_0) + 4k(x_N,t_1)y(t_1) + 2k(x_N,t_2)y(t_2) + \cdots + 4k(x_N,t_{N-1})y(t_{N-1}) + k(x_N,t_N)y(t_N)]\ S\ Rule \quad (5)$$

Furthermore, by (compact form) the integro-differential equation (1) is approximated as,

$$a_{0,1}y_1^n + a_{1,1}y_1^{n-1} + \cdots + a_{N,1}y_1 = f_1 + \lambda h[\frac{1}{2}k_{1,0}y_0 + \frac{1}{2}k_{1,1}y_1]$$
$$a_{0,2}y_2^n + a_{1,2}y_2^{n-1} + \cdots + a_{N,2}y_2 = f_2 + \lambda\frac{h}{3}[k_{2,0}y_0 + 4k_{2,1}y_1 + k_{2,2}y_2]$$
$$a_{0,3}y_3^n + a_{1,3}y_3^{n-1} + \cdots + a_{N,3}y_3 = f_3 + \lambda h[\frac{1}{2}k_{3,0}y_0 + k_{3,1}y_1 + k_{3,2}y_2 + \frac{1}{2}k_{3,3}y_3]$$
$$a_{0,4}y_4^n + a_{1,4}y_4^{n-1} + \cdots + a_{N,4}y_4 = f_4 + \lambda\frac{h}{3}[k_{4,0}y_0 + 4k_{4,1}y_1 + 2k_{4,2}y_2 + 4k_{4,3}y_3 + k_{4,4}y_4]$$

.

.

.

$$a_{0,N-1}y_{N-1}^n + a_{1,N-1}y_{N-1}^{n-1} + \cdots + a_{N,N-1}y_{N-1} = f_{N-1} + \lambda h[\frac{1}{2}k_{N-1,0}y_0 + k_{N-1,1}y_1 + k_{N-1,2}y_2 + \cdots + \frac{1}{2}k_{N-1,N-1}y_{N-1}]$$
$$a_{0,N}y_{N-1}^n + a_{1,N}y_{N-1}^{n-1} + \cdots + a_{N,N}y_N = f_N + \lambda\frac{h}{3}[k_{N,0}y_0 + 4k_{N,1}y_1 + 2k_{N,2}y_2 + \cdots + 4k_{N,N-1}y_{N-1} + k_{N,N}y_N] \quad (6)$$

taking advantage of finite differentiation we have,

$N = 1$:

$$a_{0,1}\left(\frac{n!}{h^n}\sum_{i=imin}^{imax} C_i\, y(x+ih)\right)+a_{1,1}\left(\frac{(n-1)!}{h^{n-1}}\sum_{i=imin}^{imax} C_i\, y(x+ih)\right)+\cdots+a_{n,1}y_1$$
$$=f_1+\lambda h[\frac{1}{2}k_{1,0}y_0+\frac{1}{2}k_{1,1}y_1]$$

$N\ =2:$

$$a_{0,2}\left(\frac{n!}{h^n}\sum_{i=imin}^{imax} C_i\, y(x+ih)\right)+a_{1,2}\left(\frac{(n-1)!}{h^{n-1}}\sum_{i=imin}^{imax} C_i\, y(x+ih)\right)+\cdots+a_{n,2}y_2$$
$$=f_2+\lambda\frac{h}{3}[k_{2,0}y_0+4k_{2,1}y_1+k_{2,2}y_2]$$

$N\ =3:$

$$a_{0,3}\left(\frac{n!}{h^n}\sum_{i=imin}^{imax} C_i\, y(x+ih)\right)+a_{1,3}\left(\frac{(n-1)!}{h^{n-1}}\sum_{i=imin}^{imax} C_i\, y(x+ih)\right)+\cdots+a_{n,3}y_3$$
$$=f_3+\lambda h[\frac{1}{2}k_{3,0}y_0+k_{3,1}y_1+k_{3,2}y_2+\frac{1}{2}k_{3,3}y_3]$$

$N\ =4:$

$$a_{0,4}\left(\frac{n!}{h^n}\sum_{i=imin}^{imax} C_i\, y(x+ih)\right)+a_{1,4}\left(\frac{(n-1)!}{h^{n-1}}\sum_{i=imin}^{imax} C_i\, y(x+ih)\right)+\cdots+a_{n,4}y_4$$
$$=f_4+\lambda\frac{h}{3}[k_{4,0}y_0+4k_{4,1}y_1+2k_{4,2}y_2+4k_{4,3}y_3+k_{4,4}y_4]$$

.

.

.

$N\ =N-1:$

$$a_{0,N-1}\left(\frac{n!}{h^n}\sum_{i=imin}^{imax} C_i\, y(x+ih)\right)+a_{1,N-1}\left(\frac{(n-1)!}{h^{n-1}}\sum_{i=imin}^{imax} C_i\, y(x+ih)\right)+\cdots$$
$$+a_{n,N-1}y_{N-1}$$
$$=f_{N-1}+\lambda h[\frac{1}{2}k_{N-1,0}y_0+k_{N-1,1}y_1+k_{N-1,2}y_2+\cdots+\frac{1}{2}k_{N-1,N-1}y_{N-1}]$$

$N\ =N:$

$$a_{0,N}\left(\frac{n!}{h^n}\sum_{i=imin}^{imax} C_i\, y(x+ih)\right)+a_{1,N}\left(\frac{(n-1)!}{h^{n-1}}\sum_{i=imin}^{imax} C_i\, y(x+ih)\right)+\cdots+a_{n,N}y_N$$

$$= f_N + \lambda\frac{h}{3}\left[k_{N,0}y_0+4k_{N,1}y_1+2k_{N,2}y_2+\cdots+4k_{N,N-1}y_{N-1} + k_{N,N}y_N\right] \qquad (7)$$

The matrix type KY = F can be used to represent the system of N equations mentioned above, where K is the matrix of the coefficient of the vector $Y = (y_1, y_2, y_3, y_4, \dots, y_{n-1}, y_n)^T$ and F is a column matrix containing $f(x)$ and its coefficient, and the remaining terms of the system of equation above.

FOUTH ORDER RUNGE-KUTTA METHOD

In this part, we examine the linear Volterra integro-differential equation (1). With the Runge-Kutta method, additional precision is supposedly achieved by only requiring the function values at a certain location on the subinterval. The Runge-Kutta method is widely used to resolve initial value problems. The Runge-Kutta method for solving first order differential equations affords a higher degree of accuracy, is yet another step-by-step procedure where a table of function values for a large values x is accumulated, and also has the advantage of self starting properties. These methods offer close approximations that converge to the actual solution as h approaches 0. It requires substantially more processing every step, and each level also calls for a number of intermediate calculations. The Runge-Kutta method of fourth order is given by

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \qquad (8)$$

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf(x_n + \frac{h}{2}, y_n + \frac{k_1}{2})$$

$$k_3 = hf(x_n + \frac{h}{2}, y_n + \frac{k_2}{2})$$

$$k_4 = hf(x_n + h, y_n + k_3)$$

Equation (8)'s vector format is given by

$$Y_{n+1} = Y_n + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \qquad (9)$$

Where

$$K_1 = (k_{11}, k_{21}, \dots, k_{n1})^T$$

$$K_2 = (k_{12}, k_{22}, \dots, k_{n2})^T$$

$$K_3 = (k_{13}, k_{23}, \dots, k_{n3})^T$$

$$K_1 = (k_{14}, k_{24}, \dots, k_{n4})^T$$

And

$$
\begin{aligned}
K_{i1} &= hf_i(x_n, y_{1,n}, y_{2,n}, \dots, y_{m,n}) \\
K_{i2} &= hf_i(x_n + \frac{h}{2}, y_{1,n} + \frac{k_{11}}{2}, y_{2,n} + \frac{k_{21}}{2}, \dots, y_{m,n} + \frac{k_{n1}}{2}) \\
K_{i3} &= hf_i(x_n + \frac{h}{2}, y_{1,n} + \frac{k_{12}}{2}, y_{2,n} + \frac{k_{22}}{2}, \dots, y_{m,n} + \frac{k_{n2}}{2}) \\
K_{i4} &= hf_i(x_n + h, y_{1,n} + k_{13}, y_{2,n} + k_{23}, \dots, y_{m,n} + k_{n3})
\end{aligned}
$$

$$i = 1(1)j \qquad (10)$$

NUMERICAL EXAMPLES

Now we solve the following examples using the methods described above:

Example:1

Consider the Volterra integro-differential equation

$$y^1(x) + y(x) = (x^2 + 2x + 1)e^{-x} + 5x^2 + 8 - \int_0^x ty(t)dt\,;\ \ 0 \le x \le 1$$

$$y(0) = 10 \qquad (11)$$

which has the exact solution $y(x) = 10 - xe^{-x}$.

$$h = \frac{X_n - a}{N} = \frac{1-0}{14} = 0.0714$$

Solving equation (11) for $n = 0(1)13$ using the methods presented above we have the results in table 1 below:

Comparison of Errors Between Simpson's-Trapezoid Method and Fourth Order Runge-Kutta Method

Note that:

STM = Simpson’s-Trapezoid Method

FRKM = Fourth Order Runge- Kutta Method

ESTM = Absolute Error of the Simpson’s-Trapezoid Method

EFRKM = Absolute Error for the Fourth Order Runge-Kutta Method

Here, we have defined absolute error as:

$Absolute\ error\ = |\ Exact\ solution - Approximate\ solution|$

Table 1: TABLE OF RESULTS FOR EXAMPLE 1

x_i	EXACT	STM	FRKM	ESTM	EFRKM
0.0714	9.933520218	9.9337560613	9.933521225	$2.358433E^{-04}$	$1.00700E^{-06}$
0.1428	9.876202762	9.876507767	9.876216954	$3.050050E^{-04}$	$1.419200E^{-05}$
0.2142	9.827100559	9.827593162	9.827165101	$4.9260300E^{-04}$	$6.4542000E^{-05}$
0.2857	9.785299870	9.785878737	9.785536706	$5.788670E^{-04}$	$2.368360E^{-04}$
0.3571	9.750136229	9.750891348	9.750581736	$7.551190E^{-04}$	$4.455070E^{-04}$
0.4286	9.720801197	9.721589067	9.721617291	$7.878700E^{-04}$	$8.160940E^{-04}$
0.5000	9.696734670	9.697710641	9.698017844	$9.759710E^{-04}$	$1.283174E^{-03}$
0.5714	9.677310846	9.678248547	9.679207266	$9.377010E^{-04}$	$1.896420E^{-03}$

0.6428	9.662004144	9.663088160	9.664652376	$1.084016E^{-03}$	$2.648232E^{-03}$
0.7143	9.650325388	9.651366596	9.653857850	$1.041208E^{-03}$	$3.532462E^{-03}$
0.7857	9.641877524	9.643124293	9.646362319	$1.246769E^{-03}$	$4.484795E^{-03}$
0.8571	9.636254445	9.637409977	9.641735491	$1.155532E^{-03}$	$5.481046E^{-03}$
0.9286	9.633104130	9.634436746	9.639576189	$1.332616E^{-03}$	$6.472059E^{-03}$

Example: 2

Consider the second order integro-differential equation with constant coefficients

$$y^{11}(x) = e^{-x} - x + \int_0^x xty(t)dt;\ \ 0 \leq x \leq 0.1$$

$$y(0) = 1; y^1(0) = 1 \qquad (12)$$

And the exact solution is given by $y(x) = e^x$

$$h = \frac{X_n - a}{N} = \frac{0.1 - 0}{10} = 0.01$$

Solving equation (12) for $i = 1(1)2$ and $n = 0(1)9$ using the methods presented above we have the results in table 2 below:

Table 2: TABLE OF RESULTS FOR EXAMPLE 2

x_i	EXACT	STM	FRKM	ESTM	EFRKM
0.01	1.010050167	1.010050025	1.010050000	$1.644999998E^{-07}$	$1.6700E^{-07}$
0.02	1.020201340	1.020200025	1.020200006	$1.3147979980E^{-06}$	$1.3340E^{-06}$
0.03	1.030454534	1.030450094	1.030450034	$4.440129374E^{-06}$	$4.5000E^{-06}$
0.04	1.040810774	1.040934919	1.040800110	$1.241448072E^{-04}$	$1.0664E^{-05}$
0.05	1.051271096	1.051519876	1.051250271	$2.487796336E^{-04}$	$2.0825E^{-05}$
0.06	1.061836547	1.062205021	1.061800567	$3.684741463E^{-04}$	$3.5980E^{-05}$
0.07	1.072508181	1.072990430	1.072451059	$4.822490318E^{-04}$	$5.7122E^{-05}$
0.08	1.083287068	1.083876186	1.083201815	$5.891182191E^{-04}$	$8.5253E^{-05}$
0.09	1.094174284	1.094862386	1.094052921	$6.881015300E^{-04}$	$1.2136E^{-04}$
0.10	1.105170918	1.105949138	1.105004477	$7.782201046E^{-04}$	$1.6644E^{-04}$

Example: 3

Consider the third order integro-differential equation

$$y^{111}(x) = \sin(x) - x - xy(x) + \int_0^x y(t)dt;\ \ 0 \leq x \leq 0.1$$

$$with\ y(0) = 1; y^1(0) = 0; y^{11}(0) = -1 \qquad (13)$$

and the exact solution is given by $y(x) = cos(x)$

$$h = \frac{X_n - a}{N} = \frac{0.1 - 0}{10} = 0.01$$

Solving equation (13) for $i = 1(1)3$ and $n = 0(1)9$ using the methods presented above we have the results in table 3 below:

Table 3: TABLE OF RESULTS FOR EXAMPLE 3

x_i	EXACT	STM	FRKM	ESTM	EFRKM
0.01	0.9999500004	0.9999499911	0.9999498344	$1.2827665596E^{-09}$	$1.6600E^{-07}$
0.02	0.9998000067	0.9998000000	0.9998000000	$6.6994995107E^{-09}$	$1.3170E^{-06}$
0.03	0.9995500337	0.9995500013	0.9995500013	$3.2446999754E^{-08}$	$4.4270E^{-06}$
0.04	0.9992001067	0.9992000043	0.9992000043	$1.0241096771E^{-07}$	$4.4270E^{-06}$
0.05	0.9987502604	0.9987500080	0.9987500080	$2.5235124845E^{-07}$	$2.0302E^{-05}$
0.06	0.9987502604	0.9987500080	0.9987500080	$2.5235124845E^{-07}$	$2.0302E^{-05}$
0.07	0.9975510003	0.9975500203	0.9975500203	$9.7995452308E^{-07}$	$5.5144E^{-05}$
0.08	0.9968017063	0.9968000290	0.9968000290	$1.6773043225E^{-06}$	$7.3299E^{-04}$
0.09	0.9959527330	0.9959500393	0.9959500393	$2.6937224058E^{-06}$	$1.1598E^{-04}$
0.10	0.9950041653	0.9950000052	0.9950000052	$4.1135613364E^{-06}$	$1.5825E^{-04}$

In this study, we found that as the order of the differential equation increases the Simpson's-Trapezoid approach outperformed the fourth order Runge-Kutta method in providing better results for the linear Volterra integro-differential problem. These results are shown in the tables 1, 2, and 3 above.

CONCLUSION

Analytical solutions to integral-differential equations are typically challenging. It is frequently necessary to find approximations of the solutions. In this study, a comparison between the Simpson's-Trapezoid Method and the conventional Runge-Kutta Method was done in order to solve linear Volterra integro-differential equations. A number of numerical examples are provided to show the reliability and viability of these methods.

Applying Simpson's-Trapezoid Method to problems in integral, integro-differential, and fractional calculus will be a major advancement in applied mathematics as it yields a very efficient result, but at the cost of significant computational work. Simpson's-Trapezoid Method poses to be a very simple and accurate numerical method for the numerical solution of Volterra integro-differential equations.

The Simpson's-Trapezoid Method has a notable limitation that requires nonlinear issues to be converted into corresponding linear problems before they can be solved. In order to solve nonlinear integro-differential equations and other comparable equations in science and engineering, a combination of the Simpson's-Trapezoid approach and the Newton-Raphson-Kantorovich linearization approach can be helpful.

REFERENCES

Ali F., Ali I., & Mehmet E. (2013). A fifth-order Numerical Convergence For Linear Volterra Integro-Differential Equation. *Life Science Journal.*10 (4)

Behrouz R. (2006). Numerical Solutions of The Linear Volterra Integro-Differential Equations:Homotopy Perturbation Method and Finite Difference Method. *World Applied Sciences Journal*, 9.

Jaradat H.M., Fadi A. & Alsayyed O. (2009). Series Solution to the High-Order Integro-Differential Equations. *Fasc Matematica*, *Tom XVI*, 247-257. 18

Hemeda A. A. (2012). New Iterative Method:Application to nth-order Integro-Differential Equations. *International Mathematical forum,Vol.7*(47,2317-2332).

Karem A. K. (2007). Lecture Notes on Numerical Differentiation.
Miranda, Mario J., & Paul L. F. (2002). Applied Computational Economics and Finance. *Cambridge, MA: MIT Press*.
Jain M. K., Iyengar S. R. K. & Jain R. K. (2014). Numerical Methods for Scientific Engineering Computation 6th ed. *New Age International Publishers* p403-540.
Nocedal J., & Stephen J. W. (1999). Numerical Optimization. *Springer Vergag New York, Inc*.
Zhao, J. & Corless R.M. (2006). Compact finite difference method has been used for integro-differential equations. *Applied Mathematics Computing*.,177: 271-288.