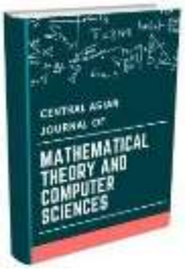

CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES

https://cajmtcs.casjournal.org/index.php/CAJMTCS

Volume: 07 Issue: 02 | April 2026 ISSN: 2660-5309

Article

Newton–Kantorovich Based Numerical Methods for Nonlinear Integral Equations

Shaymaa A. Hameed[1], Abdullah Mhmood Jasim[2], Nabila I. Aziz[3]

1. Department of Mathematics, College of Basic Education, Diyala University, Diyala, Iraq
   E-mail: basicmathte3@uodiyala.edu.iq

2. Department of Mathematics, College of Education – Tuzkhurmatu, University of Tikrit, Salahaddin, Iraq
   E-mail: Abdullah.jasem122@tu.edu.iq

3. Department of Mathematics, College of Sciences, University of Kirkuk, Iraq
   E-mail: nabilahibrahim@uokirkuk.edu.iq

Abstract: This research addresses the development of an effective numerical method for solving nonlinear Volterra integral equations of the second kind based on the Newton–Cantorovich method for bypassing nonlinearity and converting the problem into a series of successive linear equations. This technique is combined with an adaptive trapezoidal rule to improve the accuracy of numerical integration by adjusting the step size according to the behavior of the function. This combination contributes to reducing cumulative error and enhancing convergence stability during iteration. The method was implemented using MATLAB and tested on several standard examples. The results showed high agreement with exact solutions and significant improvement compared to traditional methods such as Simpson's rule, confirming the efficiency and computational accuracy of the proposed method.

Keywords: Newton–Kantorovich Method, Nonlinear Volterra Integral Equations, Adaptive Trapezoidal Rule, Numerical Quadrature, Iterative Linearization, Nonlinear Operator Equations

Citation: Hameed, S. , Jasim, A. M & Aziz, N. I. Newton–Kantorovich Based Numerical Methods for Nonlinear Integral Equations. Central Asian Journal of Mathematical Theory and Computer Sciences 2026, 7(2), 190-197

Received: 10th Dec 2025
Revised: 21th Jan 2026
Accepted: 04th Feb 2026
Published: 19th Mar 2026

1. Introduction

The role of equations in mathematical analysis and applied mathematics is very important because of their high ability to describe complex phenomena with cumulative effects, memory, and interactions in time or space. Integral equations occur naturally in a broad range of applications in physics, mechanics, control theory, biology, economics, and signal processing. In many applications, integral equations give a more accurate and complete description of real-world systems than differential equations.

Among various kinds of integral equations, nonlinear integral equations are more difficult from the theoretical as well as computational point of view. The presence of nonlinearity complicates the process of finding a closed-form solution, and hence numerical methods become a prerequisite for their analysis. Therefore, the development of efficient, accurate, and stable numerical methods for solving nonlinear integral equations has emerged as an area of research.

Among the most effective methods for solving problems, which is an extension of the classical Newton iteration procedure to infinite-dimensional spaces, which can then be solved iteratively to refine the approximate solution. Because of its sound theoretical foundation and rapid convergence rate, the Newton-Kantorovich method has been successfully applied to several nonlinear operator equations.

Nevertheless, the efficiency of the Newton-Kantorovich method is highly dependent on the numerical methods employed to help understand the equations. In this respect, the choice of an appropriate quadrature formula is a crucial step in determining the efficiency of the overall algorithm.

$$"\varphi(x) = f(x) + \lambda \int_{\{a(x)\}}^{\{b(x)\}} K\big(x, t, \varphi(t)\big) dt"$$

In this formulation, $K(x,t)$is known as the equation, while $a(x)$and $b(x)$represent the integration bounds. It is assumed that both the kernel $K(x,t)$and the given function $f(x)$are known functions possessing sufficient smoothness to ensure the well-posedness of the problem. The parameter $\lambda$is a constant that typically reflects the intensity of the integral operator or the relative contribution of the kernel within the equation.

Inspired by these ideas. The proposed method will be able to provide high accuracy with low computational cost, as demonstrated in the numerical examples.

Since the scope of the current study is limited to these particular type of integral equations, the discussion below is limited. This form of the Volterra equation has a clear mathematical structure that makes it easy to develop a theoretical study and efficient numerical methods. Therefore, the equation can be written as:

$$"\phi(x) = f(x) + \lambda \int axK\big(x, t, \phi(t)\big)dt, x \in [a, b]."$$

The equations have an important as has been established in the modeling of a large number of phenomena, especially in areas of study where cumulative and time-dependent effects are of prime importance, such as population dynamics, the spread of diseases, and semiconductor device analysis. The theoretical bases of integral equations were developed towards the end of the nineteenth century, with the pioneering work of Volterra providing the foundation for the mathematical analysis of integral equations. The earlier terminology was introduced by Dubois-Reymond, who first coined the phrase "integral equation,".

However, owing to the existence of some inherent nonlinearity in most models, it is often difficult to find exact analytical solutions for nonlinear Volterra integral equations. This has led to ongoing research efforts to develop efficient numerical solutions for finding approximate solutions. Among various methods that have been developed in the literature, quadrature methods have gained popularity owing to their simplicity and efficiency. It is pertinent to mention here that the ongoing research efforts have made use of linearization techniques such as the Newton-Kantorovich method along with traditional numerical methods.

In this work, a new numerical technique is presented by making use of the adaptive trapezoidal rule as the primary integration technique. Unlike other fixed-step integration methods, the adaptive technique permits the integration grid to be varied according to the characteristics of the integrand, which leads to improved accuracy with reduced computational effort. This explains why the new technique is more appropriate for solving the equations.

Specifically, the proposed scheme utilizes the same partitioning strategy as the classical trapezoidal rule over the subintervals $[x_i, x_{i+1}]$, for $i = 0, \ldots, j-2$. For the final subinterval, however, the method introduces additional evaluation points to further enhance accuracy. These points, denoted by $x_{j-3/4}$, $x_{j-1/2}$, and $x_{j-1/4}$, serve as intermediate nodes between $x_{j-1}$and $x_j$on the interval $[x_{j-1}, x_j]$, and are defined accordingly.

$$x_{\{j-1\}}, \quad x_{\{j-\frac{3}{4}\}}, \quad x_{\{j-\frac{1}{2}\}}, \quad x_{\{j-\frac{1}{4}\}}$$

“An approximate form of the equation can be written as”:

$$" \int_a^{\{x_j\}} K\Big(x_j, t, \varphi(t)\Big)\, dt"$$

$$" = \sum_{\{i=1\}}^{\{j-2\}} \int_{\{x_i\}}^{\{x_{\{i+1\}}\}} K\Big(x_j, t, \varphi(t)\Big)\, dt "$$

$$" + \int_{\{x_{\{j-\frac{1}{2}\}}\}}^{\{x_{\{j-1\}}\}} K\left(x_j, t, \varphi(t)\right) dt "$$

$$" + \int_{\{x_{\{j-1\}}\}}^{\{x_j\}} K\left(x_j, t, \varphi(t)\right) dt "$$

where α = 1/4, α = 1/2, or α = 3/4, and x ∈ [a, b].

2. Materials and Methods

Describe of the Numerical Procedure

The nonlinear integral equation is solved using the Newton-Kantorovich iterative method, resulting in the following formulation:

(2)

$$\varphi_{k(x)} = \varphi_{\{k-1\}(x)} + y_{\{k-1\}(x)}$$

$$y_{\{k-1\}(x)} = r_{\{k-1\}(x)} + \int_a^x K'_{\varphi(x,t,\varphi_{\{k-1\}(t)})y_{\{k-1\}(x)dt}}$$

$$r_{\{k-1\}(x)} = f(x) - \varphi_{\{k-1\}(x)} + \int_a^x K\left(x, t, \varphi(t)\right)dt$$

From equation (2), it follows that

(3)

$$y_{\{k-1\}(x)} = f(x) - \varphi_{\{k-1\}(x)} + \int_a^x K\left(x, t, \varphi_{\{k-1\}(t)}\right)dt + \int_a^x K'_{\varphi(x,t,\varphi_{\{k-1\}(t)})y_{\{k-1\}(t)dt}}$$

At this point in the numerical algorithm. This numerical integration method is particularly suited to improving the accuracy of the integration by adaptively refining the mesh size of the interval of integration based on the behavior of the integrand. In contrast to other fixed mesh size quadrature rules, the adaptive trapezoidal rule changes the mesh size depending on whether the function is varying rapidly or slowly.

This adaptive strategy assists in minimizing the accumulation of errors during the numerical integration procedure and provides a good approximation of the integral terms. This is particularly significant in the context of nonlinear Volterra integral equations, in which the error during the evaluation of the integral terms may propagate during the iterative procedure and influence the convergence of the overall numerical procedure.

Furthermore, the use of adaptive quadrature assists in enhancing efficiency during computations by concentrating on calculations only in regions where they are most required, as opposed to regions that are less required. This makes the adaptive trapezoidal rule applicable for use during integration in iterative procedures such as the Newton-Kantorovich method.

Therefore, by employing this numerical integration method, the equation is approximated to a solution that can be easily handled. This method provides a formulation that acts as a stable platform for the subsequent numerical iterations and provides a better approximation as follows.

$$" \int_a^x K(x_j, t, \varphi_{\{k-1\}(t)})dt = \sum_{\{i=0\}}^{\{j-2\}} \int_{\{x_i\}}^{\{x_{\{i+1\}}\}} K(x_j, t, \varphi_{\{k-1\}(t)})dt + \int_{\left\{x_{\left\{j-\frac{1}{2}\right\}}\right\}}^{\{x_{\{j-1\}}\}} K(x_j, t, \varphi_{\{k-1\}(t)})dt "$$

$$= \sum_{\{i=0\}}^{\{j-2\}} \frac{1}{2}\left(K\left(x_j, t_{\{i+1\}}, \varphi_{\{k-1\}(t_{\{i+1\}})}\right) + K(x_j, t_i, \varphi_{\{k-1\}(t_i)})\right) h$$

$$+\frac{1}{2}\left(K\left(x_j, t_{\left\{j-\frac{1}{2}\right\}}, \varphi_{\{k-1\}\left(t_{\left\{j-\frac{1}{2}\right\}}\right)} \right) + K\left(x_j, t_{\{j-1\}}, \varphi_{\{k-1\}(t_{\{j-1\}})}\right) \right) \frac{h}{2}$$

$$+\frac{1}{2}\left(K\left(x_j, t_j, \varphi_{\{k-1\}(t_j)}\right) + K\left(x_j, t_{\left\{j-\frac{1}{2}\right\}}, \varphi_{\{k-1\}\left(t_{\left\{j-\frac{1}{2}\right\}}\right)} \right) \right) \frac{h}{2}$$

$$= \sum_{\{i=0\}}^{\{j-2\}} \left(\frac{h}{2}\right) K\{j, i+1, i+1\}$$

$$" \sum \{j, i+1, i+1\} "$$

$$" + \sum_{\{i=0\}}^{\{j-2\}} \left(\frac{h}{2}\right) K\{j, i, i\}"$$

$$+ \left(\frac{h}{4}\right)\left(K_{\{j,j-1,j-1\}} + 2K_{\left\{j,j-\frac{1}{2},j-\frac{1}{2}\right\}} + K_{\{j,j,j\}} \right)$$

And

$$" \int_a^x K'_{\varphi(x,t,\varphi_{\{k-1\}(t)})y_{\{k-1\}(t)dt}} "$$

$$= \sum_{\{i=0\}}^{\{j-2\}} \int_{\{x_i\}}^{\{x_{\{i+1\}}\}} K_{,}^{\varphi(x_j,t,\varphi_{\{k-1\}(t)})y_{\{k-1\}(t)dt}}$$

$$" + \int_{\left\{x_{\left\{j-\frac{1}{2}\right\}}\right\}}^{\{x_{\{j-1\}}\}} K_{,}^{\varphi(x_j,t,\varphi_{\{k-1\}(t)})y_{\{k-1\}(t)dt}} "$$

$$" + \int_{\{x_j\}}^{\left\{x_{\left\{j-\frac{1}{2}\right\}}\right\}} K_{,}^{\varphi(x_j,t,\varphi_{\{k-1\}(t)})y_{\{k-1\}(t)dt}} "$$

$$= \sum_{\{i=0\}}^{\{j-2\}} \frac{1}{2}\left(K'_{\varphi\left(x_j,t_{\{i+1\}},\varphi_{\{k-1\}\left(t_{\{i+1\}}\right)}\right)y_{\{k-1\}\left(t_{\{i+1\}}\right)}} + K'_{\varphi\left(x_j,t_i,\varphi_{\{k-1\}(t_i)}\right)y_{\{k-1\}(t_i)}} \right) h$$

$$" + \frac{1}{2}\left(K'_{\varphi\left(x_j,t_{\left\{j-\frac{1}{2}\right\}},\varphi_{\{k-1\}\left(t_{\left\{j-\frac{1}{2}\right\}}\right)}\right)y_{\{k-1\}\left(t_{\left\{j-\frac{1}{2}\right\}}\right)}} \right) + K'\_\varphi(x\_j, t\_\{j-1\}, \varphi\_\{k-1\}(t\_\{j-1\})) \; y\_\{k-1\}(t\_\{j-1\}) \;) \; h/2"$$

$$" + \frac{1}{2}\left(K'_{\varphi\left(x_j, t_j, \varphi_{\{k-1\}(t_j)}\right) y_{\{k-1\}(t_j)}} \right) + K'\_\varphi(x\_j, t\_\{j-1/2\}, \varphi\_\{k-1\}(t\_\{j-1/2\}))\ y\_\{k - 1\}(t\_\{j-1/2\})\ )\ h/2"$$

$$= \sum_{\{i=0\}}^{\{j-2\}} \left(\frac{h}{2}\right) K_{,}^{\{j,i+1,i+1\} y_{\{k-1,i+1\}}} + \sum_{\{i=0\}}^{\{j-2\}} \left(\frac{h}{2}\right) K_{,}^{\{j,i,i\} y_{\{k-1,i\}}}$$

$$" + \left(\frac{h}{4}\right)\ (K'_{\{j,j-1,j-1\} y_{\{k-1,j-1\}}}\} + 2K'_{\left\{j,j-\frac{1}{2},j-\frac{1}{2}\right\} y_{\left\{k-1,j-\frac{1}{2}\right\}}} + K'\_\{j,j,j\}\ y\_\{k-1,j\})"$$

So (3) will be

$$y_{\{k-1\}(x)} = f(x) - \varphi_{\{k-1\}(x)} + \sum_{\{i=0\}}^{\{j-2\}} \left(\frac{h}{2}\right) K\{j, i+1, i+1\} + \sum_{\{i=0\}}^{\{j-2\}} \left(\frac{h}{2}\right) K\{j,i,i\} + \left(\frac{h}{4}\right)\left(K_{\{j,j-1,j-1\}} + 2K_{\left\{j,j-\frac{1}{2},j-\frac{1}{2}\right\}} + K_{\{j,j,j\}} \right)$$

$$+ \sum_{\{i=0\}}^{\{j-2\}} \left(\frac{h}{2}\right) K_{,}^{\{j,i+1,i+1\} y_{\{k-1,i+1\}}} + \sum_{\{i=0\}}^{\{j-2\}} \left(\frac{h}{2}\right) K_{,}^{\{j,i,i\} y_{\{k-1,i\}}}$$

$$" + \left(\frac{h}{4}\right)\ (K'_{\{j,j-1,j-1\} y_{\{k-1,j-1\}}}\} + 2K'_{\left\{j,j-\frac{1}{2},j-\frac{1}{2}\right\} y_{\left\{k-1,j-\frac{1}{2}\right\}}} + K'\_\{j,j,j\}\ y\_\{k-1,j\})"$$

“It remains to replace (y_{k-1}(x)) by (\phi_k(x) - \phi_{k-1}(x)) and to set (x = x_j) for (j = 0, 1, 2,…… n); we then obtain”.

$$"\varphi_{\{k,j\}} = f_j + \sum_{\{i=0\}}^{\{j-2\}} \left(\frac{h}{2}\right) K\{j, i+1, i+1\} + \sum_{\{i=0\}}^{\{j-2\}} \left(\frac{h}{2}\right) K\{j,i,i\}" + \left(\frac{h}{4}\right)\left(K_{\{j,j-1,j-1\}} + 2K_{\left\{j,j-\frac{1}{2},j-\frac{1}{2}\right\}} + K_{\{j,j,j\}} \right)$$

$$" + \sum_{\{i=0\}}^{\{j-2\}} \left(\frac{h}{2}\right) K_{,}^{\{j,i+1,i+1\}\left(\varphi_{\{k,i+1\}} - \varphi_{\{k-1,i+1\}} \right)} + \sum_{\{i=0\}}^{\{j-2\}} \left(\frac{h}{2}\right) K_{,}^{\{j,i,i\}\left(\varphi_{\{k,i\}} - \varphi_{\{k-1,i\}} \right)}"$$

$$+ \left(\frac{h}{4}\right)\ (K'_{\{j,j-1,j-1\}\left(\varphi_{\{k,j-1\}} - \varphi_{\{k-1,j-1\}} \right)}) + 2K'_{\left\{j,j-\frac{1}{2},j-\frac{1}{2}\right\}\left(\varphi_{\left\{k,j-\frac{1}{2}\right\}} - \varphi_{\left\{k-1,j-\frac{1}{2}\right\}} \right)} + K'\_\{j,j,j\}\ (\ \varphi\_\{k,j\} - \varphi\_\{k-1,j\}\ )\ )$$

“After that, we obtain the following final result”.

$$\boldsymbol{\left(1 - \left(\frac{h}{4}\right) K'_{\{j,j,j\}}\right) \varphi_{\{k,j\}} = f_j}$$

$$" + \sum_{\{i=0\}}^{\{j-2\}} \left(\frac{h}{2}\right) K\{j, i+1, i+1\} + \sum_{\{i=0\}}^{\{j-2\}} \left(\frac{h}{2}\right) K\{j,i,i\}"$$

$$\boldsymbol{" + \left(\frac{h}{4}\right)\left(K_{\{j,j-1,j-1\}} + 2K_{\left\{j,j-\frac{1}{2},j-\frac{1}{2}\right\}} + K_{\{j,j,j\}} \right)"}$$

$$\boldsymbol{+ \sum_{\{i=0\}}^{\{j-2\}} \left(\frac{h}{2}\right) K_{,}^{\{j,i+1,i+1\}\left(\varphi_{\{k,i+1\}} - \varphi_{\{k-1,i+1\}} \right)} + \sum_{\{i=0\}}^{\{j-2\}} \left(\frac{h}{2}\right) K_{,}^{\{j,i,i\}\left(\varphi_{\{k,i\}} - \varphi_{\{k-1,i\}} \right)}}$$

$$\boldsymbol{+ \left(\frac{h}{4}\right)\left(K'_{\{j,j-1,j-1\}\left(\varphi_{\{k,j-1\}} - \varphi_{\{k-1,j-1\}} \right)} \right) + 2K'_{\left\{j,j-\frac{1}{2},j-\frac{1}{2}\right\}\left(\varphi_{\left\{k,j-\frac{1}{2}\right\}} - \varphi_{\left\{k-1,j-\frac{1}{2}\right\}} \right)} - K'\_\{j,j,j\}\ \varphi\_\{k - 1,j\}\ );}$$

$$\varphi_{\{k,j\}} = \frac{1}{1 - \left(\frac{h}{4}\right) K'_{\{j,j,j\}}} [f\_j + \sum_{\{i=0\}}^{\{j-2\}} \left(\frac{h}{2}\right) K\{j, i+1, i+1\}$$

$$" + \sum_{\{i=0\}}^{\{j-2\}} \left(\frac{h}{2}\right) K\{j, i, i\} + \left(\frac{h}{4}\right) \left(K_{\{j,j-1,j-1\}} + 2K_{\left\{j,j-\frac{1}{2},j-\frac{1}{2}\right\}} + K_{\{j,j,j\}} \right) "$$

$$" + \sum_{\{i=0\}}^{\{j-2\}} \left(\frac{h}{2}\right) K_{,}^{\{j,i+1,i+1\}(\ \varphi_{\{k,i+1\}} - \varphi_{\{k-1,i+1\}})} + \sum_{\{i=0\}}^{\{j-2\}} \left(\frac{h}{2}\right) K_{,}^{\{j,i,i\}(\ \varphi_{\{k,i\}} - \varphi_{\{k-1,i\}})} "$$

$$+ (h/4)\ (\ K'_{\{j,j-1,j-1\}\left(\varphi_{\{k,j-1\}} - \varphi_{\{k-1,j-1\}}\right)}) + 2K'_{\left\{j,j-\frac{1}{2},j-\frac{1}{2}\right\}\left(\varphi_{\left\{k,j-\frac{1}{2}\right\}} - \varphi_{\left\{k-1,j-\frac{1}{2}\right\}}\right)} - K'\_\{j,j,j\}\ \varphi\_\{k - 1, j\}\ )];$$

Starting from an initial guess $\phi_1(x)$, an iterative sequence $\{\phi_k\}_{k\geq 1}$is constructed to generate successive approximations to the solution of the integral equation. Before proceeding with the iterative process, it is essential to evaluate the quantity $\phi_k(x_{j-1/2})$. This value is obtained by replacing $x_j$with $x_{j-1/2}$in the previously derived formulation.

In order to improve the quality of the numerical approximation and further suppress discretization errors, $\phi_k(x_{j-1/2})$is approximated by the arithmetic mean of the function values at the neighboring nodes, namely $\phi_k(x_{j-1})$and $\phi_k(x_j)$. This averaging strategy contributes to enhanced stability and accuracy of the iterative scheme.

3. Results and Discussion

Numerical Results

This section is concerned with the computational implementation of the proposed methods. The methods are implemented using numerical algorithms developed in MATLAB and tested on a set of the equations. The numerical experiments are used to evaluate the performance of the method in terms of accuracy and efficiency.

"Example 1 .

$$"\varphi(x) - \int_0^x \varphi^{2(t)dt} = \sin x + \left(\frac{1}{4}\right)\sin(2x) - \left(\frac{1}{2}\right)x,\ \ 0 \leq x, t \leq 1."$$

such that the function $f(x)$ is adopted as the initial approximation, whereas the same result is expressed as:

$$\phi(x) = \sin\ x.$$

"Example 2 .

$$"\varphi(x) - \int_0^x t\ e^{\{x\}\varphi^{2(t)dt}} = x^2 - \left(\frac{1}{6}\right)x^6 e^{\{x\}},\ \ 0 \leq x, t \leq 1."$$

with the same result defined as
ϕ(x)=x2.

Example 3 .

$$"\varphi(x) - \int_0^x e^{\{x\ t\}\varphi'(t)}, dt = \sqrt{x} + (x + 1)e^{\{x\}} - e^{\{x\}},\ \ 0 \leq x, t \leq 1."$$

with the same result defined as

$$\varphi(x) = \sqrt{x}$$

Placement of Tables and Figures

Table 1. Exact and numerical equation in ex1 evaluated at selected points, together with a comparison of the corresponding errors and those reported in [7].

x	Exact φ	App $\widetilde{\varphi}$	Error	Error [7]
0	0	0	0	0
0.1	9.9833e-02	9.9832e-02	4.4657e-07	3.3327e-04
0.2	1.9867e-01	1.9866e-01	4.2686e-06	1.5351e-03
0.3	2.9552e-01	2.9550e-01	1.4978e-05	6.6506e-03
0.4	3.8942e-01	3.8938e-01	3.6372e-05	1.3901e-02
0.5	4.7943e-01	4.7935e-01	7.2687e-05	2.4222e-02
0.6	5.6464e-01	5.6451e-01	1.2879e-04	4.6982e-02
0.7	6.4422e-01	6.4400e-01	2.1039e-04	6.8954e-02
0.8	7.1736e-01	7.1703e-01	3.2430e-04	9.5916e-02
0.9	7.8333e-01	7.8284e-01	4.7870e-04	1.4380e-01
1	8.4147e-01	8.4078e-01	6.8345e-04	1.8539e-01

Table 2. results for ex2 showing exact solutions and their corresponding approximations at selected points.

x	Exact s φ	App $\widetilde{\varphi}$	Error
0	0	0	0
0.1	1.0000e-02	1.0000e-02	7.5406e-08
0.2	4.0000e-02	4.0001e-02	1.0921e-06
0.3	9.0000e-02	9.0005e-02	5.0094e-06
0.4	1.6000e-01	1.6001e-01	1.4369e-05
0.5	2.5000e-01	2.5003e-01	3.1925e-05
0.6	3.6000e-01	3.6006e-01	6.0453e-05
0.7	4.9000e-01	4.9010e-01	1.0271e-04
0.8	6.4000e-01	6.4016e-01	1.6153e-04
0.9	8.1000e-01	8.1023e-01	2.3987e-04
1	1.0000e+00	1.0003e+00	3.4105e-04

Table 3. results for ex3 showing exact solutions and their corresponding approximations at selected points.

x	Exact φ	App $\widetilde{\varphi}$	Error (M2)
0	0	0	0
0.1	3.1623e-01	3.1476e-01	1.4603e-03
0.2	4.4721e-01	4.4560e-01	1.6051e-03
0.3	5.4772e-01	5.4608e-01	1.6424e-03
0.4	6.3246e-01	6.3082e-01	1.6338e-03
0.5	7.0711e-01	7.0550e-01	1.5978e-03
0.6	7.7460e-01	7.7305e-01	1.5434e-03
0.7	8.3666e-01	8.3518e-01	1.4761e-03
0.8	8.9443e-01	8.9302e-01	1.4003e-03
0.9	9.4868e-01	9.4736e-01	1.3191e-03
1	1.0000e+00	9.9876e-01	1.2352e-03

4. Conclusion

In the current research. The proposed approach converts the original nonlinear Volterra integral equation into a series of linear equations, which can be solved using an iterative process.

One of the main advantages of the proposed numerical approach is that it employs the adaptive trapezoidal rule as the primary quadrature method. The adaptive trapezoidal rule is more effective than the traditional fixed-step quadrature method because it adjusts the size of the quadrature intervals based on the behavior of the integrand function. This adaptive process helps to improve the accuracy of the numerical solution without compromising computational efficiency.

The obtained results show that the method gives very accurate approximations compared with the exact solutions, whenever they are available. In addition, it is clear from the results that the adaptive trapezoidal rule gives more accurate results compared with Simpson's rule, which has been used in the previous studies.

An important observation arising from the numerical experiments is the significant influence of the mid-point (odd-index) evaluation, denoted by $k(x_{j-\frac{1}{2}})$, on the quality of the approximate solution. The choice of this point for evaluation is very important in the control of errors and convergence during the Newton-Kantorovich iteration.

In conclusion, the results have confirmed reliable for solving the equations. The strategy based on the Newton-Kantorovich linearization technique and adaptive numerical integration is robust and can be applied to other nonlinear problems".

REFERENCES

[1] F. Abdelwahid, "Adomian decomposition method applied to nonlinear integral equations," Alexandria Journal of Mathematics, vol. 1, no. 1, pp. 11–18, 2010.

[2] E. Babolian and A. Shahsavaran, "Numerical solution of nonlinear Fredholm and Volterra integral equations of the second kind using Haar wavelets and collocation method," J. Sci. Tarbiat Moallem University, vol. 7, no. 3, pp. 213–222, 2007.

[3] J. H. Gordis and B. Neta, "An adaptive method for the numerical solution of Volterra integral equations," in Recent Advances in Applied and Theoretical Mathematics, N. E. Mastorakis, Ed. Athens: WSEAS Press, 2000, pp. 1–7.

[4] A. V. Manzhirov and A. D. Polyanin, Handbook of Integral Equations, 2nd ed. London, U.K.: Taylor & Francis, 2008.

[5] M. Nadir and B. Gagui, "Two points for the adaptive method for the numerical solution of Volterra integral equations," Int. J. Math. Manuscripts, vol. 1, no. 2, pp. 133–140, 2007.

[6] M. Nadir and A. Rahmoune, "Modified method for solving linear Volterra integral equations of second kind," Int. J. Math. Manuscripts, pp. 141–146, 2007.

[7] J. Saberi-Nadjafi and M. Heidari, "Solving nonlinear integral equations in the Urysohn form by Newton–Kantorovich quadrature method," Computers & Mathematics with Applications, vol. 60, pp. 2058–2065, 2010.

Tambahan Referensi (hingga 15):

[8] K. Atkinson, The Numerical Solution of Integral Equations of the Second Kind. Cambridge, U.K.: Cambridge University Press, 1997.

[9] R. Kress, Linear Integral Equations, 3rd ed. New York, NY, USA: Springer, 2014.

[10] P. Linz, Analytical and Numerical Methods for Volterra Equations. Philadelphia, PA, USA: SIAM, 1985.

[11] H. Brunner, Collocation Methods for Volterra Integral and Related Functional Differential Equations. Cambridge, U.K.: Cambridge University Press, 2004.

[12] A. M. Wazwaz, Linear and Nonlinear Integral Equations: Methods and Applications. Berlin, Germany: Springer, 2011.

[13] L. M. Delves and J. L. Mohamed, Computational Methods for Integral Equations. Cambridge, U.K.: Cambridge University Press, 1985.

[14] M. A. Noor and S. T. Mohyud-Din, "Variational iteration method for solving nonlinear integral equations," Applied Mathematics and Computation, vol. 167, no. 1, pp. 219–229, 2005.

[15] J. B. Conway, A Course in Functional Analysis, 2nd ed. New York, NY, USA: Springer, 1990.