Hindawi
Journal of Applied Mathematics
Volume 2019, Article ID 5678103, 9 pages
https://doi.org/10.1155/2019/5678103

Research Article

Numerical Techniques for Solving Linear Volterra Fractional Integral Equation

Safa' Hamdan, Naji Qatanani, and Adnan Daraghmeh

Department of Mathematics, Faculty of Science, An-Najah National University, Nablus, State of Palestine

Correspondence should be addressed to Naji Qatanani; nqatanani@najah.edu and Adnan Daraghmeh; adn.daraghmeh@najah.edu

Received 2 May 2019; Accepted 24 July 2019; Published 20 December 2019

Academic Editor: Mustafa Inc

Two numerical techniques, namely, Haar Wavelet and the product integration methods, have been employed to give an approximate solution of the fractional Volterra integral equation of the second kind. To test the applicability and efficiency of the numerical method, two illustrative examples with known exact solution are presented. Numerical results show clearly that the accuracy of these methods are in a good agreement with the exact solution. A comparison between these methods shows that the product integration method provides more accurate results than its counterpart.

1. Introduction

Fractional integral equations of the second kind have attracted the attention of many scientists and researchers in recent years. These equations appear frequently in heat transformations and heat radiation, population growth model, biological species living together, porous media, rheology, control, and electro-chemistry (see [1–5] for more details). The concept of fractional calculus is now considered as a partial technique in many branches of science including physics (Oldham and Spanier [6]). Recently Srivastava et al. [7] gave the model of under actuated mechanical system with fractional order derivative and Sharma [8] studied advanced generalized fractional kinetic equation in Astrophysics. Caputo refermulated the more "classic" definition of the Riemann–Liouville fractional derivatives in order to use integer order initial conditions to solve his fractional order differential Equation [9]. Kowankar and Gangal reformulated the Riemann–Liouville fractional derivative in order to differentiate nowhere differentiable fractal functions [10]. Abdou et al. [11] have implemented the Toeplitz matrix and product Nystrom methods for solving singular integral equation with logarithmic kernel and Hilbert kernel. They concluded that these methods give a very accurate approximation for these types of kernels. In [12] Hamaydi and Qatanani have solved Linear Fuzzy Volterra Integral Equation. In addition, Hamdan [13] has employed several numerical methods for solving Volterra fractional integral equations. In recent years, numerous methods have been proposed for solving fractional Volterra integral Equations [2, 14, 15]. Lepik [16] has solved fractional integral equations by the Haar wavelet method. Fractional linear multistep methods have been employed by Lubich [17] to solve Abel's Volterra integral equation of the second kind. Bruner [18] has solved Volterra integral and related functional equations by using the collocation method. Moreover, collocation method based on the orthogonal polynomials is presented to solve fractional integral Equations [19]. Other numerical methods on fractional integral equations are hybrid collocation [20], smoothing technique [15], piecewise constant orthogonal functions approximation [14], the Galerkin method [2], Berstein's approximation [19, 21], the Simpson 3/8 rule method [22], mechanical quadrature [23], Legender Pseudo spectral [24], and iterative numerical method [25]. The fractional Volterra integral equation of the second kind under investigation is defined as follows:

$$u(x) = f(x) + \frac{1}{\Gamma(a)}\int_a^x (x-t)^{\alpha-1}k(x,t)u(t)dt, \quad 0<\alpha<1, \tag{1}$$

where $k(x,t)$ is called the kernel or the nucleus of the integral equation. The function $u(x)$ to be determined appears under the integral sign. The kernel $k(x,t)$ and the function $f(x)$ are given.

In this article, two numerical techniques, namely the Harr wavelet and the product integration method are implemented to solve the Volterra fractional integral Equation (1). A comparison between these methods is carried out by solving some test examples.

The paper is organized as follows: In Section 2 the Haar wavelet method is introduced. The product integration method used to approximate solution of the fractional integral equation is addressed in Section 3. The proposed methods are implemented using numerical examples with known analytical solution by applying MAPLE software in Section 4. Conclusions are given in Section 5.

2. The Haar Wavelet Method

The Haar functions are an orthogonal family of switched rectangular waveforms where amplitudes can differ from one function to another. They are defined in the interval $[0, 1]$. However, we are interseted in defining these functions for the more general interval $[A, B]$. First, we define the integer $M = 2^I$ (I is the maximal level of resolution). We divide the interval $[A, B]$ into $2M$ subintervals of equal length; each subinterval has the length $\Delta z = (B - A)/2M$. Next, two parameters are introduced: the dilation parameter i for which $i = 0, 1, \dots, I$ and the translation parameter $c = 0, 1, \dots, m - 1$ (here the notation $m = 2^i$ is introduced). The wavelet number j is identified as $j = m + c + 1$, $(m > c$ and $m \neq c)$.

The j^{th} Haar wavelet is defined as:

$$h_j(x) = \begin{cases} 1, & \tau_1 \le x \le \tau_2, \\ -1, & \tau_2 \le x \le \tau_3, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where

$$\begin{aligned} \tau_1 &= A + 2c\mu\Delta z, \\ \tau_2 &= A + (2c+1)\mu\Delta z, \\ \tau_3 &= A + 2(c+1)\mu\Delta z, \\ \mu &= \tfrac{M}{m}. \end{aligned} \tag{3}$$

For $j = 1$ and $x \in [0, 1]$, the function $h_1(x)$ is the scaling function or the father wavelet for the family of the Haar wavelets which is defined as

$$h_1(x) = \begin{cases} 1, & 0 \le x < 1, \\ 0, & \text{otherwise}. \end{cases} \tag{4}$$

For $j = 2$ and $x \in [0, 1]$, the function $h_2(x)$ is the mother wavelet for the family of the Haar wavelet which is defined as

$$h_2(x) = \begin{cases} 1, & 0 \le x < \frac{1}{2}, \\ -1, & \frac{1}{2} \le x < 1, \\ 0, & \text{otherwise}. \end{cases} \tag{5}$$

For $j = 3$ and $x \in [0, 1]$, the function $h_3(x)$ is defined as

$$h_3(x) = \begin{cases} 1, & 0 \le x < \frac{1}{4}, \\ -1, & \frac{1}{4} \le x < \frac{2}{4}, \\ 0, & \text{otherwise}. \end{cases} \tag{6}$$

For $j = 4$ and $x \in [0, 1]$, the function $h_4(x)$ is defined as

Table 1: Index computations for Haar basis function.

i	0	1	1	2	2	2	2	⋯⋯
c	0	0	1	0	1	2	3	⋯⋯
$j = 2^i + c + 1$	2	3	4	5	6	7	8	⋯⋯

$$h_4(x) = \begin{cases} 1, & \frac{2}{4} \le x < \frac{3}{4}, \\ -1, & \frac{3}{4} \le x < \frac{4}{4}, \\ 0, & \text{otherwise}. \end{cases} \tag{7}$$

In fact Table 1 shows the index computations for Harr basis functions. The Haar wavelet series expansion of a given function $g(x)$ is given as [24]

$$g(x) = \sum_{j=1}^{2M} b_j h_j(x), \tag{8}$$

where $x \in [A, B]$ and $b_{j's}$ are the wavelet coefficients. There are many methods for computing $b_{j's}$, one of these is the collocation method.

The collocation points are defined as

$$x_k = A + (k - 0.5)\Delta z, \quad k = 1, 2, \dots, 2M. \tag{9}$$

Substituting Equation (9) into Equation (8) gives the discrete version, i.e.,

$$g(x_k) = \sum_{j=1}^{2M} b_j h_j(x_k). \tag{10}$$

It is convenient to put Equation (10) into the matrix form as

$$g = bH, \tag{11}$$

where g and b are $2M$-dimensional row vectors and the Haar coefficients $H(j, k) = h_j(x_k)$ is the element of a $2M \times 2M$ matrix.

Now, in virtue of Equation (9) and upon substituting Equation (10) into Equation (1) we get

$$g(x_k) = f(x_k) + \frac{1}{\Gamma(\alpha)} \int_0^{x_k} (x_k - t)^{\alpha-1} k(x_k, t) g(t) dt \tag{12}$$

or

$$\sum_{j=1}^{2M} b_j h_j(x_k) = f(x_k) + \frac{1}{\Gamma(\alpha)} \int_0^{x_k} (x_k - t)^{\alpha-1} k(x_k, t) \sum_{j=1}^{2M} b_j h_j(t) dt, \tag{13}$$

$$f(x_k) = \sum_{j=1}^{2M} b_j h_j(x_k) - \frac{1}{\Gamma(\alpha)} \int_0^{x_k} (x_k - t)^{\alpha-1} k(x_k, t) \sum_{j=1}^{2M} b_j h_j(t) dt, \tag{14}$$

$$f(x_k) = \sum_{j=1}^{2M} b_j \left[h_j(x_k) - \frac{1}{\Gamma(\alpha)} \int_0^{x_k} (x_k - t)^{\alpha-1} k(x_k, t) h_j(t) dt \right]. \tag{15}$$

Assume that

$$u_j(x_k) = \frac{1}{\Gamma(\alpha)} \int_0^{x_k} (x_k - t)^{\alpha-1} k(x_k, t) h_j(t) dt \quad (16)$$

then Equation (15) becomes

$$f(x_k) = \sum_{j=1}^{2M} b_j \left[h_j(x_k) - u_j(x_k) \right], \quad k = 1, 2, \dots, 2M. \quad (17)$$

The matrix form of Equation (17) is

$$F = b[H - U], \quad (18)$$

where $U(j,k) = u_j(x_k)$, $F(k) = f(x_k)$.

The solution of Equation (18) is

$$b = F([H - U])^{-1}. \quad (19)$$

Upon using Equation (2), then Equation (16) can be rewritten in the following form

$$\begin{cases} U(j,k) = 0, & x_k < \tau_1, \\ U(j,k) = \frac{1}{\Gamma(\alpha)} \int_{\tau_1}^{x_k} (x_k - t)^{\alpha-1} k(x_k, t) dt, & \tau_1 \le x \le \tau_2, \\ U(j,k) = \frac{1}{\Gamma(\alpha)} \left[\int_{\tau_1}^{\tau_2} (x_k - t)^{\alpha-1} k(x_k, t) dt - \int_{\tau_2}^{x_k} (x_k - t)^{\alpha-1} k(x_k, t) dt \right], & \tau_2 \le x \le \tau_3, \\ U(j,k) = \frac{1}{\Gamma(\alpha)} \left[\int_{\tau_1}^{\tau_2} (x_k - t)^{\alpha-1} k(x_k, t) dt - \int_{\tau_2}^{\tau_3} (x_k - t)^{\alpha-1} k(x_k, t) dt \right], & \tau_3 \le x_k. \end{cases} \quad (20)$$

3. The Product Integration Method

We introduce the product integration method that can be used to solve linear Volterra fractional integral equation of the second kind.

To use the product integration method as a numerical technique [3, 10], we consider the linear Volterra integral equation of the second kind which has the general form:

$$g(x) = f(x) + \lambda \int_a^x k(x,t) g(t) dt, \quad a \le x \le b, \quad (21)$$

where $k(x,t)$ is the kernel (and is known), λ is a constant parameter and $g(x)$ is unknown function to be determined. This method is based on factoring of the singularity of the kernel $k(x,t)$ as follows:

$$k(x,t) = p(x,t) G(x,t), \quad (22)$$

where $G(x,t)$ is well-behaved function and $p(x,t)$ is badly behaved function.

Set Equation (22) into Equation (21) yields

$$g(x) = f(x) + \lambda \int_a^x p(x,t) G(x,t) g(t) dt, \quad a \le x \le b. \quad (23)$$

First, we decompose the interval $[a,b]$ into N subintervals h_i where:

$$h_i = x_{(i+1)} - x_i, \quad i = 0, 1, 2, \dots, N-1 \quad (24)$$

and

$$a = x_0 < x_1 < \dots < x_N = b. \quad (25)$$

If we use the N-points quadrature rule and collocate Equation (23) at the nodes $\{x_i\}_{i=1}^N \cup \{0\}$, we get

Table 2: The exact and numerical solutions using the Haar wavelet method with $N = 64$.

x_k	Exact solution $g(x) = x^{1/2} - 1$	Numerical solution $g_h(x)$	*Error* : $\lvert g - g_h \rvert$
0.0078	−0.9116116523	−0.9116009499	$1.0702e^{-05}$
0.1172	−0.6576734015	−0.6576324164	$4.0985e^{-05}$
0.2266	−0.5240141808	−0.5239489931	$6.5188e^{-05}$
0.3359	−0.4203988440	−0.4203067536	$9.2090e^{-05}$
0.4453	−0.3326826092	−0.3325562333	$1.2638e^{-04}$
0.5547	−0.2552265445	−0.2550525941	$1.7395e^{-04}$
0.6641	−0.1850996993	−0.1848551078	$2.4459e^{-04}$
0.7734	−0.1205470450	−0.1201906958	$3.5635e^{-04}$
0.8828	−0.0604189763	−0.0598738640	$5.4511e^{-04}$
0.9922	−0.0039139093	−0.0030264584	$8.8745e^{-04}$

$$g(x_i) = f(x_i) + \lambda \int_a^{x_i} p(x_i, t) G(x_i, t) g(t) dt, \quad i = 0, 1, 2, \dots, N. \quad (26)$$

Using Lagrange interpolation polynomial, Equation (26) can be written as:

$$g(x_i) = f(x_i) + \lambda \sum_{j=0}^{i} w_{ij} G(x_i, t_j) g(t_j), \quad (27)$$

where $t_i = x_i = a + ih$ for $i = 0, 1, \dots, N$, with $h = (b-a)/N$ and w_{ij} are the weights which can be determined directly. Approximating the integral terms by a product integration using composite trapezoidal rule, where $x = x_i$, we get:

$$\int_a^{x_i} p(x_i, t) G(x_i, t) g(t) dt \approx \sum_{j=0}^{i} w_{ij} G(x_i, t_j) g(t_j), \quad i = 0, 1, 2, \dots, N. \quad (28)$$

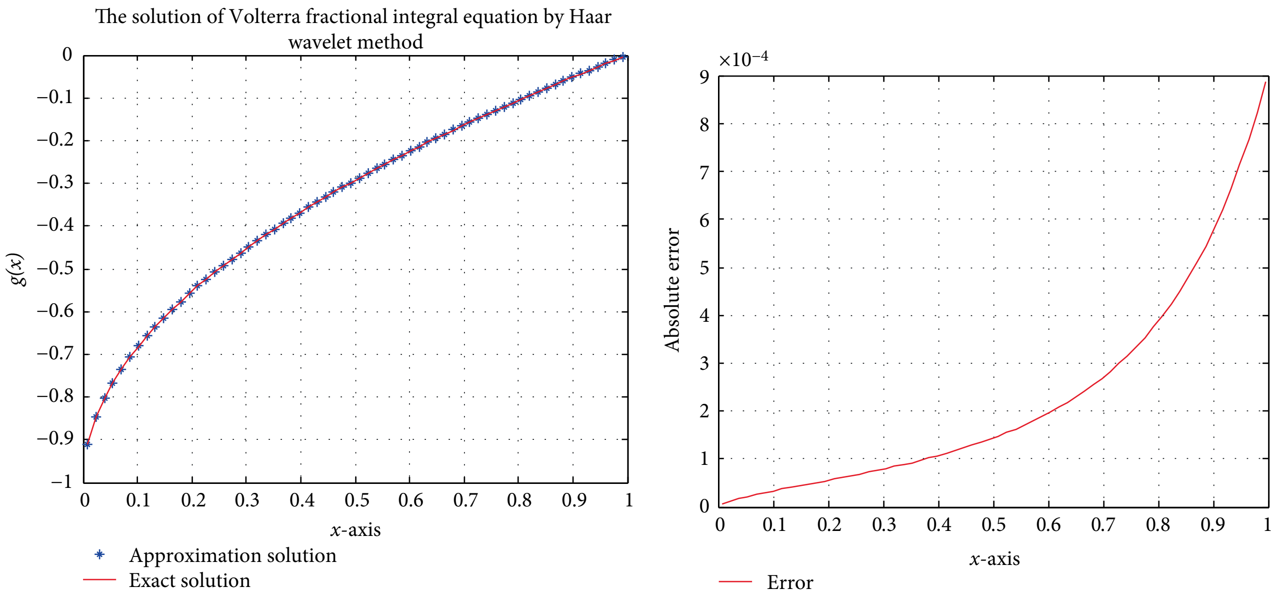
It follows that

$$\begin{aligned} \int_a^{x_i} p(x_i,t) G(x_i,t) g(t) dt &= \sum_{j=0}^{i-1} \int_{t_j}^{t_{j+1}} p(x_i,t) G(x_i,t) g(t) dt \\ &\approx \sum_{j=0}^{i-1} \int_{t_j}^{t_{j+1}} p(x_i,t) \left[\frac{(t_{j+1}-t)}{h_j} G(x_i,t_j) g(t_j) + \frac{(t-t_{j+1})}{h_j} G(x_i,t_{j+1}) g(t_j) \right] dt \\ &= \sum_{j=0}^{i} w_{ij} G(x_i,t_j) g(t_j), \end{aligned} \quad (29)$$

where

$$w_{i0} = \int_{t_0}^{t_1} p(x_i,t) \left(\frac{(t_1 - t)}{h_0} \right) dt, \quad \text{for} \quad j = 0 \quad (30)$$

$$\begin{aligned} w_{ij} = &\int_{t_j}^{t_{j+1}} p(x_i,t) \left(\frac{(t_{j+1} - t)}{h_j} \right) dt \\ &+ \int_{t_{j-1}}^{t_j} p(x_i,t) \left(\frac{(t - t_{j-1})}{h_{j-1}} \right) dt, \quad \text{for } j = 1, \dots, i-1, \end{aligned} \quad (31)$$



(a) (b)

FIGURE 1: A comparison and absolute error between exact and numerical solutions in Example 1 with $N = 64$. (a) A comparison. (b) Absolute error.

$$w_{ii} = \int_{t_{j-1}}^{t_j} p(x_i, t)\left(\frac{(t - t_{j-1})}{h_{j-1}}\right)dt, \quad \text{for } j = i, \tag{32}$$

The approximate solution to Equation (26) is determined recursively using:

$$\hat{g}(x_i) = f(x_i) + \lambda \sum_{j=0}^{i} w_{ij} G(x_i, t_j) g(t_j), \quad i = 1, 2, \ldots, N \tag{33}$$

with

$$\hat{g}_n(x_0) = f(a). \tag{34}$$

Hence, Equation (26) yields a system of linear algebraic equations:

$$B\hat{G}_n = F. \tag{35}$$

Solving Equation (35) gives

$$\hat{G}_n = B^{-1}F, \tag{36}$$

where the matrix $B = (b_{ij}), 0 \le i \le N, 0 \le j \le i - 1$ with:

$$\begin{cases} b_{00} = 1 \\ b_{ij} = 0, \quad \forall j \le i + 1 \\ b_{0j} = 0, \quad \forall j \ge 1 \\ b_{ij} = 0, \quad \forall j > i \\ b_{i0} = \lambda w_{i0} G(x_i, x_0), \quad j = 0 \\ b_{ii} = 1 - \lambda w_{ii} G(x_i, x_i), \quad j = i \\ b_{ij} = \lambda w_{ij} G(x_i, x_j), \quad j = 0, 1, \ldots, i - 1 \end{cases} \tag{37}$$

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & \ldots & 0 \\ b_{10} & b_{11} & 0 & 0 & \ldots & 0 \\ b_{20} & b_{21} & b_{22} & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ b_{N0} & b_{N1} & b_{N2} & 0 & \ldots & b_{NN} \end{bmatrix}, \tag{38}$$

$$F = [f(x_0) = f(a), f(x_1), \cdots, f(x_N) = f(b)]^T, \tag{39}$$

$$\hat{G}_n = [\hat{g}(x_0) = \hat{g}(a), \hat{g}(x_1), \cdots, \hat{g}(x_N) = \hat{g}(b)]^T. \tag{40}$$

1: Input:

- A, B: $[A, B]$ is the interval for the solution function.
- I: The maximal level of resolution.
- i: The dilation parameter.
- c: The translation parameter.
- $f(x)$: The function of the integral equation.
- $k(x, t)$: The kernel function.
- $h_1(x)$: The scaling function.
- λ: Is a constant parameter.

2: Calculate:

- The quantity $M = 2^I$.
- $N = 2M$
- The length of each subintervals $\Delta z = \frac{B-A}{N}$.
- $m = 2^i$
- The wavelet number $j = m + c + 1$.

3: Calculate: The collocation points $x_k = A + (k - 0.5)\Delta z$.
4: Calculate: $h_j(x_k), u_j(x_k)$ and $f(x_k)$.
5: Calculate:

- The matrix $[H - U]$.
- Calculate $b = F([H - U]^{-1})$.

6: Determine the solution of the linear system $g = bH$.

ALGORITHM 1: Numerical realization using the Haar wavelet method.

1: Input:

- a, b: $[a, b]$ is the interval for the solution function.
- N: The number of subdivisions of $[a, b]$.
- $f(x)$: The function of the integral equation.
- $G(x,t)$, $p(x,t)$: The kernel functions.
- λ: Is a constant parameter.
- Set $x_0 = a$ and $x_N = b$.

2: Calculate:

- $h = \frac{b-a}{N}$
- $t_i = x_i = a + ih$
- $h_i = x(i+1) - x_i$.

3: Calculate: w_{i0}, w_{ij}, w_{ii}

$$w_{i0} = \int_{t_0}^{t_1} p(x_i,t)\left(\frac{(t_1-t)}{h_0}\right)dt, \quad \text{for } j = 0$$

$$w_{ij} = \int_{t_j}^{t_{j+1}} p(x_i,t)\left(\frac{(t_{j+1}-t)}{h_j}\right)dt + \int_{t_{j-1}}^{t_j} p(x_i,t)\left(\frac{(t-t_{j-1})}{h_{j-1}}\right)dt, \quad \text{for } j = 1,\cdots,i-1$$

$$w_{ii} = \int_{t_{j-1}}^{t_j} p(x_i,t)\left(\frac{(t-t_{j-1})}{h_{j-1}}\right)dt, \quad \text{for } j = i$$

4: Solve the recurrence relation:

$$\hat{g}_n(x_0) = f(x_0) = f(a)$$

$$\hat{g}(x_i) = f(x_i) + \lambda \sum_{j=0}^{i} w_{ij} G(x_i,t_j) g(t_j), \quad i = 1,2,\cdots,N$$

or we can also solve the linear system of algebraic equations

$$\hat{G}_n = B^{-1}F$$

Algorithm 2: Numerical realization using the product integration method.

Here B is a lower triangular matrix $(N+1)*(N+1)$ and $(1-\lambda w_{ii} G(x_i,x_i)) \neq 0$.

3.1. Uniqueness of a Solution. To prove the uniqueness of the solution, we can apply the Cauchy–Minkoviski inequality (see [11] for more details) and get $|\lambda| \le 1$ which represents a sufficient condition to have a unique solution, and the value of $|g_i|$ satisfies the inequality:

$$|g_i| \le \frac{|f_i|}{1-\lambda}. \tag{41}$$

4. Numerical Examples and Results

In this section, in order to examine the accuracy of the proposed methods, we solve two numerical examples of fractional Volterra integral equations. Moreover, the numerical results will be compared with exact solution.

Example 1. Consider the linear Volterra fractional integral equation of the second kind:

$$g(x) = x^{1/2} - \frac{3}{4}\sqrt{\pi}x^2 + \frac{8}{3\sqrt{\pi}}x^{3/2} - 1 + \frac{1}{\sqrt{\pi}}\int_0^x (x-t)^{-1/2} 2t g(t)dt. \tag{42}$$

The analytical solution of Equation (42) is:

$$g(x) = x^{1/2} - 1. \tag{43}$$

We implement Algorithm 1 to solve Equation (42) using the Haar wavelet method. Table 2 displays the exact and the numerical results using the Haar wavelet method for Equation (42) with $\Delta z = 0.0156$ and $N = 64$. The maximum error with $N = 64$ is $8.8745e^{-04}$. Figure 1(a) compares both the exact and numerical solutions for the Volterra fractional integral Equation (42). Moreover, Figure 1(b) shows the absolute error between exact and numerical solutions. Now, we implement Algorithm 2 to solve Equation (42) using the Product Integration Method. We obtain the following results:

4.1. W Function. If $p(x_i,t) = (x_i - t)^{-1/2}$, then we have

$$w_{i0} = \frac{1}{h_0}\int_{t_0}^{t_1} p(x_i,t)(t_1 - t)dt, \quad \text{for } j = 0, \tag{44}$$

$$w_{i0} = \frac{1}{h}\int_{t_0}^{t_1} (x_i - t)^{-1/2}(h - t)dt, \tag{45}$$

$$w_{i0} = \frac{1}{h}\int_{t_0}^{t_1} (ih - t)^{-1/2}(h - t)dt, \tag{46}$$

$$w_{i0} = \frac{2}{3}h^{1/2}\left[3i^{1/2} + 2\left[(i-1)^{3/2} - i^{3/2}\right]\right], \tag{47}$$

$$w_{ij} = \int_{t_j}^{t_{j+1}} p(x_i,t)\left(\frac{(t_{j+1}-t)}{h_j}\right)dt + \int_{t_{j-1}}^{t_j} p(x_i,t)\left(\frac{(t-t_{j-1})}{h_{j-1}}\right)dt, \quad j = 1,\ldots,i-1, \tag{48}$$

$$w_{ij} = \int_{jh}^{(j+1)h} (x_i - t)^{-1/2}\left(\frac{(j+1)h - t}{h}\right)dt + \int_{(j-1)h}^{jh} (x_i - t)^{-1/2}\left(\frac{t-(j-1)h)}{h}\right)dt, \tag{49}$$

$$w_{ij} = \frac{4}{3}h^{1/2}\left[(i-j-1)^{3/2} + \left[(i-j+1)^{3/2} - 2(i-j)^{3/2}\right]\right], \tag{50}$$

$$w_{ii} = \frac{1}{h_{i-1}}\int_{t_{i-1}}^{t_i} p(x_i,t)(t - t_{i-1})dt, \quad j = i, \tag{51}$$

$$w_{ii} = \frac{1}{h}\int_{(i-1)h}^{ih} (x_i - t)^{-1/2}(t - (i-1)h)dt, \tag{52}$$

$$w_{ii} = \frac{1}{h}\int_{(i-1)h}^{ih} (ih - t)^{-1/2}(t - (i-1)h)dt, \tag{53}$$

$$w_{ii} = \frac{4}{3}h^{1/2}. \tag{54}$$

Table 3: The exact and numerical solutions using the product integration method with $N = 64$.

x_k	Exact solution $g(x) = x^{1/2} - 1$	Numerical solution $g_h(x)$	*Error* : $\lvert g - g_h \rvert$
0	−1	−1	0
0.125	−0.6464466094	−0.6463939905	$5.2619e^{-05}$
0.250	−0.5000000000	−0.4999385093	$6.1491e^{-05}$
0.375	−0.3876275643	−0.3875532017	$7.4363e^{-05}$
0.500	−0.2928932188	−0.2927990146	$9.4204e^{-05}$
0.625	−0.2094305849	−0.2093044528	$1.2613e^{-04}$
0.750	−0.1339745962	−0.1337942129	$1.8038e^{-04}$
0.875	−0.0645856533	−0.0643065527	$2.7910e^{-04}$
1	0	$4.7435e^{-04}$	$4.7435e^{-04}$

From above equations we obtain the linear system of algebraic equations:

$$G_n = B^{-1}F. \tag{55}$$

Table 3 displays the exact and the numerical results using the product integration method for Equation (42) with $h = 0.0156$. The maximum error with $N = 64$ is $4.7435e^{-04}$. Figure 2(a) compares both the analytical and numerical solutions for Equation (42) with $N = 64$. Moreover, Figure 2(b) shows the absolute error between exact and numerical solutions.

Example 2. Consider the Abel's integral equation of the second kind:

$$g(x) = \frac{1}{3}x^2 - x + 0.2399x^{7/3} + 0.8399x^{4/3} + 0.3733\int_0^x (x-t)^{-2/3} g(t)\,dt. \tag{56}$$

The exact solution of Equation (56) is:

$$g(x) = \frac{1}{3}x^2 - x. \tag{57}$$

Applying Algorithms 1 to Equation (56), we obtain the following results. Table 4 displays the exact and the numerical results using the Haar wavelet method for Equation (56) with $\Delta z = 0.0156$ and $N = 64$. The maximum error with $N = 64$ is $1.1113e^{-03}$. Figure 3(a) compares both the exact and numerical solutions for the Abel's integral equation (56). Moreover, Figure 3(b) shows that the absolute error between exact and numerical solutions. Applying Algorithm 2 to Equation (56) using the product integration method, we get

4.2. W Function. If $p(x_i, t) = (x_i - t)^{-2/3}$, then we have

$$w_{i0} = \frac{1}{h_0}\int_{t_0}^{t_1} p(x_i,t)(t_1 - t)\,dt, \quad \text{for } j = 0, \tag{58}$$

$$w_{i0} = \frac{1}{h}\int_{t_0}^{t_1} (x_i - t)^{-2/3}(h - t)\,dt, \tag{59}$$

$$w_{i0} = \frac{1}{h}\int_{t_0}^{t_1} (ih - t)^{-2/3}(h - t)\,dt, \tag{60}$$

$$w_{i0} = \frac{3}{4}h^{1/3}\left[4i^{1/3} + 3\left[(i-1)^{4/3} - i^{4/3}\right]\right], \tag{61}$$

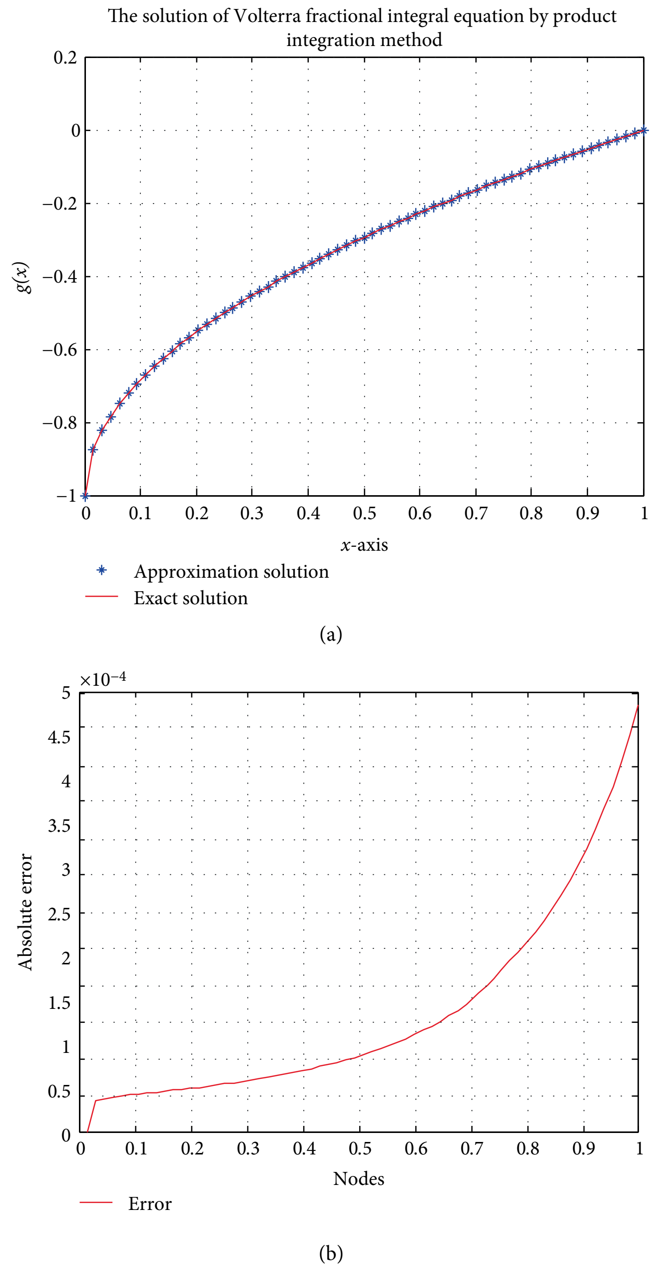


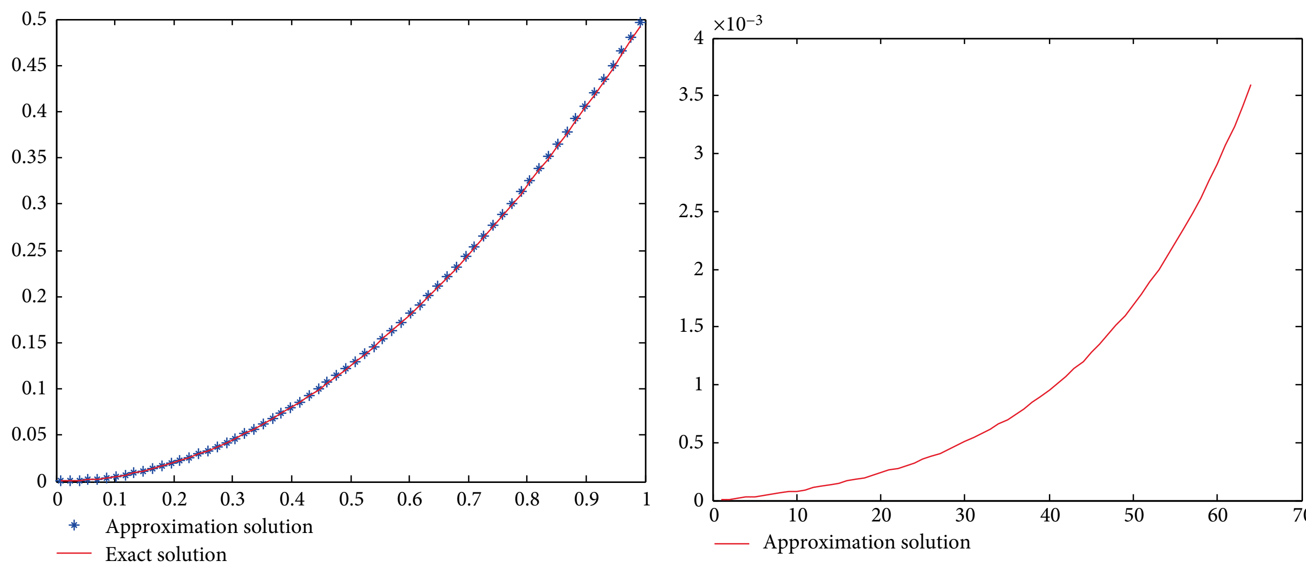
Figure 2: A comparison and absolute error between exact and numerical solutions in Example 1 with $N = 64$. (a) A comparison. (b) Absolute error.

$$w_{ij} = \int_{t_j}^{t_{j+1}} p(x_i,t)\left(\frac{(t_{j+1} - t)}{h_j}\right)dt + \int_{t_{j-1}}^{t_j} p(x_i,t)\left(\frac{(t - t_{j-1})}{h_{j-1}}\right)dt, \quad j = 1,\ldots,i-1, \tag{62}$$

$$w_{ij} = \int_{jh}^{(j+1)h} (x_i - t)^{-2/3}\left(\frac{(j+1)h - t}{h}\right)dt + \int_{(j-1)h}^{jh} (x_i - t)^{-2/3}\left(\frac{t - (j-1)h)}{h}\right)dt, \tag{63}$$

Table 4: The exact and numerical solutions using the Haar wavelet method with $N = 64$.

x_k	Exact solution $g(x) = x^{1/2} - 1$	Numerical solution $g_h(x)$	*Error* : $\lvert g - g_h\rvert$
0.0078	−0.0077921549	−0.0083481389	$5.5598e^{-04}$
0.1172	−0.1126098632	−0.1131662257	$5.5636e^{-04}$
0.2266	−0.2094523111	−0.2100916889	$6.3938e^{-04}$
0.3359	−0.2983194986	−0.2990367942	$7.1730e^{-04}$
0.4453	−0.3792114257	−0.3800030241	$7.9160e^{-04}$
0.5547	−0.4521280924	−0.4529904969	$8.6240e^{-04}$
0.6641	−0.5170694986	−0.5179996508	$9.3015e^{-04}$
0.7734	−0.5740356445	−0.5750303556	$9.9471e^{-04}$
0.8828	−0.6230265299	−0.6240817729	$1.0552e^{-03}$
0.9922	−0.6640421549	−0.6651534675	$1.1113e^{-03}$



(a) (b)

Figure 3: A comparison and absolute error between exact and numerical solutions in Example 2 with $N = 64$. (a) A comparison. (b) Absolute error.

Table 5: The exact and numerical solutions using the product integration method with $N = 64$.

x_k	Exact solution $g(x) = x^{1/2} - 1$	Numerical solution $g_h(x)$	*Error* : $\lvert g - g_h\rvert$
0	0	0	0
0.125	−0.1197916666	−0.1197818969	$9.7697e^{-05}$
0.250	−0.2291666666	−0.2291501145	$1.6552e^{-05}$
0.375	−0.3281250000	−0.3280982056	$2.6794e^{-05}$
0.500	−0.4166666666	−0.4166237294	$4.2937e^{-05}$
0.625	−0.4947916666	−0.4947244207	$6.7246e^{-04}$
0.750	−0.5625000000	−0.5623979179	$1.0208e^{-04}$
0.875	−0.6197916666	−0.6196416593	$1.5001e^{-04}$
1	−0.6666666666	−0.6664528164	$2.1385e^{-04}$

$$w_{ij} = \int_{jh}^{(j+1)h} (ih - t)^{-2/3} \left(\frac{(j+1)h - t}{h} \right) dt + \int_{(j-1)h}^{jh} (ih - t)^{-2/3} \left(\frac{t - (j-1)h}{h} \right) dt, \tag{64}$$

$$w_{ij} = \frac{9}{4} h^{1/3} \left[(i - j - 1)^{4/3} + \left[(i - j + 1)^{4/3} - 2(i - j)^{4/3} \right] \right], \tag{65}$$

$$w_{ii} = \frac{1}{h_{i-1}} \int_{t_{i-1}}^{t_i} p(x_i, t)(t - t_{i-1}) dt, \quad j = i, \tag{66}$$

$$w_{ii} = \frac{1}{h} \int_{(i-1)h}^{ih} (x_i - t)^{-2/3} (t - (i-1)h) dt, \tag{67}$$

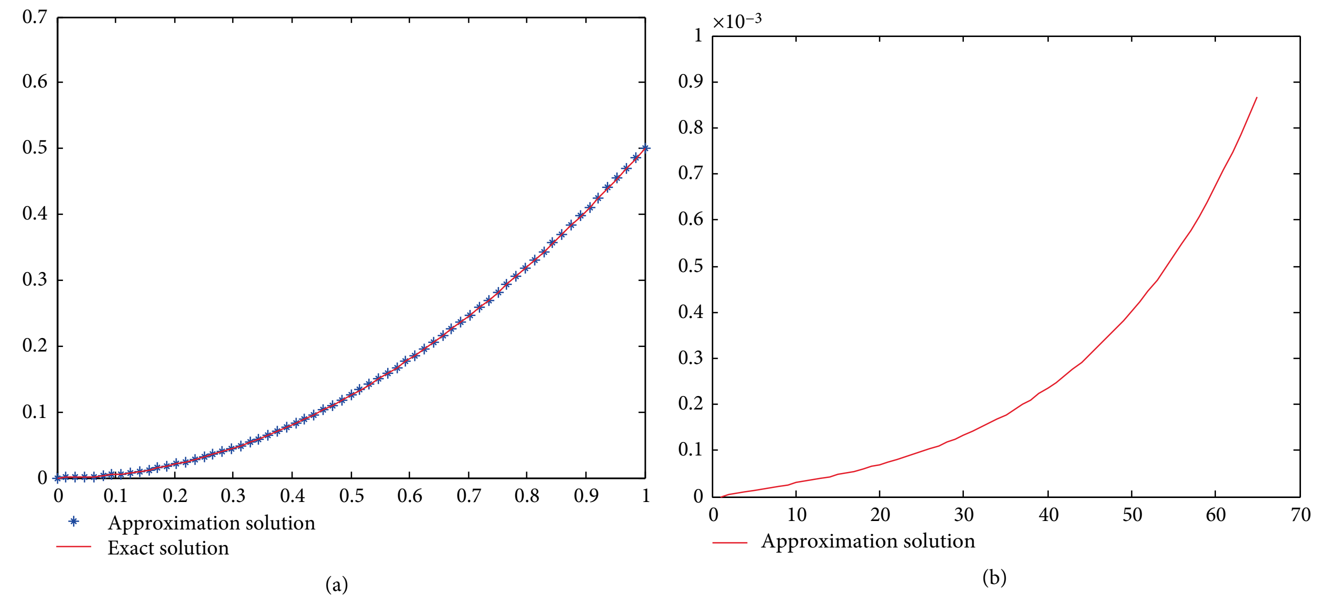


FIGURE 4: A comparison and absolute error between exact and numerical solutions in Example 2 with $N = 64$. (a) A comparison. (b) Absolute error.

$$w_{ii} = \frac{1}{h}\int_{(i-1)h}^{ih} (ih - t)^{-2/3}(t - (i-1)h)dt, \quad (68)$$

$$w_{ii} = \frac{9}{4}h^{1/3}. \quad (69)$$

Table 5 displays the exact and the numerical results using the product integration method for Equation (56) with $h = 0.0156$. The maximum error with $N = 64$ is $2.1385e^{-04}$. Figure 4(a) compares both the analytical and numerical solutions for Equation (56) with $N = 64$. Moreover, Figure 4(b) shows the absolute error between exact and numerical solutions.

5. Conclusions

In this article, two numerical schemes namely, the Harr wavelet and the product integration methods have been employed to solve Volterra fractional integral equation of the second kind. The advantages and flexibility of these techniques are demonstrated with a variety of numerical examples. The results show that both techniques are in a good agreement with the analytical solution. According to comparison of numerical results, mentioned in tables and figures, we conclude that the product integration method provides more accurate results than its counterpart and therefore is more advantageous. Moreover, in comparison of our numerical results with those presented in the literature in particular Abdo et al. [11], we observe that the Toeplitz matrix and product Nystrom methods are more effective for integral equations with logarithmic and Hilbert kernels. However, the Haar wavelet and the product integration methods are more suitable for weakly singular kernels.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] R. Agarwal, S. Jain, and R. P. Agarwal, "Solution of fractional Volterra integral equation and nonhomogeneous time fractional heat equation using integral transform of pathway type," *Progress in Fractional Differentiation and Applications*, vol. 1, no. 3, pp. 145–155, 2015.

[2] B. Bandrowski, A. Karczewska, and P. Rozmej, "Numerical solutions to integral equations equivalent to differential equations with fractional time," *International Journal of Applied Mathematics and Computer Science*, vol. 20, no. 2, pp. 261–269, 2010.

[3] A. Carpinteri and F. Mainardi, *Fractals and Fractional Calculus in Continuum Mechanics*, vol. 378, Springer, 2014.

[4] R. Metzler, W. Schick, H.-G. Kilian, and T. F. Nonnenmacher, " Relaxation in filled polymers: a fractional calculus approach," *The Journal of Chemical Physics*, vol. 103, no. 16, pp. 7180–7186, 1995.

[5] A.-M. Wazwaz, *Linear and Nonlinear Integral Equations*, vol. 639, Springer, 2011.

[6] K. Oldham and J. Spanier, *The Fractional Calculus Theory and Applications of Differentiation and Integration to Arbitrary Order*, vol. 111, Elsevier, 1st edition, 1974.

[7] T. Srivastava, A. P. Singh, and A. Himanshu, "Modeling the under-actuated mechanical system with fractional order derivative," *Progress in Fractional Differentiation and Applications*, vol. 1, no. 1, pp. 57–64, 2015.

[8] M. Sharma, M. F. Ali, and R. Jain, “Advanced generalized fractional kinetic equation in astrophysics,” *Progress in Fractional Differentiation and Applications*, vol. 1, no. 1, pp. 65–71, 2015.

[9] M. Caputo, “Linear models of dissipation whose *q* is almost frequency independent–ii,” *Geophysical Journal International*, vol. 13, no. 5, pp. 529–539, 1967.

[10] K. M. Kolwankar and A. D. Gangal, “Fractional differentiability of nowhere differentiable functions and dimensions,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 6, no. 4, pp. 505–513, 1996.

[11] M. A. Abdou, K. J. Mohamed, and A. S. Ismal, “Toeplitz matrix and product nystrom methods for solving the singular integral equation,” *Le Mathematica*, vol. 57, no. 1, pp. 21–37, 2002.

[12] J. Hamaydi and N. Qatanani, “Computational methods for solving linear fuzzy volterra integral equation,” *Journal of Applied Mathematics*, vol. 2017, Article ID 2417195, 12 pages, 2017.

[13] S. Hamdan, “Numerical methods for solving Volterra functional integral equations with applications,” An-Najah National University, 2018, MSc.Thesis.

[14] E. Babolian and S. A. Shamloo, “Numerical solution of Volterra integral and integro-differential equations of convolution type by using operational matrices of piecewise constant orthogonal functions,” *Journal of Computational and Applied Mathematics*, vol. 214, no. 2, pp. 495–508, 2008.

[15] P. Baratella and A. P. Orsi, “A new approach to the numerical solution of weakly singular Volterra integral equations,” *Journal of Computational and Applied Mathematics*, vol. 163, no. 2, pp. 401–418, 2004.

[16] U. Lepik, “Solving fractional integral equations by the Haar wavelet method,” *Applied Mathematics and Computation*, vol. 214, no. 2, pp. 468–478, 2009.

[17] Ch. Lubich, “Fractional linear multistep methods for Abel–Volterra integral equations of the second kind,” *Mathematics of Computation*, vol. 45, no. 172, pp. 463–469, 1985.

[18] H. Brunner, *Collocation Methods for Volterra Integral and Related Functional Differential Equations*, vol. 15, Cambridge University Press, 2004.

[19] K. Maleknejad, E. Hashemizadeh, and R. Ezzati, “A new approach to the numerical solution of Volterra integral equations by using Bernstein’s Approximation,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 2, pp. 647–655, 2011.

[20] Y. Cao, T. Herdman, and X. Yuesheng, “A hybrid collocation method for Volterra integral equations with weakly singular kernels,” *SIAM Journal on Numerical Analysis*, vol. 41, no. 1, pp. 364–381, 2003.

[21] S. Bhattacharya and B. N. Mandal, “Use of Bernstein Polynomials in numerical solutions of Volterra integral equations,” *Applied Mathematical Sciences*, 2008.

[22] A. Atangana and N. Bildik, “Existence and numerical solution of the Volterra fractional integral equations of the second kind,” *Mathematical Problems in Engineering*, vol. 2013, pp. 1–11, 2013.

[23] V. Kalitvin, “Numerical solution of integral equations with fractional and partial integrals and variable integration limits,” *Journal of Mathematical Sciences*, vol. 219, no. 1, 2016.

[24] J. Eshaghi, H. Adibi, and S. Kazem, “Solution of nonlinear weakly singular Volterra integral equations using the fractional-order Legendre functions and pseudospectral method,” *Mathematical Methods in the Applied Sciences*, vol. 39, no. 12, pp. 3411–3425, 2016.

[25] S. Micula, “An iterative numerical method for fractional integral equations of the second kind,” *Journal of Computational and Applied Mathematics*, vol. 339, pp. 124–133, 2018.