Vandermonde-Interpolation Method with Chebyshev Nodes for Solving Volterra Integral Equations of the Second Kind with Weakly Singular Kernels

E. S. Shoukralla, B. M. Ahmed, Ahmed Saeed, and M. Sayed

***Abstract*— in this work, we present an advanced interpolation method via the Vandermonde matrix for solving weakly singular Volterra integral equations of the second kind. The optimal rules for the node distributions of the two kernel variables were created to guarantee that the kernel's singularity was isolated. The unknown function is interpolated using three matrices: one of which is the monomial matrix, based on the Vandermonde matrix and Chebyshev nodes; the second is the known square Vandermyde matrix, and the third is the unknown coefficient matrix. The singular kernel is interpolated twice and transformed into a double-interpolated non-singular function through five matrices, two of which are monomials. A linear algebraic system can be obtained without using the collocation points by inserting the interpolated unknown function on the left and right sides of the integral equation. The solution of the obtained system yields the unknown coefficients matrix and thereby finds the interpolated solution. The obtained results from solving six examples are faster to converge to the exact ones using the lowest degree of interpolants and are better than those achieved by the other indicated method, which confirms the novelty and efficiency of the presented method.**

***Index Terms*— Singular integral equation; barycentric interpolation; weakly singular kernels; computational methods; Chebyshev nodes; Vandermonde matrix; scattering; radiation; image processing; genetic engineering.**

I. Introduction

Singular integral equations appear in many scientific applications in the fields of scattering theory, potential theory, radiation theory, radar, and the effects of magnetic fields on viruses, artificial intelligence, genetic engineering, nanotechnology, thermodynamics, virology, and epidemiology, among others. The Dirichlet or Niemann conditions are commonly imposed when solving initial, boundary, or mixed value problems of Laplace equations, energy equations, electromagnetic wave equations, Helmholtz equations, and other problems using the integral equation method, resulting in equivalent singular boundary integral equations [1-7]. The unknown function's singularity near the integration domain, the kernel's singularity when one of its variables approaches the other variable, or both singularities induce the singularity of integral equations. Shoukralla et al. [8-13] proposed several methods for solving the weakly singular Fredholm integral equations of the first kind, as well as techniques for dealing with the singularities of the unknown functions and kernels. There were used a variety of strategies with orthogonal functions, including monic and economized monic Chebyshev polynomials, as well as other special functions.

However, because of the different features of the Volterra equation's weakly singular kernel, as well as the fact that the endpoint of the limit of integration is a variable rather than a constant, it is difficult to apply these techniques to the Volterra equation. The proposed method in this work employs Lagrange interpolation via the Vandermonde matrix, together with an analytical analysis of the kernel singularity. We established and used one criterion to regulate the strategy of selecting the variable distribution nodes in such a way that no negative or zero values emerge under the square root sign; the kernel singularity was entirely isolated. Many methods for computing the numerical solutions to weakly singular Volterra integral equations of the second kind have recently been developed [14–22]. Zhao et al. [14] created super implicit multistep collocation methods for solving weakly singular Volterra integral equations. Karimi Vanani et al. [15] used the Tau approach to solve weakly singular Volterra integral equations and Abel's integral equations numerically. Kapil Kant et al. [16] used projection methods to attain the convergence rates for weakly singular Volterra integral equations. They used Galerkin and multi-Galerkin methods to solve weakly singular Volterra integral equations. Hou et al. [17] utilized a fractional Jacobi-collocation spectral method to solve the Volterra integral equations of the second kind with a weakly singular kernel. Boykov et al. [18] solved weakly singular Volterra integral equations of various types and determined the orders of Babenko and Kolmogorov n-widths of compact sets from certain classes of functions. Yang [19] proposed pseudo-spectral Jacobi Galerkin for solving weakly singular Volterra integral equation. Zhang et al. [20] rewrote the equation as a new Volterra integral equation with pantograph delays and offered a convergence analysis for the second-kind Volterra integral equation with a weakly singular kernel. Araghi et al.

Manuscript received November 10, 2021; revised July 13, 2022.
E. S. Shoukralla is a Professor of Engineering Mathematics, Faculty of Electronic Engineering, Menofia University, Egypt. (e-mail: shoukralla@el-eng.menofia.edu.eg).
B. M. Ahmed is a Lecturer Assistant of Engineering Mathematics, Faculty of Engineering and Technology, Future University in Egypt, Cairo, Egypt. (corresponding author, phone: +201152157861; e-mail: Basma.magdy@fue.edu.eg).
Ahmed Saeed is an Assistant Professor in the Electrical Engineering Department, Future University in Egypt, Cairo, Egypt. (e-mail: asaeed@fue.edu.eg).
M. Sayed is a Professor of Engineering Mathematics and Head of Department of Physics and Engineering Mathematics, Faculty of Electronic Engineering, Menofia University, Egypt. (e-mail: Mohamed.abdelkader@el-eng.menofia.edu.eg).

[21] utilized Navot-Simpson's Quadrature for solving the singular Abel integral equation of the second kind. Convergence properties for splines and iterated collocation for weakly singular Volterra integral equations relevant to heat conduction problems were introduced by Diogo [22]. Using the least-squares method, Xu et al. [23] presented a novel method for solving all sorts and classes of integral equations. They looked at the stability and uniform convergence in great detail. The hp-form of the discontinuous Galerkin time-stepping approach for weakly singular Volterra integral equations of the second kind was developed by Wang et al. [24]. In [25], Talaei proposed a new method for solving weakly singular Volterra integral equations using an operational matrix based on Chelyshkov polynomials; the integral equation is transformed into a system of algebraic equations. Hashmi et al. [26] obtained the approximate solutions for weakly singular Volterra integral equations using the optimal homotopy asymptotic method (OHAM). For the first time, Shoukralla et al. [27–32] used matrices to upgrade the barycentric Lagrange interpolation formula to solve non-singular linear second-kind Volterra equations. Depending on the smoothness of the kernel and the provided data functions, exact solutions are obtained and proven to be highly convergent. These methods may be sufficient for solving singular Volterra equations, but the most crucial question is how to eliminate the kernel singularity when utilizing the proposed method. The main goal of this research is to use an enhanced Chebyshev node in a barycentric Lagrange interpolation formula and the Vandermonde matrix to interpolate both unknown and provided data functions. On the other hand, the kernel is interpolated twice using the best node distributions for the two variables to isolate its singularity. As a result, the functional values matrix, the known Vandermonde matrix, and the monomial matrix of the primary argument are all represented by three matrices for each one-variable function. We improved the method for interpolating the weakly singular kernel described in [13] and established an original procedure for determining the best node distribution for the two sets of nodes corresponding to the two kernel variables.

As a result, it has been ensured that the kernel's denominator never approaches 0 or becomes imaginary for any value of the nodes. Therefore, a double interpolated non-singular kernel is obtained using five matrices: two monomial matrices related with the two kernel variables, two Vandermonde matrices subjected to the two kernel variables, and a square known coefficient matrix.

The required solution is turned into a system of equations by using some matrix abbreviations. The unknown coefficient matrix and, as a result, the unknown function can be found by solving them directly. Section II presents an interpolation method based on an enhanced barycentric interpolation formula via the Vandermonde matrix with Chebyshev nodes for solving weakly singular Volterra integral equations of the second kind. Six examples were presented and solved in section III, one of which being a non-singular equation. For $n=5$, the interpolated solutions are identical to the precise ones and superior to the solution presented in [33]. The remaining four examples are solved for various upper integration limit values and lowest interpolation degrees. As shown in the tables and figures, the obtained results, including absolute errors, significantly converge to the exact solutions and are superior to those presented in [26]. This emphasizes the proposed method's uniqueness and its ability to produce accurate results with minimal interpolation degree.

II. MATRIX-VECTOR VANDERMONDE INTERPOLATION METHOD

Consider weakly singular Volterra integral equations of the second kind

$$\vartheta(x)=\upsilon(x)+\int_a^x k(x,t)\,\vartheta(t)\,dt\,;x\in I=[a,b] \tag{1}$$

where $\vartheta(x)$ is the unknown solution, $\upsilon(x)$ is a given function, and $k(x,t)=(x-t)^{-\alpha}$ is the given kernel defined on the domain $\Gamma=\{(x,t): a\le x\le t\le b\}$ with b is a real positive number and $0<\alpha<1$. Forming $\vartheta(x)$ in the tabulated form function $\vartheta(x_i)=\vartheta_i$ for $\{x_i\}_{i=0}^n\subset[a,b]$ is the Chebyshev nodes defined by

$$x_i=\frac{b}{2}\left(\xi_j+1\right);\xi_j=\cos\left(\pi\frac{2(n-j)+1}{2(n+1)}\right);\,j=\overline{0,n} \tag{2}$$

Let $\vartheta_n(x)$ be the interpolated polynomial that interpolates $\vartheta(x)$ at $\{x_i\}_{i=0}^n$ under the interpolation condition $\vartheta_n(x_i)=\vartheta(x_i)=\vartheta_i$ for each $i=\overline{0,n}$. Then, it can be obtained in the matrix form

$$\vartheta_n(x)=\mathrm{AX}(x) \tag{3}$$

Here, $\mathrm{A}=\left[a_j\right]_{j=0}^n$ is the $1\times(n+1)$ unknown coefficients matrix and $\mathrm{X}(x)=\left[x^j\right]_{j=0}^n$ is the $(n+1)\times 1$ monomial basis matrix. The unknown coefficients $\{a_j\}_{j=0}^n$ are determined by solving the system

$$\mathrm{VA}=\mathrm{U} \tag{4}$$

where $\mathrm{U}=\left[u(x_i)\right]_{i=0}^n$ is a column matrix, whereas the Vandermonde square matrix V is given by

$$\mathrm{V}=\left[x_{ij}\right]_{i,j=0}^n;x_{ij}=x_i^j,\;x_{ij}=1\;\text{for}\;\;j=0 \tag{5}$$

The solution of the algebraic system (4) yields A, and hence, it results in

$$\vartheta_n(x)=\mathrm{U}^T\left(\mathrm{V}^{-1}\right)^T\mathrm{X}(x) \tag{6}$$

Similarly, the matrix-vector single interpolant $\upsilon_n(x)$ can be obtained by interpolating the given data function $\upsilon(x)$ to get

$$\upsilon_n(x)=\mathrm{F}^T\left(\mathrm{V}^{-1}\right)^T\mathrm{X}(x) \tag{7}$$

where $F^T=\left[\upsilon(x_i)\right]_{i=0}^n$ is the data functional values row matrix. The kernel $k(x,t)$ will be interpolated twice, one for each variable x and one for each variable t. To interpolate the kernel, its denominator singularities must be first isolated, ensuring that the denominator never becomes zero or imaginary. The implementation of this strategy led to design a rule that allows overcoming any kernel singularity when $x\to t$ and $x\to 0$.

Let $c=(b-a)/2$, the two sets of nodes $\{\tilde{x}_i\}_{i=0}^n$ and $\{\tilde{t}_i\}_{i=0}^n$ are then chosen so that they are associated with the two variables x and t as follows

$$\begin{aligned}&\tilde{x}_i=(c+\delta)+ih\,;\;\tilde{t}_j=(a+\delta)+jh\,;\\&h=(c-2\delta)/(n-1)\,;\;\delta=\frac{b}{\omega n};\omega\ge 0\end{aligned}\tag{8}$$

Analogous to (7), the matrix-vector single interpolate kernel $k_n(x,t)$ of degree n is obtained, which corresponds to the nodes $\{\tilde{x}_i\}_{i=0}^n$ in the form

$$k_n(x,t)=X^T(x)V^{-1}K(\tilde{x}_i,t).\tag{9}$$

where $K(\tilde{x}_i,t)$ is a column matrix defined by

$$K(\tilde{x}_i,t)=\left[k(\tilde{x}_i,t)\right]_{i=0}^n\tag{10}$$

Similarly, according to the nodes $\{\tilde{t}_i\}_{i=0}^n$, each entry in the set $\{k(\tilde{x}_i,t)\}_{i=0}^n$ will be interpolated appropriately to the variable t. As a result, the matrix-vector double interpolate kernel $k_{n,n}(x,t)$ is obtained

$$k_{n,n}(x,t)=X^T(x)V^{-1}K\left(\tilde{V}^{-1}\right)^T X(t)\tag{11}$$

where $\tilde{V}$ is the Vandermonde matrix associated with the variable t such that

$$\tilde{V}=\left[t_{ij}\right]_{i,j=0}^n;t_{ij}=t_i^j,\ t_{ij}=1\ \text{for}\ j=0\tag{12}$$

From (6) and (11), the following equation is obtained

$$k_{n,n}(x,t)\vartheta_n(t)=X^T(x)V^{-1}K\left(\tilde{V}^{-1}\right)^T T(t)V^{-1}U\tag{13}$$

where $K=\left[k_{ij}\right]_{i,j=0}^n$ is a square known matrix such that $k_{ij}=k(\tilde{x}_i,\tilde{t}_j)\ \forall\, i,j=\overline{0:n}$, and $T(t)=X(t)X^T(t)$ or $T(t)=\left[t^{i+j}\right]_{i,j=0}^n$. Furthermore, $\vartheta_n(x)$ is determined using (13) and replacing the $\vartheta_n(x)$ supplied by (6) with $\vartheta(t)$ on the right-hand side of (1). Thus,

$$\vartheta_n(x)=\upsilon(x)+X^T(x)V^{-1}K\left(\tilde{V}^{-1}\right)^T\tilde{T}(x)V^{-1}U\tag{14}$$

where $\tilde{T}(x)=\int_0^x T(t)\,dt$.

The following section presents the investigation of a novel method for turning the solution of the integral equation (1) into a linear algebraic system of equations that does not require the use of the collocation method. The implementation of this concept begins by replacing $\vartheta(x)$ in the left side of (1) with $\vartheta_n(x)$ that was given by (14), replacing the kernel $k(x,t)$ in equation (1) with $k_{n,n}(x,t)$ that was given by (11), replacing the $\vartheta(t)$ in the right side with $\vartheta_n(x)$ given by (14), and finally replacing the given data function $\upsilon(t)$ on the right side of (1) with $\upsilon_n(t)$. Hence,

$$\begin{aligned}&X^T(x)V^{-1}K\left(\tilde{V}^{-1}\right)^T\tilde{T}(x)V^{-1}A\\&-\int_0^x X^T(x)V^{-1}K\left(\tilde{V}^{-1}\right)^T X(t)\left[X^T(t)V^{-1}K\left(\tilde{V}^{-1}\right)^T\tilde{T}(t)V^{-1}U\right]dt\\&=\int_0^x X^T(x)V^{-1}K\left(\tilde{V}^{-1}\right)^T X(t)\left[X^T(t)V^{-1}F\right]dt\end{aligned}\tag{15}$$

Simplifying (15), gives

$$\begin{aligned}&X^T(x)V^{-1}K\left(\tilde{V}^{-1}\right)^T\tilde{T}(x)V^{-1}U\\&-X^T(x)V^{-1}K\left(\tilde{V}^{-1}\right)^T\Psi(x)V^{-1}U\\&=X^T(x)V^{-1}K\left(\tilde{V}^{-1}\right)^T\tilde{T}(x)V^{-1}F\end{aligned}\tag{16}$$

where

$$\begin{aligned}&\Psi(x)=\int_0^x\left(T(t)V^{-1}K\left(\tilde{V}^{-1}\right)^T\tilde{T}(t)\right)dt\,,\\&\tilde{T}(x)=\int_0^x T(t)\,dt\end{aligned}\tag{17}$$

By using some matrix algebra, the system (16) is obtained in the simplified form

$$\left(\tilde{T}(x)V^{-1}-\Psi(x)V^{-1}\right)U=\tilde{T}(x)V^{-1}F\tag{18}$$

The coefficient unknown matrix U is obtained from the direct solution of (18), which is then substituted into (6) to provide the interpolated solution $\vartheta_n(x)$

$$\vartheta_n(x)=X^T(x)\left(\tilde{P}(x)-\Psi(x)\right)^{-1}\tilde{T}(x)V^{-1}F\tag{19}$$

III. Computational Results

Six examples were solved using the proposed method in section II, one of which was a non-singular Volterra equation of the second sort, while the others were weakly singular equations. For computing the interpolated numerical solutions, we used the MATLAB software package. For instances 2, 3, 4, 5 and 6, the interpolated numerical solutions to the four singular equations are indicated by $\vartheta_n^b(x)$, and they are determined for $b=0.1, 0.3$, and $n=2,4$. The absolute errors are represented by $E_n^b(x_i)=\left|\vartheta(x_i)-\vartheta_n^b(x_i)\right|$, where

$\vartheta(x_i)$ denote the exact solution values while $\vartheta_n^b(x_i)$ denote the interpolated numerical solutions of degree n at the integration interval $\left[\frac{b}{10}, b\right]$. The obtained interpolated numerical solutions of degree $n=2,4$ significantly converge to the exact solutions.

Example 1

$$\vartheta(x)=x+\frac{7}{12}x^5-\int_0^x(xt^2+x^2t)\vartheta(t)dt$$

The exact solution is $\vartheta(x)=x$ [33]. Table 1 shows the exact solution values $\vartheta(x_i)$, the interpolated numerical solutions $\vartheta_5(x)$, and the absolute errors $\mathrm{E}_5(x_i)$ for any value of b. The obtained solution $\vartheta_5(x)$ is found to be the exact solution. The CPU time for this case is 8.11287 seconds.

Table 1. The exact solution values $\vartheta(x_i)$, the interpolated numerical values $\vartheta_5(x_i)$ and the absolute errors $\mathrm{E}_5(x_i)$

x_i	$\vartheta(x_i)$	$\vartheta_5(x_i)$	$\mathrm{E}_5(x_i)$
0.1	0.1000	0.1000	0
0.2	0.2000	0.2000	0
0.3	0.3000	0.3000	0
0.4	0.4000	0.4000	0
0.5	0.5000	0.5000	0
0.6	0.6000	0.6000	0
0.7	0.7000	0.7000	0
0.8	0.8000	0.8000	0
0.9	0.9000	0.9000	0
1	1.0000	1.0000	0

Example 2

$$\vartheta(x)=x^2+\frac{16}{15}x^{\frac{5}{2}}-\int_0^x\frac{\vartheta(t)}{\sqrt{x-t}}dt \qquad x\in[0,1]$$

The exact solution is $\vartheta(x)=x^2$ [15]. Tables 2 and 3 show the exact solutions $\vartheta(x_i)$ and the numerical solutions $\vartheta_n^b(x_i)$ for $n=2,4$, and $b=0.1,0.3$. In tables 4 and 5, the absolute errors $\mathrm{E}_n^b(x_i)$ for $n=2,4$, and $b=0.1,0.3$ are shown. The graphical representation of the absolute error $\mathrm{E}_n^b(x_i)$ for $n=2,4$, and $b=0.1,0.3$ are shown in Figures 1 and 2, respectively. The CPU time for $b=0.1$ are 2.97405 and 12.126859 seconds, and for $b=0.3$ are 3.17492 and 13.94830, respectively.

Table 2. The exact solution $\vartheta(x_i)$, the interpolated solution $\vartheta_n^{0.1}(x_i)$ for $n=2,4$.

x_i	$\vartheta(x_i)$	$\tilde{\vartheta}_2^{0.1}(x_i)$	$\tilde{\vartheta}_4^{0.1}(x_i)$
0.01	0.0001	0.0000	0.0001
0.02	0.0004	0.0004	0.0004
0.03	0.0009	0.0009	0.0010
0.04	0.0016	0.0017	0.0017
0.05	0.0025	0.0027	0.0026
0.06	0.0036	0.0040	0.0038
0.07	0.0049	0.0054	0.0053
0.08	0.0064	0.0072	0.0070
0.09	0.0081	0.0091	0.0090
0.1	0.0100	0.0112	0.0103

Table 3. The exact solutions $\vartheta(x_i)$, the interpolated solutions $\vartheta_n^{0.3}(x_i)$ for $n=2,4$.

x_i	$\vartheta(x_i)$	$\tilde{\vartheta}_2^{0.3}(x_i)$	$\tilde{\vartheta}_4^{0.3}(x_i)$
0.03	0.0009	0.0001	0.0010
0.06	0.0036	0.0031	0.0041
0.09	0.0081	0.0088	0.0092
0.12	0.0144	0.0165	0.0160
0.15	0.0225	0.0262	0.0244
0.18	0.0324	0.0377	0.0354
0.21	0.0441	0.0517	0.0500
0.24	0.0576	0.0686	0.0658
0.27	0.0729	0.0878	0.0851
0.30	0.0900	0.1063	0.0925

Table 4. The absolute error $\mathrm{E}_n^{0.1}(x_i)$ for $n=2,4$.

x_i	$\mathrm{E}_2^{0.1}(x_i)$	$\mathrm{E}_4^{0.1}(x_i)$
0.01	5.49E-05	5.82 E -06
0.02	3.51E-05	3.28 E -05
0.03	3.78E-05	7.46 E -05
0.04	1.38 E-04	1.16 E -04
0.05	2.46 E-04	1.374 E -04
0.06	3.68 E-04	2.18 E -04
0.07	.32 E-04	4.15 E -04
0.08	7.68 E-04	5.81 E -04
0.09	1.05 E-03	8.58 E -04
0.1	1.17 E-03	2.79 E -04

Table 5. The absolute error $\mathrm{E}_n^{0.3}(x_i)$ for $n=2,4$.

x_i	$\mathrm{E}_2^{0.3}(x_i)$	$\mathrm{E}_4^{0.3}(x_i)$
0.03	8.27 E-04	9.02E-05
0.06	4.82 E-04	4.92 E-04
0.09	6.51E-04	1.09 E-03
0.12	2.13 E-03	1.65 E-03
0.15	3.66E-03	1.89E-03
0.18	5.33 E-03	3.03 E-03
0.21	7.61E-03	5.93 E-03
0.24	1.09 E-02	8.22E-03
0.27	1.49 E-02	1.22 E-02
0.30	1.63E-02	2.54E-03

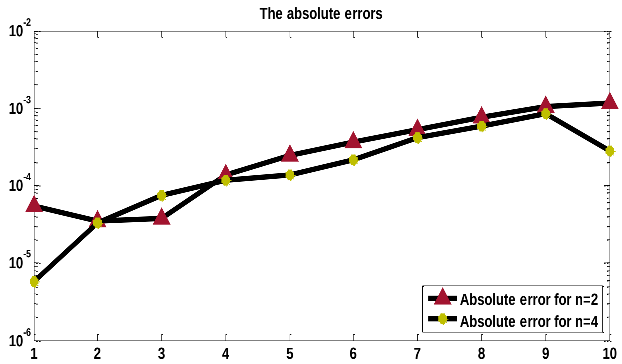

Fig.1. The absolute errors $\mathrm{E}_n^{0.1}(x_i)$ for $n=2,4$

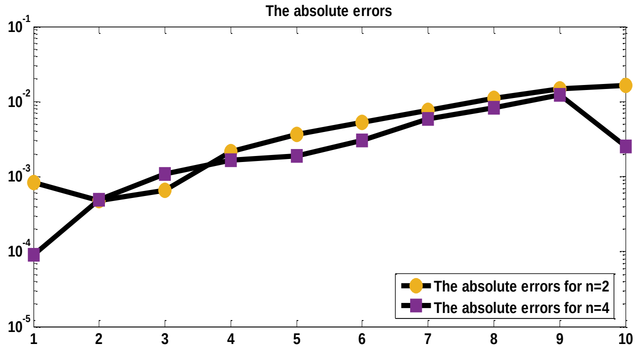

Fig.2. The absolute errors $E_n^{0.3}(x_i)$ for $n=2,4$

Example 3

$$\vartheta(x)=1-\frac{1}{4}\pi x+\frac{1}{2}\int_0^x \frac{\vartheta(t)}{\sqrt{x-t}}dt \ ;x\in[0,1]$$

The exact solution is $\vartheta(x)=1+\sqrt{x}$ [16]. In Tables 6 and 7, the exact solutions $\vartheta(x_i)$, the interpolated numerical solutions $\vartheta_n^b(x_i)$ for $n=2,4$, and $b=0.1,0.3$ are shown. Tables 8 and 9 show the absolute errors $E_n^b(x_i)$ for $n=2,4$, and $b=0.1,0.3$ are shown. In figures 3 and 4, the representation of the absolute error $E_n^b(x_i)$ for $n=2,4$, and $b=0.1,0.3$ are shown. The CPU time for $b=0.1$ are 3.74901 and 11.83681 seconds and for $b=0.3$ are 3.97130 and 14.87301, respectively.

Table 6. The exact solution $\vartheta(x_i)$, the interpolated solution $\vartheta_n^{0.1}(x_i)$ for $n=2,4$.

x_i	$\vartheta(x_i)$	$\tilde{\vartheta}_2^{0.1}(x_i)$	$\tilde{\vartheta}_4^{0.1}(x_i)$
0.01	1.1000	1.0558	1.0827
0.02	1.1414	1.1064	1.1387
0.03	1.1732	1.1520	1.1771
0.04	1.2000	1.1925	1.2050
0.05	1.2236	1.2278	1.2278
0.06	1.2449	1.2582	1.2492
0.07	1.2646	1.2835	1.2709
0.08	1.2828	1.3038	1.2929
0.09	1.3000	1.3191	1.3134
0.1	1.3162	1.3291	1.3288

Table 7. The exact solution $\vartheta(x_i)$, the interpolated solution $\vartheta_n^{0.3}(x_i)$ for $n=2,4$.

x_i	$\vartheta(x_i)$	$\tilde{\vartheta}_2^{0.3}(x_i)$	$\tilde{\vartheta}_4^{0.3}(x_i)$
0.03	1.1732	1.0985	1.1449
0.06	1.2449	1.1890	1.2445
0.09	1.3000	1.2712	1.3144
0.12	1.3464	1.3451	1.3668
0.15	1.3873	1.4109	1.4111
0.18	1.4243	1.4689	1.4539
0.21	1.4583	1.5193	1.4981
0.24	1.4899	1.5620	1.5434
0.27	1.5196	1.5969	1.5870
0.30	1.5477	1.6228	1.6214

Table 8. The absolute error $E_n^{0.1}(x_i)$ for $n=2,4$.

x_i	$E_2^{0.1}(x_i)$	$E_4^{0.1}(x_i)$
0.01	4.42E-02	1.73 E-02
0.02	3.50 E-02	2.7 E-03
0.03	2.12 E-02	3.9 E-03
0.04	7.5 E-03	5.0 E-03
0.05	4.2 E-03	4.2 E-03
0.06	1.32 E-02	4.2 E-03
0.07	1.89 E-02	6.3 E-03
0.08	2.10 E-02	1.00 E-02
0.09	1.91 E-02	1.34 E-02
0.1	1.29 E-02	1.26 E-02

Table 9. The absolute error $E_n^{0.3}(x_i)$ for $n=2,4$.

x_i	$E_2^{0.3}(x_i)$	$E_4^{0.3}(x_i)$
0.03	7.47 E-02	2.83 E-02
0.06	5.59 E-02	4.0 E-04
0.09	2.88 E-02	1.44 E-02
0.12	1.3 E-03	2.04 E-02
0.15	2.36 E-02	2.38 E-02
0.18	4.46 E-02	2.96 E-02
0.21	6.10 E-02	3.99 E-02
0.24	7.21 E-02	5.35 E-02
0.27	7.72 E-02	6.74 E-02
0.30	7.51 E-02	7.37 E-02

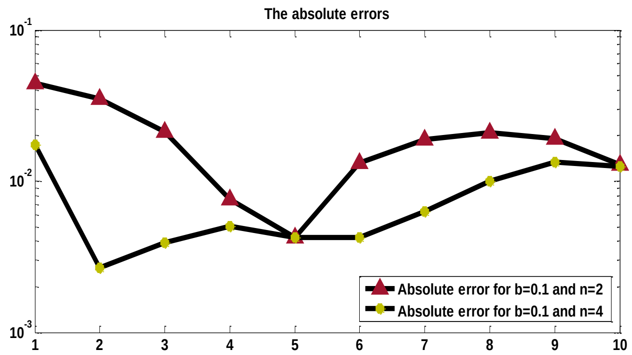

Fig.3. The absolute errors $E_n^{0.1}(x_i)$ for $n=2,4$

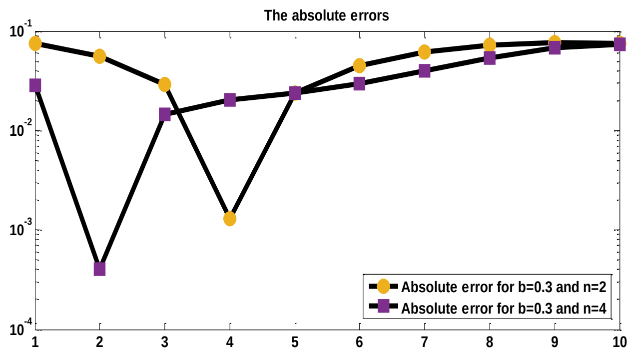

Fig.4. The absolute errors $E_n^{0.3}(x_i)$ for $n=2,4$

Example 4

$$\vartheta(x)=e^x\left(1+\sqrt{\pi}erf(\sqrt{x})\right)-\int_0^x \frac{\vartheta(t)}{\sqrt{x-t}}dt;\ x\in[0,1]$$

The exact solution is $\vartheta(x)=e^x$ [15]. In Tables 10 and 11, the exact solution $\vartheta(x_i)$ the numerical solutions $\vartheta_n^b(x_i)$ for $n=2,4$ and $b=0.1,0.3$ is shown. In tables 12 and 13, the absolute errors $E_n^b(x_i)$ for $n=2,4$ and $b=0.1,0.3$ are shown. In figures 5 and 6, the graphical representation of the absolute error $E_n^b(x_i)$ for $n=2,4$ and $b=0.1,0.3$ is presented. The CPU time for $b=0.1$ are 4.38921 and 14.04929 seconds and for $b=0.3$ are 5.93870 and 15.918041, respectively.

Table 10. The exact solution $\vartheta(x_i)$, the interpolated solution $\vartheta_n^{0.1}(x_i)$ for $n=2,4$.

x_i	$\vartheta(x_i)$	$\tilde{\vartheta}_2^{0.1}(x_i)$	$\tilde{\vartheta}_4^{0.1}(x_i)$
0.01	1.0101	1.0106	1.0105
0.02	1.0202	1.0214	1.0213
0.03	1.0305	1.0325	1.0323
0.04	1.0408	1.0437	1.0436
0.05	1.0513	1.0552	1.0551
0.06	1.0618	1.0669	1.0669
0.07	1.0725	1.0790	1.0790
0.08	1.0833	1.0913	1.0912
0.09	1.0942	1.1039	1.1037
0.1	1.1052	1.1166	1.1162

Table 11. The exact solution $\vartheta(x_i)$, the interpolated solution $\vartheta_n^{0.3}(x_i)$ for $n=2,4$.

x_i	$\vartheta(x_i)$	$\tilde{\vartheta}_2^{0.3}(x_i)$	$\tilde{\vartheta}_4^{0.3}(x_i)$
0.03	1.0305	1.0331	1.0328
0.06	1.0618	1.0681	1.0675
0.09	1.0942	1.1047	1.1041
0.12	1.1275	1.1430	1.1425
0.15	1.1618	1.1832	1.1827
0.18	1.1972	1.2254	1.2251
0.21	1.2337	1.2698	1.2695
0.24	1.2712	1.3165	1.3157
0.27	1.3100	1.3652	1.3640
0.30	1.3499	1.4152	1.4132

Table 12. The absolute error $E_n^{0.1}(x_i)$ for $n=2,4$.

x_i	$E_2^{0.1}(x_i)$	$E_4^{0.1}(x_i)$
0.01	5.0 E-04	4.0 E-04
0.02	1.2 E-03	1.1 E-03
0.03	2.0 E-02	1.9 E-03
0.04	2.9 E-03	2.8 E-03
0.05	3.9 E-03	3.8 E-03
0.06	5.1 E-03	5.1 E-03
0.07	6.5 E-03	6.5 E-03
0.08	8.0 E-03	7.9 E-03
0.09	9.7 E-03	9.5 E-03
0.1	1.14 E-02	1.11 E-02

Table 13. The absolute error $E_n^{0.3}(x_i)$ for $n=2,4$.

x_i	$E_2^{0.3}(x_i)$	$E_4^{0.3}(x_i)$
0.03	2.6 E-03	2.3 E-03
0.06	6.2 E-03	5.7 E-03
0.09	1.06 E-02	9.9 E-03
0.12	1.55 E-02	1.50 E-02
0.15	2.13 E-02	2.09 E-02
0.18	2.82 E-02	2.78 E-02
0.21	3.61 E-02	3.59 E-02
0.24	4.53 E-02	4.45 E-02
0.27	5.53 E-02	5.40 E-02
0.30	6.53 E-02	6.34 E-02

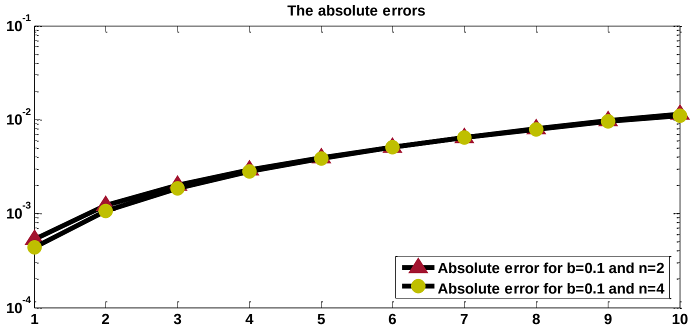

Fig.5. The absolute errors $E_n^{0.1}(x_i)$ for $n=2,4$

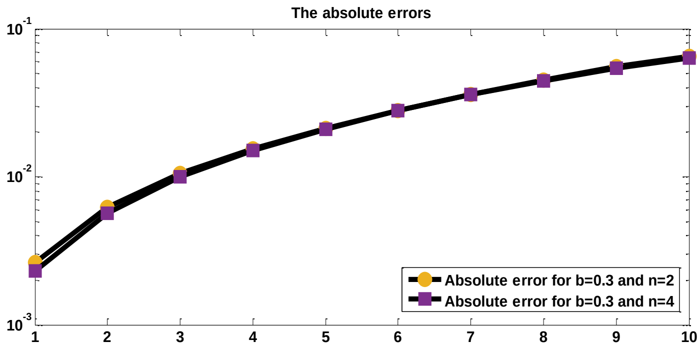

Fig.6. The absolute errors $E_n^{0.3}(x_i)$ for $n=2,4$

Example 5

$$\vartheta(x)=2\sqrt{x}-\int_0^x \frac{\vartheta(t)}{\sqrt{x-t}}dt;\ x\in[0,1]$$

The exact solution is $\vartheta(x)=1-e^{\pi x}erfc\sqrt{\pi x}$ [15]. In tables 14 and 15, the exact solution $\vartheta(x_i)$, the numerical solutions $\vartheta_n^b(x_i)$ for $n=2,4$ and $b=0.1,0.3$ are presented. In tables 16 and 17, the absolute errors $E_n^b(x_i)$ for $n=2,4$ and $b=0.1,0.3$ are shown. In figures 7 and 8, the graphical representation of the absolute error $E_n^b(x_i)$ for $n=2,4$, and $b=0.1,0.3$ are shown. The CPU time for $b=0.1$ are 4.74829 and 12.04592 seconds, respectively, and for $b=0.3$ are 5.93829 and 14.79281, respectively.

Table 14. The exact solution $\vartheta(x_i)$, the interpolated solution $\vartheta_n^{0.1}(x_i)$ for $n=2,4$.

x_i	$\vartheta(x_i)$	$\tilde{\vartheta}_2^{0.1}(x_i)$	$\tilde{\vartheta}_4^{0.1}(x_i)$
0.01	0.1723	0.1062	0.1570
0.02	0.2301	0.1956	0.2494
0.03	0.2702	0.2676	0.3014
0.04	0.3014	0.3225	0.3290
0.05	0.3270	0.3630	0.3441
0.06	0.3489	0.3940	0.3666
0.07	0.3680	0.4201	0.3987
0.08	0.3850	0.4430	0.4222
0.09	0.4002	0.4582	0.4462
0.1	0.4141	0.4518	0.4122

Table 15. The exact solution $\vartheta(x_i)$, the interpolated solution $\vartheta_n^{0.3}(x_i)$ for $n=2,4$.

x_i	$\vartheta(x_i)$	$\tilde{\vartheta}_2^{0.3}(x_i)$	$\tilde{\vartheta}_4^{0.3}(x_i)$
0.03	0.2702	0.1789	0.2630
0.06	0.3489	0.3215	0.4028
0.09	0.4002	0.4276	0.4703
0.12	0.4384	0.4987	0.4954
0.15	0.4687	0.5436	0.5005
0.18	0.4938	0.5753	0.5289
0.21	0.5151	0.6048	0.5797
0.24	0.5336	0.6351	0.6077
0.27	0.5498	0.6544	0.6396
0.30	0.5643	0.6291	0.5393

Table 16. The absolute error $E_n^{0.1}(x_i)$ for $n=2,4$.

x_i	$E_2^{0.1}(x_i)$	$E_4^{0.1}(x_i)$
0.01	6.61 E-02	1.53 E-02
0.02	3.46 E-02	1.93 E-02
0.03	2.6 E-03	3.13 E-02
0.04	2.11 E-02	2.77 E-02
0.05	3.59 E-02	1.71 E-02
0.06	4.50 E-02	1.77 E-02
0.07	5.21 E-02	3.07 E-02
0.08	5.80 E-02	3.72 E-02
0.09	5.79 E-02	4.60 E-02
0.1	3.77 E-02	1.8 E-03

Table 17. The absolute error $E_n^{0.3}(x_i)$ for $n=2,4$.

x_i	$E_2^{0.3}(x_i)$	$E_4^{0.3}(x_i)$
0.03	9.13 E-02	7.2 E-03
0.06	2.75 E-02	5.38 E-02
0.09	2.73 E-02	7.00 E-02
0.12	6.03 E-02	5.70 E-02
0.15	7.49 E-02	3.18 E-02
0.18	8.15 E-02	3.51 E-02
0.21	8.97 E-02	6.46 E-02
0.24	0.1015	7.42 E-02
0.27	0.1046	8.97 E-02
0.30	6.48 E-0.2	2.50 E-02

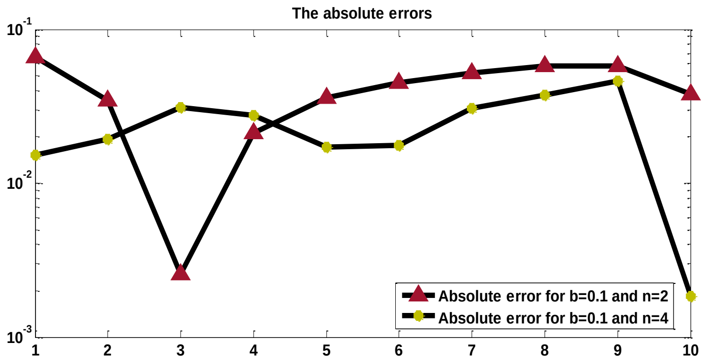

Fig.7. The absolute errors $E_n^{0.1}(x_i)$ for $n=2,4$

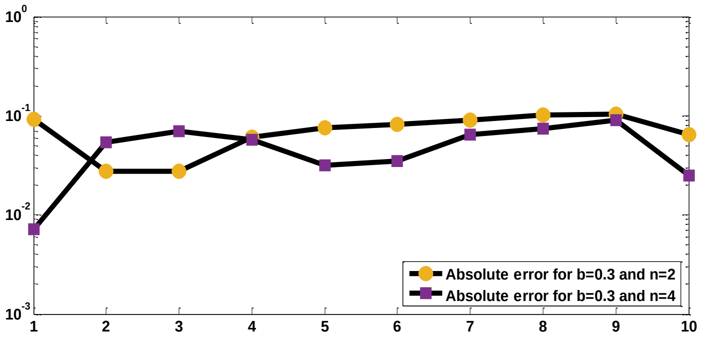

Fig.8. The absolute errors $E_n^{0.3}(x_i)$ for $n=2,4$

Example 6

$$\vartheta(x)=x^7\left(1-\frac{4096}{6435}\sqrt{x}\right)+\int_0^x \frac{\vartheta(t)}{\sqrt{x-t}}dt;\ x\in[0,1]$$

The exact solution is $\vartheta(x)=x^7$ [26]. Tables 18 and 19 show the exact solution $\vartheta(x_i)$ and the numerical solutions $\vartheta_n^b(x_i)$ for $n=2,4$ and $b=0.1, 0.3$. In tables 20 and 21, the absolute errors $E_n^b(x_i)$ for $n=2,4$ and $b=0.1, 0.3$ are presented. In figures 9 and 10, the graphical representation of the absolute error $E_n^b(x_i)$ for $n=2,4$ and $b=0.1, 0.3$. are presented. The CPU time for $b=0.1$ are 5.83760 and 11.84902 seconds respectively and for $b=0.3$ are 6.84083 and 13.48078, respectively. Tables 20 and 21 show the absolute errors $E_n^b(x_i)$ for $b=0.1, 0.3$ and $n=2,4$ respectively.

Table 20 shows that the solutions obtained using the proposed method converge strongly with the exact solutions and that the absolute error $E_2^{0.1}(x_i)=2.081E-07$ is much smaller than the absolute error $1.16E-02$ for $x=0.1$ and $n=2$ of the same example mentioned in [26], indicating that the results obtained based on the proposed solution method are superior to those in [26].

Table 18. The exact solution $\vartheta(x_i)$, the interpolated solution $\vartheta_n^{0.1}(x_i)$ for $n=2,4$.

x_i	$\vartheta(x_i)$	$\tilde{\vartheta}_2^{0.1}(x_i)$	$\tilde{\vartheta}_4^{0.1}(x_i)$
0.01	1.00E-14	7.979E-09	2.174E-09
0.02	1.28E-12	1.251E-08	1.101E-09
0.03	2.19E-11	1.345E-08	-3.978E-10
0.04	1.64E-10	1.053E-08	-8.458E-10
0.05	7.81E-10	3.277E-09	-2.464E-10
0.06	2.80E-09	-8.805E-09	-2.635E-10
0.07	8.24E-09	-2.594E-08	-4.486E-09
0.08	2.10E-08	-4.799E-08	-1.837E-08
0.09	4.78E-08	-7.488E-08	-4.896E-08
0.1	1.00E-07	-1.081E-07	-1.108E-07

Table 19. The exact solution $\vartheta(x_i)$, the interpolated solution $\vartheta_n^{0.3}(x_i)$ for $n=2,4$.

x_i	$\vartheta(x_i)$	$\tilde{\vartheta}_2^{0.3}(x_i)$	$\tilde{\vartheta}_4^{0.3}(x_i)$
0.03	2.187E-11	4.446E-05	1.136E-05
0.06	2.799E-09	7.212E-05	6.634E-06
0.09	4.783E-08	8.1815E-05	-7.263E-07
0.12	3.5832E-07	7.141E-05	-3.411E-06
0.15	1.7091E-06	3.635E-05	-1.437E-06
0.18	6.122E-06	-2.883E-05	-3.709E-06
0.21	1.801E-05	-1.273E-04	-3.025E-05
0.24	4.586E-05	-2.581 E-04	-1.1230 E-04
0.27	1.0460E-04	-4.2054E-04	-2.893 E-04
0.30	2.187 E-04	-6.294 E-04	-6.719 E-04

Table 20. The absolute error $E_n^{0.1}(x_i)$ for $n=2,4$.

x_i	$E_2^{0.1}(x_i)$	$E_4^{0.1}(x_i)$
0.01	7.979E-09	2.174E-09
0.02	1.251E-08	1.100E-09
0.03	1.342E-08	4.197E-10
0.04	1.036E-08	1.009E-09
0.05	2.496E-09	1.028E-09
0.06	1.160E-08	3.063E-09
0.07	3.418E-08	1.272E-08
0.08	6.896E-08	3.934E-08
0.09	1.227E-07	9.679E-08
0.1	2.081E-07	2.108E-07

Table 21. The absolute error $E_n^{0.3}(x_i)$ for $n=2,4$.

x_i	$E_2^{0.3}(x_i)$	$E_4^{0.3}(x_i)$
0.03	4.447E-05	1.136E-05
0.06	7.212E-05	6.632E-06
0.09	8.177E-05	7.742E-07
0.12	7.106E-05	3.769E-06
0.15	3.464E-05	3.145E-06
0.18	3.495E-05	9.831E-06
0.21	1.453E-04	4.826E-05
0.24	3.039 E-04	1.582E-04
0.27	5.251 E-04	3.939 E-04
0.30	8.481 E-04	8.906 E-04

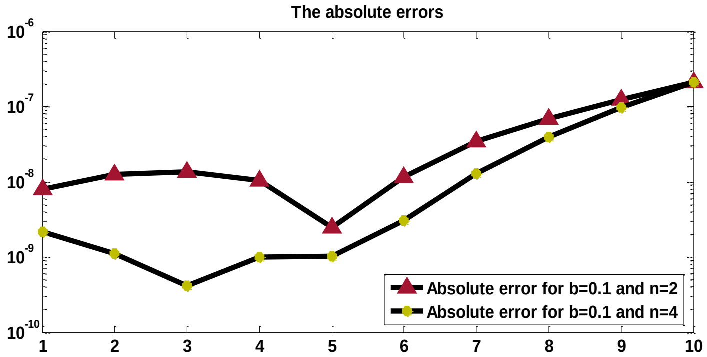

Fig.9. The absolute errors $E_n^{0.1}(x_i)$ for $n=2,4$

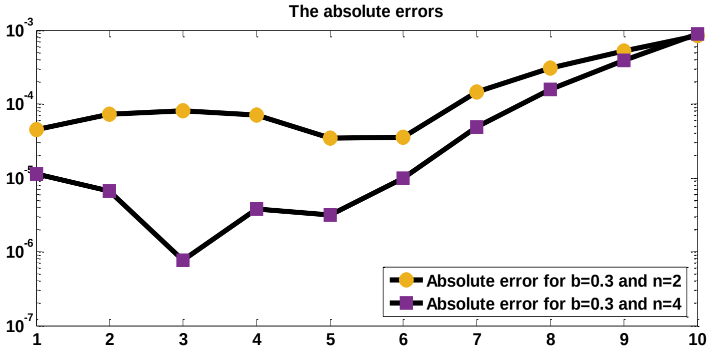

Fig.10. The absolute errors $E_n^{0.3}(x_i)$ for $n=2,4$

Table 22 shows the CPU time comparison for the presented examples for $b=0.1, 0.3$ and $n=2,4$. In example 1 for any value of b, the obtained solution $\vartheta_5(x)$ is found to be the exact solution. The CPU time was 8.11287 seconds for any value of b.

Table 22. Comparison CPU time for $b=0.1, 0.3$ and $n=2,4$ between different examples

Examples	$\upsilon(x)$	$k(x,t)$	CPU time for $b=0.1$ and $n=2,4$ (sec)	CPU time for $b=0.3$ and $n=2,4$ (sec)
1	$x+\frac{7}{12}x^5$	$-xt^2-x^2t$	8.112	8.112
2	$x^2+\frac{16}{15}x^{\frac{5}{2}}$	$\frac{-1}{\sqrt{x-t}}$	2.974 and 12.127	3.175 and 13.948
3	$1-\frac{1}{4}\pi x$	$\frac{1}{2\sqrt{x-t}}$	3.749 and 11.837	3.971 and 14.873
4	$e^x\left(1+\sqrt{\pi}erf(\sqrt{x})\right)$	$\frac{-1}{\sqrt{x-t}}$	4.389 and 14.049	5.939 and 15.918
5	$2\sqrt{x}$	$\frac{-1}{\sqrt{x-t}}$	4.748 and 12.046	5.938 and 14.793
6	$x^7\left(1-\frac{4096}{6435}\sqrt{x}\right)$	$\frac{1}{\sqrt{x-t}}$	5.838 and 11.849	6.841 and 13.481

IV CONCLUSION

An interpolation method based on an enhanced barycentric interpolation formula via the Vandermonde matrix with Chebyshev nodes is presented for solving second-kind weakly singular Volterra integral equations. The method presents a new rule for isolating the kernel singularity, which involves selecting the best node distributions for the two variables to ensure that the numerator does not be imaginary or zero. Without employing the collocation strategy, the required unknown function is converted to an algebraic system by inserting the interpolant solution on both sides of the integral equation. The interpolated solutions of the six solved instances converge faster to the exact ones when the lowest interpolant degree is used, and the results are better than those achieved by the other indicated methods. As a result, the superiority and uniqueness of the proposed method are assured.

REFERENCES

[1] Atkinson, K. E., "The Numerical Solution of Integral Equations of the Second Kind," Cambridge University Press 2010.

[2] Kythe, P. and Puri, P., "Computational Methods for Linear Integral Equations," Birkhäuser, Boston 2002.

[3] Abdul-Majid Wazwaz, "A First Course in Integral Equations-Solutions Manual," 2nd ed., World Scientific Publishing Co. Pte. Ltd 2015.

[4] Kumar, P. and Dubey, G. C., "An Application of Volterra Integral Equation by Expansion of Taylor's Series in the Problem of Heat Transfer and Electrostatics," IOSR Journal of Mathematics (IOSR-JM) 2015, vol.11, no. 5, pp.59-62.

[5] Keaveny, E.E. and Shelley, M.J., "Applying a Second-Kind Boundary Integral Equation for Surface Tractions in Stokes Flow," Journal of Computation Physics 2011, vol. 230, no. 5, pp. 2141-2159.

[6] Hatamzadeh, S. and Naser-Moghadasi, M., "An Integral Equation Modelling of Electromagnetic Scattering from the Surfaces of Arbitrary Resistance Distribution," Progress In Electromagnetics Research B 3 2008, pp. 157–172.

[7] Shoukralla, E. S., "Numerical solution of Helmholtz equation for an open boundary in space," Applied Mathematical Modeling 1997, vol. 21, no. 4, pp. 231-232.

[8] Shoukralla, E., "A Numerical Method for Solving Fredholm Integral Equations of the First Kind with Logarithmic Kernels and Singular Unknown Functions," International Journal of Applied and Computational Mathematics, Springer Nature 2020, vol. 6, no. 6, pp. 1-14.

[9] Shoukralla, E. S., "Application of Chebyshev Polynomials of the Second Kind to the Numerical Solution of Weakly Singular Fredholm Integral Equations of the First Kind," IAENG International Journal of Applied Mathematics 2021, vol. 51, no. 1, pp. 66-81.

[10] Shoukralla, E. S., Kamel, M. and Markos, M. A., "A New Computational Method for Solving Weakly Singular Fredholm Integral Equations of the First Kind," IEEE International Conf. on Computer Engineering and Systems (ICCES 2018) 2018, Cairo, Egypt, pp. 202-207.

[11] Shoukralla E. S, Markos M., "The Economized Monic Chebyshev Polynomials for Solving Weakly Singular Fredholm Integral Equations of the First Kind," Asian-European Journal of Mathematics 2020, vol. 13, no. 1, p. 2050030.

[12] Shoukralla, E.S., "Approximate Solution to Weakly Singular Integral Equations," Applied Mathematical Modelling 1996, vol.20, no. 11, pp. 800-803.

[13] Shoukralla, E. S., "Interpolation Method for Solving Weakly Singular Integral Equations of the Second Kind," Applied and Computational Mathematics 2021, vol. 10, no. 3, pp. 76-85.

[14] Zhao, J., Long, T. and Xu, Y., "Super Implicit Multistep Collocation Methods for Weakly Singular Volterra Integral Equations," Numerical Mathematics: Theory, Methods & Applications 2019, vol. 12, no. 4, pp. 1039-1065.

[15] Vanani, S. K., and Soleymani F., "Tau Approximate Solution of Weakly Singular Volterra Integral Equations," Mathematical and Computer Modelling 2013, vol. 57, no. 3-4, pp. 494-502.

[16] Kant, K., and Nelakanti G., "Approximation Methods for Second Kind Weakly Singular Volterra Integral Equations," Journal of Computational and Applied Mathematics 2020, vol. 368, p. 112531.

[17] Hou, D., Lin, Y., Azaiez, M. and Xu, C., "A Müntz-Collocation Spectral Method for Weakly Singular Volterra Integral Equations," Journal of Scientific Computing 2019, vol. 81, no. 3, pp. 2162-2187.

[18] Boykov, I.V. and Tynda, A.N., "Numerical Methods of Optimal Accuracy for Weakly Singular Volterra Integral Equations," Annals of Functional Analysis 2015, vol. 6, no. 4, pp. 114-133.

[19] Yang, Y., "Jacobi Spectral Galerkin Methods for Volterra Integral Equations with Weakly Singular Kernel," Bulletin of the Korean Mathematical Society 2016, vol. 53, no. 1, pp. 247-262.

[20] Zhang, R., Zhu, B. and Xie, H., "Spectral Methods for Weakly Singular Volterra Integral Equations with Pantograph Delays," Frontiers of Mathematics in China 2013, vol. 8, no. 2, pp. 281-299.

[21] Araghi, M.A.F. and Kasmaei, H.D., "Numerical Solution of the Second Kind Singular Volterra Integral Equations by Modified Navot-Simpson's quadrature," International Journal Open Problems Compt. Math 2008, vol. 1, no. 3, pp. 201-213.

[22] Diogo, T., "Collocation and Iterated Collocation Methods for a Class of Weakly Singular Volterra Integral Equations," Journal of Computational and Applied Mathematics 2009, vol. 229, no.2, pp. 363-372.

[23] Xu, M., Niu, J., Tohidi, E., Hou, J. and Jiang, D., "A New Least-Squares-Based Reproducing Kernel Method for Solving Regular and Weakly Singular Volterra-Fredholm Integral Equations with Smooth and Non-Smooth Solutions," Mathematical Methods in the Applied Sciences 2021, vol. 44, no.13, pp. 10772-10784.

[24] Wang, L., Tian, H. and Yi, L., "An Hp-Version of the Discontinuous Galerkin Time-Stepping Method for Volterra Integral Equations with Weakly Singular Kernels," Applied Numerical Mathematics 2021, vol. 161, pp. 218-232.

[25] Talaei, Y., "Chelyshkov Collocation Approach for Solving Linear Weakly Singular Volterra Integral Equations," Journal of Applied Mathematics and Computing 2019, vol. 60, no.1, pp. 201-222.

[26] Hashmi, M.S., Khan, N. and Iqbal, S., "Numerical Solutions of Weakly Singular Volterra Integral Equations Using the Optimal Homotopy Asymptotic Method," Computers & Mathematics with Applications 2012, vol. 64, no. 6, pp. 1567-1574.

[27] Shoukralla, E. S. and Ahmed, B.M., "The Barycentric Lagrange Interpolation via Maclaurin Polynomials for Solving the Second Kind Volterra Integral Equations," 15th International Conference on Computer Engineering and Systems (ICCES) 2020, December, pp. 1-6. IEEE.

[28] Shoukralla, E. S. and Ahmed, B.M., 2019, "Multi-techniques method for Solving Volterra Integral Equations of the Second Kind," 14th International Conference on Computer Engineering and Systems (ICCES) 2019, December, pp. 209-213. IEEE.

[29] Shoukralla, E. S., Elgohary, H. and Ahmed, B.M., "Barycentric Lagrange Interpolation for Solving Volterra Integral Equations of the Second Kind," Journal of Physics: Conference Series IOP Publishing 2020, vol. 1447, no. 1, p. 012002.

[30] Shoukralla, E.S. and Ahmed, B.M., "Numerical Solutions of Volterra Integral Equations of the Second Kind Using Barycentric Lagrange with Chebyshev Interpolation," Menoufia Journal of Electronic Engineering Research 2019, vol. 28, ICEEM2019-Special Issue, pp. 275-279.

[31] Shoukralla, E. S. and Magdy, B., "Barycentric-Maclaurin interpolation method for solving Volterra integral equations of the second kind," Menoufia Journal of Electronic Engineering Research 2020, vol. 29, no. 1, pp. 84-90.

[32] Shoukralla, E. S. and Ahmed, B.M., "Numerical Solutions of Volterra Integral Equations of the Second Kind Using Lagrange Interpolation Via the Vandermonde Matrix," Journal of Physics: Conference Series IOP Publishing 2020, vol. 1447, no. 1, p. 012003.

[33] Masouri, Z., "Numerical Expansion-Iterative Method for Solving Second Kind Volterra and Fredholm Integral Equations Using Block-Pulse Functions," Advanced Computational Techniques in Electromagnetics 2012, vol. 20, pp. 7-17.