Applied and Computational Mathematics
2021; 10(3): 76-85
http://www.sciencepublishinggroup.com/j/acm
doi: 10.11648/j.acm.20211003.14
ISSN: 2328-5605 (Print); ISSN: 2328-5613 (Online)

SciencePG
Science Publishing Group

Interpolation Method for Solving Weakly Singular Integral Equations of the Second Kind

Emil Sobhy Shoukralla

Department of Engineering Mathematics and Physics, Faculty of Electronic Engineering, Menoufia University, Shibin El Kom, Egypt

Email address:
Shoukralla@el-eng.menofia.edu.eg

To cite this article:
Emil Sobhy Shoukralla. Interpolation Method for Solving Weakly Singular Integral Equations of the Second Kind. *Applied and Computational Mathematics*. Vol. 10, No. 3, 2021, pp. 76-85. doi: 10.11648/j.acm.20211003.14

Received: June 15, 2021; **Accepted**: June 24, 2021; **Published**: June 30, 2021

Abstract: We establish a new straightforward interpolation method for solving linear Volterra integral equations with weakly singular kernels. The proposed method is fundamentally different from all other published methods for solving this type of equations. We have modified some vector-matrix barycentric Lagrange interpolation formulas to be convenient for interpolating the kernel twice concerning the two variables of the kernel and introducing new ideas for selecting interpolation nodes that ensure isolation of the singularity of the kernel. We create two rules for selecting the distribution nodes of the two kernel variables that do not allow the denominator of the kernel to contain an imaginary value. We interpolate the unknown and data functions into the corresponding interpolant polynomial; each of the same degree via three matrices, one of which is a monomial. By applying the presented method based on the two created rules, we transformed the kernel into a double interpolant polynomial with a degree equal to that of the unknown function via five matrices, two of which are monomials. We substitute the interpolate unknown function twice; on the left side and on the right side of the integral equation to get an algebraic linear system without applying the collocation method. The solution of this system yields the unknown coefficients matrix that is necessary to find the interpolant solution. We solve three different examples for different values of the upper integration variable. The obtained results as shown in tables and figures prove that the obtained interpolate solutions are extraordinarily faster to converge to the exact ones using interpolants of lowest degrees and give better results than those obtained by other methods. This confirms the originality and the potential of the presented method.

Keywords: Lagrange Interpolation, Singular Integral, Weakly Singular Volterra Kernels, Computational Methods, Vandermonde Matrix, Scattering, Radiation, Image Processing

1. Introduction

In this study, we present an interpolation method to find the interpolate solution of the linear second kind weakly singular Volterra integral equation of the second kind. The method is based on the application of Lagrange interpolation through the Vandermonde matrix with the analytical treatment of the kernel singularity. The complete isolation of the kernel singularity was accomplished through two rules that we created to control the strategy of selecting the variables distribution nodes in a way that ensures that there are no negative or zero quantities under the square root sign.

First, we emphases that the successive emergence of the application of the integral equation method in many scientific fields, and more recently in the sciences of artificial intelligence, genetic engineering, nanotechnology, thermodynamics, virology, and epidemiology prompted researchers to search for methods adapting to the mathematical properties of each integral equation and, in particular, to the behavior of the kernel at the boundary points of the integration [1-3]. One of the most important advantages of the integral equation method is that it can transform the solution of initial, boundary, and mixed value problems for a partial differential equation in two independent variables, into one boundary integral equation involving an unknown function of only one variable. We need to find solutions of the integral equations that fit the imposed conditions on each value problem which satisfies the mathematical properties of the data function, the kernel, and the unknown functions that form the equivalent integral equation. In this study, we focus on solutions of weakly singular Volterra integral equations of the second kind.

$$u(x)=f(x)+\int_a^x k(x,t)u(t)\,dt\ ;x\in I:=[a,b] \qquad (1)$$

where $f(x)$ is given real continuous function on the domain I ,$k(x,t)$ is given a continuous kernel on the domain $\Omega=\{(x,t):a\le x\le t\le b\}$ where b is a real positive number. For the weakly singular kernel $k(x,t)$, it is usually given in the form $k(x,t)=\frac{g(x,t)}{(x-t)^{\alpha}}$ where $0<\alpha<1$ with the assumption that $\|k(x,t)\|_2\le M<\infty$ where $x,t\in\Omega$. Here, $u(x)$ is the unknown function to be determined. The Volterra operator which is defined by $\mathrm{T}u=\int_a^x k(x,t)u(t)\,dt$, is acting in $L_2[a,b]$. In this type of equation, we are faced with the problem of how to overcome the kernel singularity near and at the endpoints of the integration domain [4]. For the weakly singular linear Fredholm integral equation of the first kind, Shoukralla et al. [5-10], treated the singularity of the kernel near and at the endpoints of the integration domain and provided numerical solutions to the first-kind Fredholm integral equation which contains a weakly singular logarithmic kernel, using various techniques with applications of orthogonal functions such as monic and economized monic Chebyshev polynomials and other special functions. However, it was difficult to apply these techniques to equation (1) in terms of the different properties of the weakly singular kernel of the Volterra equation, in addition to the fact that the endpoint of the limit of integration is a variable and not a constant like the Fredholm equations. For the non-singular linear Volterra equations of the second kind, Shoukralla et al. [11-13] modified the barycentric Lagrange interpolation formula [14, 15], through matrices, and for the first time, with remarkable success, they solved non-singular linear Volterra equations of the second kind using different techniques and obtained exact numerical solutions or strongly convergent solutions depending on the smoothness of the kernel and the given data functions. Recently, many methods and various techniques have spread to find numerical solutions to weakly singular Volterra integral equations of the second kind [16-20]. Zhang Xiao-Yong [16] introduced a Jacobi spectral method for the solution of Volterra integral equations of the second kind with a weakly singular kernel. The author provided a different technique than the spectral-collocation method based on the advantage of the property of orthogonal polynomials. Can Huang et al. [17] applied the spectral Galerkin polynomial method and a variant with the trial and test space is augmented by suitably chosen functions, besides the Sloan iterations of both methods for solving the second kind weakly singular linear Volterra integral equations. They proved the convergence rates for all these methods. Lina Wang et al [18] presented and realized a certain hp-version of the DG approach for the linear Volterra integral equation of the second kind with weakly singular kernels. They applied the piecewise and discontinuous polynomials to trial and test spaces of the DG method. They developed two improvement tools for the singularity of the solution at and near the initial point of the integration domain. Jingjun Zhao et al. [19], solved the weakly singular Volterra integral equations of the first kind by using smoothing transformations and then applied the super implicit multistep collocation. Hossein et al. [20] applied the multi-projection method and developed a fully discretized version using specific quadrature rules upon using collocation projection techniques. This yields a super convergence solution of Volterra integral equations of the second kind with weakly singular kernels. The main goal of this paper is to apply the Lagrange interpolation formula via the Vandermonde matrix to interpolate both the unknown and the given data functions by interpolating polynomials of the same degree n. Thus, each of these is expressed via three matrices; the first is the functional values, the second is the known Vandermonde matrix, and the third is the monomial basis functions matrix of the main argument. For interpolating the weakly singular kernel we developed an approach based on two rules. First, we divided the integration domain into two sub-domains. On the right half domain, we interpolate the kernel concerning the variable x to obtain interpolant polynomial of degree n and on the left half domain, we interpolate the kernel concerning the second variable t to obtain another interpolating polynomial of the same degree n. We achieve this idea by choosing the two sets of nodes $\{x_i\}_{i=0}^{n}$ and $\{t_j\}_{j=0}^{n}$ so that we ensure that the denominator of the kernel never approaches zero or becomes imaginary for any value of the nodes x_i and t_j . To implement this aim, we define the step-sizes for each $\{x_i\}_{i=0}^{n}$ depending on the some real numbers $\delta_1\ge 0$ and $\{t_j\}_{j=0}^{n}$ depending on some real numbers $\delta_2\ge 0$ depending on some real numbers $\delta_1\ge 0$, such that the value of each $\delta_i\ge 0\ ;i=1,2$ is depending on the value of the parameter b and the degree n of interpolant kernel. These rules are valid for any values of b and also valid for $n\ge 2$. To simplify the procedure and represent the double interpolant kernel through five matrices, we reverse the order of the three matrices of the second single interpolant kernel and multiply the result by the first three matrices of the first single interpolant kernel. Thus, we obtain a double interpolant kernel of degree $n\times n$ through five matrices, two of which are monomial basis functions matrices for the variables x and t respectively, two are the Vandermonde matrices for x and t respectively, and the fifth matrix is a square known coefficients matrix. To transform the solution (unknown function $u(x)$) of (1) into the solution of a linear algebraic system of equations we substitute the single interpolant polynomial in both sides of (1), and by taking into account the substitution of the double interpolate kernel, and the single interpolate data function on the right-hand side of (1).

The direct solution of this system yields the unknown coefficient matrix and thereby we can find the unknown matrix itself. We solved three different examples for different values of b and for $n = 2$. We estimated the limits of the obtained interpolant solutions at $x \to 0$ and estimated the maximum error on each subinterval corresponding to the value of b. The obtained results are compared with the exact solution and tables are constructed including also the absolute and relative errors. From the given tables and figures, it turns out that the obtained interpolate solutions are strongly and faster converging to the exact solution than the solution obtained by using other methods. This proves the originality of the presented new method and its distinction in giving accurate solutions for the lowest possible degree of the interpolant solution.

2. Interpolation Method

Let $u(x) \in C^{n+1}[a,b]$ and $u_n(x)$ be the Lagrange interpolating polynomial of degree n that interpolates $u(x)$ at the $(n+1)$ equally spaced distinct nodes $\{x_i\}_{i=0}^{n} \subset [a,b]$. By choosing a step size $h > 0$ such that $h = \dfrac{b-a}{n}$, we get the required $(n+1)$ equidistant interpolation nodes $x_i = a + ih$; $t_j = a + jh$; $i, j = \overline{0,n}$. The presented method relies directly on the Lagrange interpolation theory and not on the barycentric Lagrange interpolation. We will express the interpolation Lagrange formula through what is called the Vandermonde Matrix, and use this idea to interpolate the unknown and the given functions $u(x)$ and $f(x)$, as we use it to interpolate the kernel twice, which makes us at the end of the solution procedures get an algebraic system of linear equations. Solving this system yields the matrix of unknown coefficients and then we can get the interpolant polynomial which interpolates the unknown function. Let $u_n(x)$ be the Lagrange interpolating polynomial of degree n that interpolates $u(x)$ at the $(n+1)$ equally spaced distinct nodes $\{x_i\}_{i=0}^{n} \subset [a,b]$ such that $u_n(x_i) = u(x_i) = u_i \ \forall i = \overline{0,n}$. By choosing a step size $h > 0$ such that, $h = \dfrac{b-a}{n}$ we can get the $(n+1)$ equidistant interpolation nodes $x_i = a + ih$; $i = \overline{0,n}$. We begin by pouting t $u_n(x)$ in the form

$$u_n(x) = \sum_{j=0}^{n} a_j x^j ; j = \overline{0,n} \tag{2}$$

In matrix form, we get

$$u_n(x) = \mathrm{UP}(x) \tag{3}$$

Here $\mathrm{U}=\left[a_j\right]_{j=0}^{n}$ is the unknown coefficients row matrix of order $1\times(n+1)$, $\mathrm{P}(x)=\left[x^j\right]_{j=0}^{n}$ is the $(n+1)\times 1$ monomial basis functions column matrix. The unknown coefficients $\left\{a_j\right\}_{j=0}^{n}$ are determined by solving the linear algebraic system of equations

$$\mathrm{XU = A} \tag{4}$$

Where A is the column matrix $\mathrm{A}=\left[u(x_i)\right]_{i=0}^{n}$, where $\left\{x_i, u(x_i)\right\}_{i=0}^{n}$; $x_i = a + ih$ are the interpolation knots that interpolate $u(x)$, whereas the Vandermonde square matrix X is given by

$$\mathrm{X}=\left[b_{ij}\right]_{i,j=0}^{n} ; b_{ij} = x_i^j,\ b_{ij} = 1 \ \forall\ i = \overline{0:n}, j = 0 \tag{5}$$

By virtue of (4) we can find $u_n(x)$ in the following matrix-vector single interpolant form

$$u_n(x) = \mathrm{A}^T \tilde{\mathrm{X}} \mathrm{P}(x);\ \tilde{\mathrm{X}} = \left(\mathrm{X}^{-1}\right)^T . \tag{6}$$

similarly, the given data function $f(x)$ can be interpolated like $u(x)$ to get the matrix-vector single interpolant $f_n(x)$ in the matrix form

$$f_n(x) = \mathrm{F}^T \tilde{\mathrm{X}} \mathrm{P}(x) \tag{7}$$

where $\mathrm{F}^T = \left[f(x_i)\right]_{i=0}^{n}$ is the row matrix of the order $1\times(n+1)$. The kernel $k(x,t)$ will be interpolated twice corresponding to x, and. .. First, we divided the integration domain $[a,b]$ to become $\left[a + \dfrac{b-a}{2} - \delta_1\right] \cup \left[\dfrac{b-a}{2} + \delta_2, b\right]$ for some real numbers $\delta_1, \delta_2 \geq 0$ such that $\delta_1 = \delta_1(n)$ and $\delta_2 = \delta_2(n)$, where n is the interpolant degree. To interpolate the kernel while retaining $x > t$, we have to set the rules that ensure this procedure overcomes any singularity of the kernel when $x \to t$ and when $x \to 0$. We interpolate $k(x,t)$ with respect to x on the right-half interval and with respect to t on the left-half interval. Let $\Delta = (b-a)/2$. Then we choose the two sets of nodes $\{\tilde{x}_i\}_{i=0}^{n}$ and $\left\{\tilde{t}_j\right\}_{j=0}^{n}$ as follows

$$\tilde{x}_i = (\Delta + \delta_1) + i \times h1 \ ; h1 = (b - \Delta - 2\delta_1)/n \ ; \ \delta_1 = \frac{b}{\lambda_1 n} ; \lambda_1 \geq 0 \tag{8}$$

and

$$\tilde{t}_j = (a+\delta_2) + j \times h2 \;; h2 = (a + \Delta - 2\delta_2) / n \;;\; \delta_2 = \frac{b}{\lambda_2 n}; \lambda_2 \geq 0 \quad (9)$$

Analogical to (6) and (7), we get the matrix-vector single interpolate kernel $k_n(x,t)$ of degree n that corresponding the set of nodes $\{\tilde{x}_i\}_{i=0}^{n}$ in the form

$$k_n(x,t) = \mathrm{P}^T(x)\mathrm{X}^{-1}\mathrm{K}(\tilde{x}_i,t). \quad (10)$$

Where $\mathrm{K}(\tilde{x}_i,t)$ is the $(n+1)\times 1$ column matrix

$$\mathrm{K}(\tilde{x}_i,t) = \left[k(\tilde{x}_0,t) \quad k(\tilde{x}_1,t) \quad \ldots \quad k(\tilde{x}_n,t)\right]^T \quad (11)$$

Now, each entry of the entries set $\{k(\tilde{x}_i,t)\}_{i=0}^{n}$ will be interpolated corresponding to the variable t according to the set of nodes $\{\tilde{t}_j\}_{j=0}^{n}$ given by (9). Without going into substitutions steps upon using the matrix-algebra, we invert the order of the matrices of the second interpolation and multiplying by the first interpolation from the right side, so we get the matrix-vector double interpolate kernel $k_{n,n}(x,t)$ in the form

$$k_{n,n}(x,t) = \mathrm{P}^T(x)\mathrm{X}^{-1}\mathrm{K}\tilde{\mathrm{T}}^{-1}\mathrm{P}(t) \quad (12)$$

From (6) and (12), we get

$$k_{n,n}(x,t)u_n(t) = \mathrm{P}^T(x)\mathrm{X}^{-1}\mathrm{K}\tilde{\mathrm{T}}^{-1}\tilde{\mathrm{P}}(t)\mathrm{X}^{-1}\mathrm{A} \quad (13)$$

Where $\mathrm{K} = \left[k_{ij}\right]_{i,j=0}^{n}$ is the $(n+1)\times(n+1)$ square known matrix whose entries are determined by $k_{ij} = k(\tilde{x}_i,\tilde{t}_j) \;\forall\; i,j = \overline{0:n} \quad \tilde{\mathrm{P}}(t) = \mathrm{P}(t)\mathrm{P}^T(t) = \left[t^{i+j}\right]_{i,j=0}^{n}$ is the $(n+1)\times(n+1)$ square matrix.

Consequently, by replacing $u_n(x)$ given by (6) with $u(t)$ in the right side of Eq. (1) we find that

$$\begin{aligned} u_n(x) &= f(x) + \int_0^x \mathrm{P}^T(x)\mathrm{X}^{-1}\mathrm{K}\tilde{\mathrm{T}}^{-1}\tilde{\mathrm{P}}(t)\mathrm{X}^{-1}\mathrm{A}dt \;\;; \tilde{\mathrm{T}}^{-1} = \left(\mathrm{T}^{-1}\right)^T \\ &= f(x) + \mathrm{P}^T(x)\mathrm{X}^{-1}\mathrm{K}\tilde{\mathrm{T}}^{-1}\Phi(x)\mathrm{X}^{-1}\mathrm{A} \;;\; \Phi(x) = \int_0^x \tilde{\mathrm{P}}(t)dt \end{aligned} \quad (14)$$

By replacing $u(x)$ with $u_n(x)$ given by (6) into the left side of (1) and by $u_n(t)$ into the right side, besides replacing $k(x,y)$ with $k_{n,n}(x,t)$ given by (13) and $f_n(t)$ given by (7) with $f(t)$ we get

$$\begin{aligned} &\mathrm{P}^T(x)\mathrm{X}^{-1}\mathrm{K}\tilde{\mathrm{T}}^{-1}\Phi(x)\mathrm{X}^{-1}\mathrm{A} - \mathrm{P}^T(x)\mathrm{X}^{-1}\mathrm{K}\tilde{\mathrm{T}}^{-1}\tilde{\Phi}(x)\mathrm{X}^{-1}\mathrm{A} \\ &= \mathrm{P}^T(x)\mathrm{X}^{-1}\mathrm{K}\tilde{\mathrm{T}}^{-1}\Phi(x)\tilde{\mathrm{X}}^T\mathrm{F} \end{aligned} \quad (15)$$

where

$$\tilde{\Phi}(x) = \int_a^x \tilde{\mathrm{P}}(t)\mathrm{X}^{-1}\mathrm{K}\tilde{\mathrm{T}}^{-1}\Phi(t)dt \quad (16)$$

Simplifying (15), we get the unknown coefficient matrix A of the interpolate solution $u_n(x)$ given by (6) by solving the linear algebraic system

$$\left(\Phi(x) - \tilde{\Phi}(x)\right)\mathrm{X}^{-1}\mathrm{A} = \Phi(x)\mathrm{X}^{-1}\mathrm{F} \;\;; \tilde{\mathrm{X}} = \left(\mathrm{X}^{-1}\right)^T \quad (17)$$

Let

$$\mathrm{N}^{-1} = \left(\left(\Phi(x) - \tilde{\Phi}(x)\right)\mathrm{X}^{-1}\right)^{-1} = \mathrm{X}\left(\Phi(x) - \tilde{\Phi}(x)\right)^{-1} \quad (18)$$

Hence, we get

$$\mathrm{A} = \mathrm{X}\left(\Phi(x) - \tilde{\Phi}(x)\right)^{-1}\Phi(x)\mathrm{X}^{-1}\mathrm{F} \quad (19)$$

Substituting the matrix A into (6), we get the required solution in the matrix form

$$u_n(x) = \mathrm{P}^T(x)\mathrm{X}^{-1}\mathrm{A} = \mathrm{P}^T(x)\left(\Phi(x) - \tilde{\Phi}(x)\right)^{-1}\Phi(x)\mathrm{X}^{-1}\mathrm{F} \quad (20)$$

3. Computational Results

We designed a MATLAB code to find the interpolant solution $u_n(x)$ for $n \geq 2$ and for any value of the parameter b . We solve three examples and finding $u_2(x)$ for $b = 0.1, 0.2, 0.4, 0.6$. We have chosen $n = 2$ to avoid the complicated roundoff error due the calculations for $n \geq 3$. Upon using (7) and (8), we set $\lambda_1 = \lambda_2 = 3$ and $b = 0.1, 0.2, 0.4, 0.6$ for the three examples. We have chosen the values of the parameters δ_1, δ_2 depending on b , and n . For the three examples we denote the absolute error by $R_2^b(x_i) = \left|u(x_i) - u_2^b(x_i)\right|$, where $u(x_i)$ denotes the exact solution values and $u_2^b(x_i)$ denotes the interpolate solution values of degree 2 at the sub-integration interval $[b/10, b]$. By $\mathfrak{R}_2^b(\xi)$, where $\mathfrak{R}_2^b(\xi) = \max\limits_{\xi \in [b/10,b]} \left|\mathfrak{R}_2^b(\xi)\right|$, we denote the maximum Lagrange interpolation error of interpolating the exact solution $u(x)$ on each corresponding sub-interval $[b/10, b]$. Then we can estimate $\mathfrak{R}_2^b(\xi)$ using the following inequality $\max\limits_{x \in [b/10,b]} \left|\mathfrak{R}_2^b(x)\right| \leq \dfrac{\max\limits_{x \in [b/10,b]} u^{(n+1)}(x)}{(n+1)!} \left|\prod_{i=0}^{n}(x - \tilde{x}_i)\right|$.

Let $\mathrm{E}_2^b(x_i)$ denote the relative error of interpolation, then we can estimate this error by using the following relative error inequality $\mathrm{E}_2^b(x_i) = |u(x_i) - u_2^b(x_i)| < \frac{1}{2} \times 10^{-5} |u(x_i)|$. The obtained interpolant solutions for $n = 2$ and for any value of b are faster converge to the exact ones than the methods described in [17, 18].

Example 1

Consider the weakly singular Volterra integral equation of the second kind $u(x) = \frac{\pi x}{2} + \sqrt{x} - \int_0^x \frac{u(t)}{\sqrt{x-t}} dt$. The exact solution [19] is given by $u(x) = \sqrt{x}$, and $0 \le x \le b$ where b is a real number. Tables 1, 2, 3, and 4, show the exact solution values $u(x_i)$, the interpolate solution values $u_2^b(x_i)$, the absolute errors $R_2^b(x_i)$, and the relative errors $\mathrm{E}_2^b(x_i)$ at the corresponding set of nodes $x_i = b/10 : b/10 : b$. In figures 1, 2, 3, and 4, plotted are the graphs of the exact solution $u(x)$ and the graphs of the interpolant solutions $u_2^b(x)$ for $b = 0.1, 0.2, 0.4, 0.6$. We find from the exact solution of (1) that $\lim_{x \to 0} u(x) = 0$, while the presented interpolation method gives $\lim_{x\to 0} u_2^{0.1}(x) = 0.10235, \lim_{x\to 0} u_2^{0.2}(x) = 0.14475, \lim_{x\to 0} u_2^{0.4}(x) = 0.20471$, and $\lim_{x\to 0} u_2^{0.6}(x) = 0.25071$. The maximum error on each subinterval $[b/10, b]$ are given by $\max_{\xi[0.01,0.1]} |\mathfrak{R}_n^{0.1}(\xi)| \le 0.086806$, $\max_{\xi[0.02,0.2]} |\mathfrak{R}_n^{0.2}|(\xi) \le 0.12276$, $\max_{\xi[0.04,0.4]} |\mathfrak{R}_n^{0.4}(\xi)| \le 0.17361$, and $\max_{\xi[0.06,0.6]} |\mathfrak{R}_n^{0.6}(\xi)| \le 0.21263$.

Table 1. *The exact solutions* $u(x_i)$*, the interpolate solutions* $u_2^{0.1}(x_i)$*, the absolute error* $R_2^{0.1}(x_i)$*, and the relative error* $\mathrm{E}_2^{0.1}(x_i)$ *at* $x_i = 0.01 : 0.01 : 0.1$.

x_i	$u(x_i)$	$u_2^{0.1}(x_i)$	$R_2^{0.1}(x_i)$	$\mathrm{E}_2^{0.1}(x_i)$
0.01	0.1	0.13553	0.03553	5.0e-7
0.02	0.14142	0.16681	0.02539	7.0711e-7
0.03	0.17321	0.19477	0.02156	8.6603e-7
0.04	0.2	0.2192	0.0192	1.0e-6
0.05	0.22361	0.24114	0.01753	1.118e-6
0.06	0.24495	0.26243	0.01748	1.2247e-6
0.07	0.26458	0.28471	0.02013	1.3229e-6
0.08	0.28284	0.30831	0.02547	1.4142e-6
0.09	0.3	0.3307	0.0307	1.5e-6
0.1	0.31623	0.34469	0.02846	1.5811e-6

Table 2. *The exact solutions* $u(x_i)$*, the interpolate solutions* $u_2^{0.2}(x_i)$*, the absolute error* $R_2^{0.2}(x_i)$*, and the relative error* $\mathrm{E}_2^{0.2}(x_i)$ *at* $x_i = 0.02 : 0.02 : 0.2$.

x_i	$u(x_i)$	$u_2^{0.2}(x_i)$	$R_2^{0.2}(x_i)$	$\mathrm{E}_2^{0.2}(x_i)$
0.02	0.14142	0.1965	0.05377	7.0711e-7
0.04	0.2	0.24463	0.04222	1.0e-6
0.06	0.24495	0.28436	0.03818	1.2247e-6
0.08	0.28284	0.31547	0.03474	1.4142e-6
0.1	0.31623	0.34174	0.03163	1.5811e-6
0.12	0.34641	0.36891	0.03125	1.7321e-6
0.14	0.37417	0.4018	0.03607	1.8708e-6
0.16	0.4	0.44119	0.04613	2.0e-6
0.18	0.42426	0.47954	0.05596	2.1213e-6
0.2	0.44721	0.49555	0.051	2.2361e-6

Table 3. *The exact solutions* $u(x_i)$*, the interpolate solutions* $u_2^{0.4}(x_i)$*, the absolute error* $R_2^{0.4}(x_i)$*, and the relative error* $\mathrm{E}_2^{0.4}(x_i)$ *at* $x_i = 0.04 : 0.02 : 0.4$.

x_i	$u(x_i)$	$u_2^{0.4}(x_i)$	$R_2^{0.4}(x_i)$	$\mathrm{E}_2^{0.4}(x_i)$
0.04	0.2	0.2828	0.0828	1.0e-6
0.08	0.28284	0.3543	0.07146	1.4142e-6
0.12	0.34641	0.41403	0.06762	1.7321e-6
0.16	0.4	0.46173	0.06173	2.0e-6
0.2	0.44721	0.50238	0.05517	2.2361e-6
0.24	0.4899	0.54348	0.05358	2.4495e-6
0.28	0.52915	0.59132	0.06217	2.6458e-6
0.32	0.56569	0.64663	0.08094	2.8284e-6
0.36	0.6	0.69912	0.09912	3.0e-6
0.4	0.63246	0.72058	0.08812	3.1623e-6

Table 4. *The exact solutions $u(x_i)$, the interpolate solutions $u_2^{0.6}(x_i)$, the absolute error $R_2^{0.6}(x_i)$, and the relative error $\mathrm{E}_2^{0.6}(x_i)$ at $x_i = 0.06:0.06:0.6$.*

x_i	$u(x_i)$	$u_2^{0.6}(x_i)$	$R_2^{0.6}(x_i)$	$\mathrm{E}_2^{0.6}(x_i)$
0.06	0.24495	0.35247	0.10752	1.2247e-6
0.12	0.34641	0.44415	0.09774	1.7321e-6
0.18	0.42426	0.51835	0.09409	2.1213e-6
0.24	0.4899	0.57517	0.08527	2.4495e-6
0.3	0.54772	0.62251	0.07479	2.7386e-6
0.36	0.6	0.67167	0.07167	3.0e-6
0.42	0.64807	0.7317	0.08363	3.2404e-6
0.48	0.69282	0.8034	0.11058	3.4641e-6
0.54	0.73485	0.87128	0.13643	3.6742e-6
0.6	0.7746	0.89371	0.11911	3.873e-6

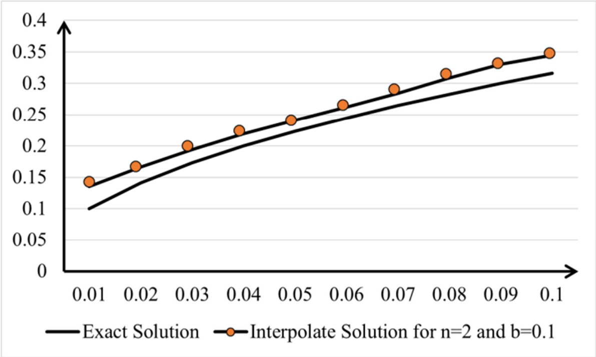

Figure 1. *The exact solution $u(x)$ and the interpolate solution $u_2^{0.1}(x)$ graphs.*

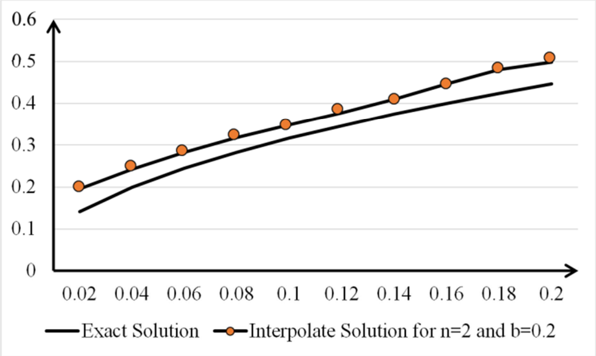

Figure 2. *The exact solution $u(x)$ and the interpolate solutions $u_2^{0.2}(x)$ graphs.*

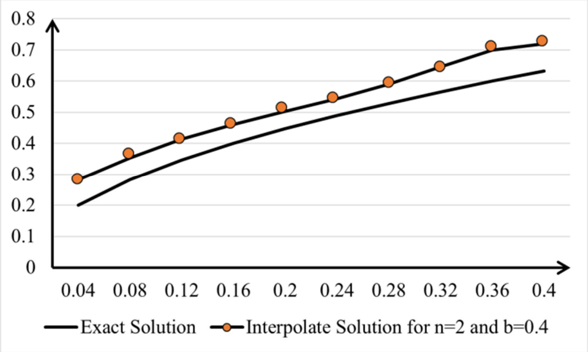

Figure 3. *The exact solution $u(x)$ and the interpolate solution $u_2^{0.4}(x)$ graphs.*

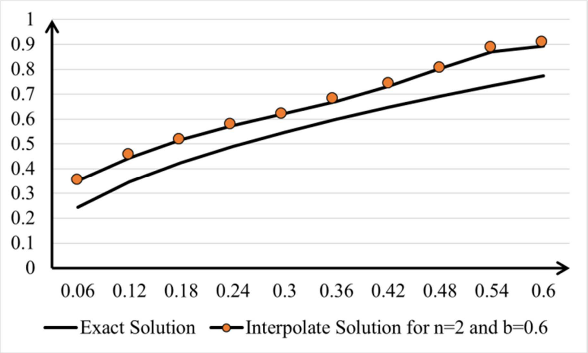

Figure 4. *The exact solution $u(x)$ and the interpolate solution $u_2^{0.6}(x)$ graphs.*

Example 2

Consider the weakly singular Abel-Volterra integral equation of the second kind $\phi(x)+\int_0^x k(x,t)\phi(t)\,dt = f(x)$, where $f(x)=\frac{\sin(x)}{\sqrt{x}}+\frac{\pi}{2}J_0\left(\frac{\pi}{2}\right)$ and $k(x,t)=\frac{-1}{\sqrt{x-t}}$. The exact solution $u(x)$ is given by $\phi(x)=\frac{\sin(x)}{\sqrt{x}}$ [20], for $0 \le x \le b$ where b is a real number. The given function $f(x)$ is singular at $x=0$ and the kernel $k(x,t)$ is singular when $x \to t$ and at $x=0$. Tables 5, 6, 7, and 8, show the exact solution values $\phi(x_i)$, the interpolate solution values $\phi_2^b(x_i)$, the absolute errors $R_2^b(x_i)$, and the relative errors $\mathrm{E}_2^b(x_i)$ at the corresponding set of nodes $x_i = b/10:b/10:b$. In figures 5, 6, 7, and 8 plotted are the graphs of the exact solution $\phi(x)$ and the graphs of the interpolant solutions $\phi_2^b(x)$ for $b=0.1,0.2,0.4,0.6$. From the exact solution of integral equation (1), we find that $\lim_{x\to 0} u(x)=0$ while the interpolation method gives, $\lim_{x\to 0} u_2^{0.1}(x)=0.10235, \lim_{x\to 0} u_2^{0.2}(x)=0.14475,$ $\lim_{x\to 0} u_2^{0.4}(x)=0.20182$, and $\lim_{x\to 0} u_2^{0.6}(x)=0.2433.$ The maximum error on each sub-interval $\left[b/10,b\right]$ are given by $\max\limits_{\xi[0.01,0.1]}\left|\Re_n^{0.1}\left(\xi\right)\right|\le 0.086806$, $\max\limits_{\xi[0.02,0.2]}\left|\Re_n^{0.2}\right|\left(\xi\right)\le 0.12276$, $\max\limits_{\xi[0.04,0.4]}\left|\Re_n^{0.4}\left(\xi\right)\right|\le 0.17338$, and $\max\limits_{\xi[0.06,0.6]}\left|\Re_n^{0.6}\left(\xi\right)\right|\le 0.21199$.

***Table 5.** The exact solutions $\phi(x_i)$, the interpolate solutions $\phi_2^{0.1}(x_i)$, the absolute error $R_2^{0.1}(x_i)$, and the relative error $\mathrm{E}_2^{0.1}(x_i)$ at $x_i = 0.01:0.01:0.1$.*

x_i	$\phi(x_i)$	$\phi_2^{0.1}(x_i)$	$R_2^{0.1}(x_i)$	$\mathrm{E}_2^{0.1}(x_i)$
0.01	0.099998	0.1275	0.027502	4.9999e-7
0.02	0.14141	0.15134	0.00993	7.0706e-7
0.03	0.17318	0.17239	0.00079	8.659e-7
0.04	0.19995	0.19045	0.0095	9.9973e-7
0.05	0.22351	0.20646	0.01705	1.1176e-6
0.06	0.2448	0.22196	0.02284	1.224e-6
0.07	0.26436	0.23831	0.02605	1.3218e-6
0.08	0.28254	0.2557	0.02684	1.4127e-6
0.09	0.2996	0.27195	0.02765	1.498e-6
0.1	0.3157	0.28104	0.03466	1.5785e-6

***Table 6.** The exact solutions $\phi(x_i)$, the interpolate solutions $\phi_2^{0.2}(x_i)$, the absolute error $R_2^{0.2}(x_i)$, and the relative error $\mathrm{E}_2^{0.2}(x_i)$ at $x_i = 0.02:0.02:0.2$*

x_i	$\phi(x_i)$	$\phi_2^{0.2}(x_i)$	$R_2^{0.2}(x_i)$	$\mathrm{E}_2^{0.2}(x_i)$
0.02	0.14141	0.17919	0.03778	7.0706e-7
0.04	0.19995	0.21207	0.01212	9.9973e-7
0.06	0.2448	0.24016	0.00464	1.224e-6
0.08	0.28254	0.26317	0.01937	1.4127e-6
0.1	0.3157	0.283	0.0327	1.5785e-6
0.12	0.34558	0.30263	0.04295	1.7279e-6
0.14	0.37294	0.32456	0.04838	1.8647e-6
0.16	0.3983	0.34903	0.04927	1.9915e-6
0.18	0.42198	0.37181	0.05017	2.1099e-6
0.2	0.44424	0.3816	0.06264	2.2212e-6

***Table 7.** The exact solutions $\phi(x_i)$, the interpolate solutions $\phi_2^{0.4}(x_i)$, the absolute error $R_2^{0.4}(x_i)$, and the relative error $\mathrm{E}_2^{0.4}(x_i)$ at $x_i = 0.04:0.04:0.4$.*

x_i	$\phi(x_i)$	$\phi_2^{0.4}(x_i)$	$R_2^{0.4}(x_i)$	$\mathrm{E}_2^{0.4}(x_i)$
0.04	0.19995	0.25059	0.05064	9.9973e-7
0.08	0.28254	0.29588	0.01334	1.4127e-6
0.12	0.34558	0.33258	0.013	1.7279e-6
0.16	0.3983	0.3604	0.0379	1.9915e-6
0.2	0.44424	0.38326	0.06098	2.2212e-6
0.24	0.48521	0.40682	0.07839	2.426e-6
0.28	0.52226	0.43548	0.08678	2.6113e-6
0.32	0.55608	0.46939	0.08669	2.7804e-6
0.36	0.58712	0.50054	0.08658	2.9356e-6
0.4	0.61572	0.50826	0.10746	3.0786e-6

***Table 8.** The exact solutions $\phi(x_i)$, the interpolate solutions $\phi_2^{0.6}(x_i)$, the absolute error $R_2^{0.6}(x_i)$, and the relative error $\mathrm{E}_2^{0.6}(x_i)$ at $x_i = 0.06:0.06:0.6$.*

x_i	$\phi(x_i)$	$\phi_2^{0.6}(x_i)$	$R_2^{0.6}(x_i)$	$\mathrm{E}_2^{0.6}(x_i)$
0.06	0.2448	0.30352	0.05872	1.224e-6
0.12	0.34558	0.3584	0.01282	1.7279e-6
0.18	0.42198	0.40069	0.02129	2.1099e-6
0.24	0.48521	0.43039	0.05482	2.426e-6
0.3	0.53954	0.45352	0.08602	2.6977e-6
0.36	0.58712	0.47823	0.10889	2.9356e-6
0.42	0.62919	0.51054	0.11865	3.1459e-6
0.48	0.66652	0.55033	0.11619	3.3326e-6
0.54	0.69965	0.58613	0.11352	3.4983e-6
0.6	0.72895	0.5891	0.13985	3.6448e-6

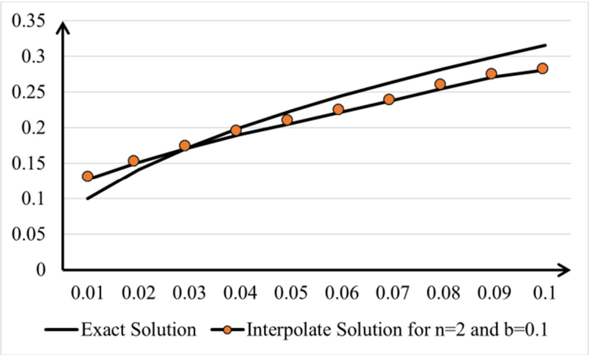

***Figure 5.** The exact solution $\phi(x)$ and the interpolate solution $\phi_2^{0.1}(x)$.*

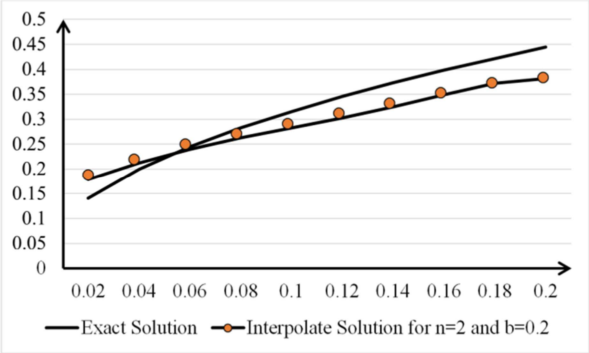

***Figure 6.** The exact solution $\phi(x)$ and the interpolate solution $\phi_2^{0.2}(x)$ graphs.*

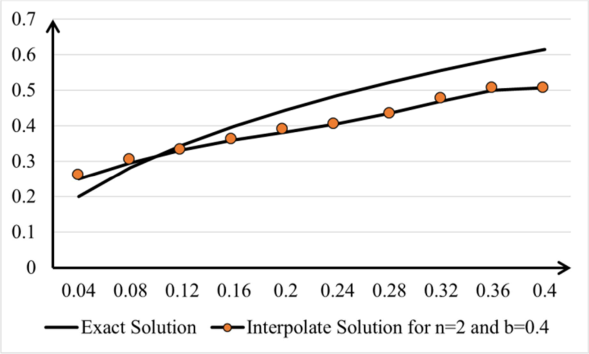

***Figure 7.** The exact solution $\phi(x)$ and the interpolate solutions $\phi_2^{0.4}(x)$ graphs.*

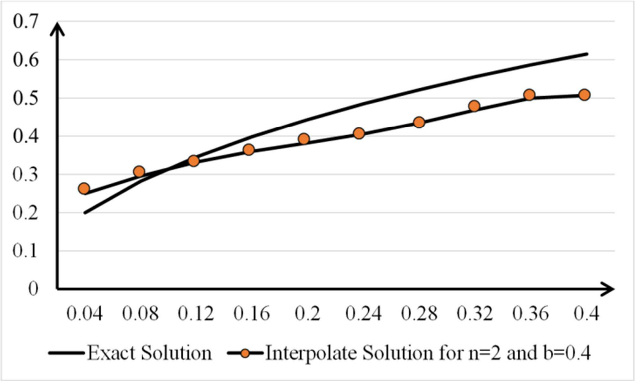

Figure 8. *The exact solutions $\phi(x)$ and the interpolate solution $\phi_2^{0.6}(x)$ graphs.*

Example 3

Consider the weakly singular Abel-Volterra integral equation of the second kind $\varphi(x)+\int_0^x k(x,t)\varphi(t)\,dt = f(x)$, where $f(x)=1$ and $k(x,t)=\frac{-1}{\sqrt{x-t}}$. The exact solution $u(x)$ is given by $\varphi(x)=e^{\pi x}+erfc\left(\sqrt{\pi x}\right)$ [20], for $0\le x\le b$ where b is a real number. The given function $f(x)$ is singular at $x=0$ and the kernel $k(x,t)$ is singular when $x\to t$ and at $x=0$. Tables 9, 10, 11, and 12, show the exact solution values $\varphi(x_i)$, the interpolate solution values $\varphi_2^b(x_i)$ and the absolute errors $R_2^b(x_i)$, and the relative errors $\mathrm{E}_2^b(x_i)$ at the corresponding set of nodes $x_i=b/10:b/10:b$. In figures 9, 10, 11, and 12, plotted are the graphs of the exact solution $\varphi(x)$ and the graphs of the interpolant solutions $\varphi_2^b(x)$ for $b=0.1,0.2,0.4,0.6$. From the presented interpolation method, it turns out that $\lim\limits_{x\to 0}\varphi_2^{0.1}(x)=\lim\limits_{x\to 0}\varphi_2^{0.2}(x)=\lim\limits_{x\to 0}\varphi_2^{0.4}(x)=\lim\limits_{x\to 0}\varphi_2^{0.6}(x)=1$ exactly equal $\varphi(0)=1$. We find $\max\limits_{\xi[0.01,0.1]}\left|\Re_n^{0.1}(\xi)\right|\le 0.009028$, $\max\limits_{\xi[0.02,0.2]}\left|\Re_n^{0.2}(\xi)\right|\le 0.011233$, $\max\limits_{\xi[0.04,0.4]}\left|\Re_n^{0.4}(\xi)\right|\le 0.012991$, and $\max\limits_{\xi[0.06,0.6]}\left|\Re_n^{0.6}(\xi)\right|\le 0.013567$.

Table 9. *The exact solutions $\varphi(x_i)$, the interpolate solutions $\varphi_2^{0.1}(x_i)$, the absolute error $R_2^{0.1}(x_i)$, and the relative error $\mathrm{E}_2^{0.1}(x_i)$ at $x_i=0.01:0.01:0.1$.*

x_i	$\varphi(x_i)$	$\varphi_2^{0.1}(x_i)$	$R_2^{0.1}(x_i)$	$\mathrm{E}_2^{0.1}(x_i)$
0.01	0.82767	0.92446	0.09679	4.1384e-6
0.02	0.76985	0.86795	0.0981	3.8493e-6
0.03	0.72981	0.81694	0.08713	3.6491e-6
0.04	0.69864	0.76944	0.0708	3.4932e-6
0.05	0.67296	0.72878	0.05582	3.3648e-6
0.06	0.65107	0.69908	0.04801	3.2553e-6
0.07	0.63196	0.68219	0.05023	3.1598e-6
0.08	0.61501	0.67554	0.06053	3.0751e-6
0.09	0.59977	0.67004	0.07027	2.9989e-6
0.1	0.58594	0.64818	0.06224	2.9297e-6

Table 10. *The exact solutions $\varphi(x_i)$, the interpolate solutions $\varphi_2^{0.2}(x_i)$, the absolute error $R_2^{0.2}(x_i)$, and the relative error $\mathrm{E}_2^{0.2}(x_i)$ at $x_i=0.02:0.02:0.2$.*

x_i	$\varphi(x_i)$	$\varphi_2^{0.2}(x_i)$	$R_2^{0.2}(x_i)$	$\mathrm{E}_2^{0.2}(x_i)$
0.02	0.76985	0.89497	0.12512	3.8493e-6
0.04	0.69864	0.81909	0.12045	3.4932e-6
0.06	0.65107	0.75285	0.10178	3.2553e-6
0.08	0.61501	0.69328	0.07827	3.0751e-6
0.1	0.58594	0.64428	0.05834	2.9297e-6
0.12	0.56161	0.61032	0.04871	2.808e-6
0.14	0.54071	0.59276	0.05205	2.7036e-6
0.16	0.52243	0.58766	0.06523	2.6121e-6
0.18	0.50621	0.58363	0.07742	2.531e-6
0.2	0.49165	0.5597	0.06805	2.4582e-6

***Table 11.** The exact solutions $\varphi(x_i)$, the interpolate solutions $\varphi_2^{0.4}(x_i)$, the absolute error $R_2^{0.4}(x_i)$, and the relative error $\mathrm{E}_2^{0.4}(x_i)$ at* $x_i = 0.04:0.04:0.4$.

x_i	$\varphi(x_i)$	$\varphi_2^{0.4}(x_i)$	$R_2^{0.4}(x_i)$	$\mathrm{E}_2^{0.4}(x_i)$
0.04	0.69864	0.85498	0.15634	3.4932e-6
0.08	0.61501	0.75528	0.14027	3.0751e-6
0.12	0.56161	0.6723	0.11069	2.808e-6
0.16	0.52243	0.60132	0.07889	2.6121e-6
0.2	0.49165	0.54623	0.05458	2.4582e-6
0.24	0.46643	0.511	0.04457	2.3321e-6
0.28	0.44517	0.4957	0.05053	2.2258e-6
0.32	0.42686	0.49454	0.06768	2.1343e-6
0.36	0.41084	0.49389	0.08305	2.0542e-6
0.4	0.39665	0.4701	0.07345	1.9833e-6

***Table 12.** The exact solutions $\varphi(x_i)$, the interpolate solutions $\varphi_2^{0.6}(x_i)$, the absolute error $R_2^{0.6}(x_i)$, and the relative error $\mathrm{E}_2^{0.6}(x_i)$ at* $x_i = 0.06:0.06:0.6$.

x_i	$\varphi(x_i)$	$\varphi_2^{0.6}(x_i)$	$R_2^{0.6}(x_i)$	$\mathrm{E}_2^{0.6}(x_i)$
0.06	0.65107	0.82561	0.17454	3.2553e-6
0.12	0.56161	0.71021	0.1486	2.808e-6
0.18	0.50621	0.61766	0.11145	2.531e-6
0.24	0.46643	0.54155	0.07512	2.3321e-6
0.3	0.43569	0.48514	0.04945	2.1785e-6
0.36	0.41084	0.45148	0.04064	2.0542e-6
0.42	0.39013	0.43937	0.04924	1.9507e-6
0.48	0.37247	0.44181	0.06934	1.8624e-6
0.54	0.35715	0.44409	0.08694	1.7858e-6
0.6	0.34368	0.42164	0.07796	1.7184e-6

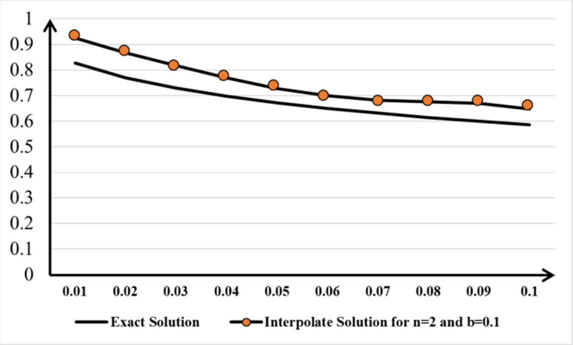

***Figure 9.** The exact solution $\varphi(x)$ and the interpolate solution $\varphi_2^{0.1}(x)$ graphs.*

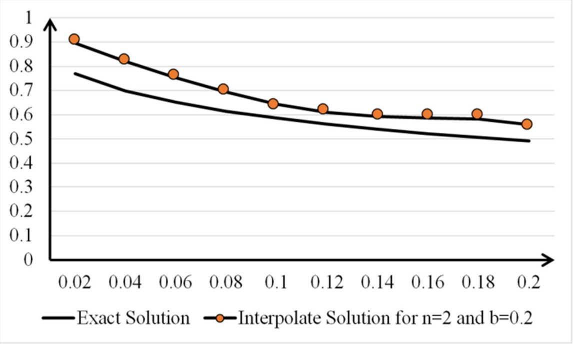

***Figure 10.** The exact solution $\varphi(x)$ and the interpolate solution $\varphi_2^{0.2}(x)$ graphs.*

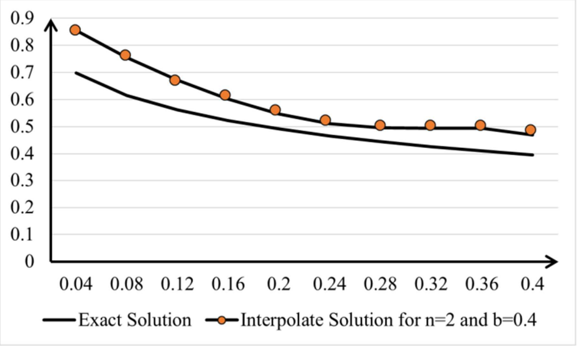

***Figure 11.** The exact solution $\varphi(x)$ and the interpolate solutions $\varphi_2^{0.4}(x)$ graphs.*

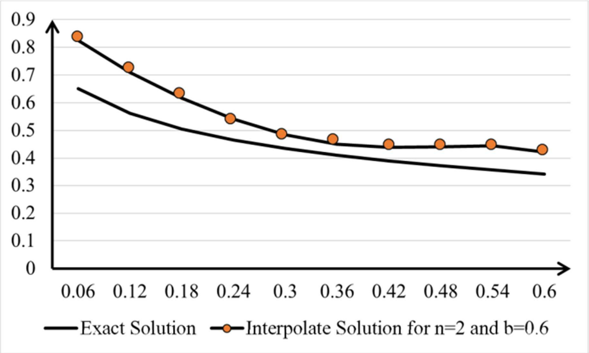

***Figure 12.** The exact solution $\varphi(x)$ and the interpolate solution $\varphi_2^{0.6}(x)$ graphs.*

4. Conclusion

In this paper, we created an interpolation method for solving Volterra integral equations of the second kind with a weakly singular kernel. For isolating the kernel singularity, we established two rules for exemplary choosing the node distributions of the two variables of the kernel in such a manner that the quantity under the square root never becomes negative or zero. We implemented this idea by carrying out the step sizes depending on some quantity greater than or equal to zero, which consists of the upper integration variable and the interpolant degree. We applied the presented method under the consideration of these two rules to obtain a double interpolant kernel completely free from any kind of singularities. The substitution of the interpolant solution on both sides of the integral equation enabled us to access an algebraic linear system without applying the collocation points. The interpolant solution estimated the solution values at the singular points. The obtained interpolate solutions of the three solved examples are faster to converge to the exact ones using the lowest interpolant degree. This ensures the superiority of the presented method compared with others.

References

[1] Kendall E. Atkinson: The Numerical Solution of Integral Equations of the Second Kind. Cambridge University Press (2010).

[2] Prem K. Kythe and Pratap Puri, 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] H. Brunner, Collocation methods for Volterra integral and related functional Equations, Cambridge University Press, Cambridge, 2004.

[5] E. S. Shoukralla, "A Numerical Method for Solving Fredholm Integral Equations of the First Kind with Logarithmic Kernels and Singular Unknown Functions" Journal of Applied and Computational Mathematics, Springer Nature, (2020) 6: 172.

[6] E. S. Shoukralla, "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", Vol. 51, issue. 1, IJAM_51_1_08, (2021).

[7] E. S. Shoukralla, M. A. Markos, "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), Cairo, Egypt, 202-207. 2018.

[8] E. S. Shoukralla and M. A. Markos, "The economized monic Chebyshev polynomials for solving weakly singular Fredholm integral equations of the first kind," Asian-European Journal of Mathematics, Vol. 12, No. 1, pp. 2050030-1-2050030-10, October 2018.

[9] E. S. Shoukralla, "Approximate solution to weakly singular integral equations, Journal of appl. Math Modelling. 20 (1996) 800-803.

[10] E. S. Shoukralla, "Numerical solution of Helmholtz equation for an open boundary in space, Journal of appl. Math Modeling. 21 (4): 231-232, 1997.

[11] The Barycentric Lagrange Interpolation via Maclaurin Polynomials for Solving the Second Kind Volterra Integral Equations, Proceedings of 2020 15th IEEE International Conference on Computer Engineering and Systems (ICCES), Ain Shams University, Cairo, Egypt, 2020.

[12] Multi-techniques method for Solving Volterra Integral Equations of the Second Kind, Proceedings of 2019 14th IEEE International Conference on Computer Engineering and Systems (ICCES), Ain Shams University, Cairo, Egypt, 2019.

[13] E. S. Shoukralla, H. Elgohary and B. M. Ahmed, "Barycentric Lagrange interpolation for solving Volterra integral equations of the second kind", Journal of Physics, England, Conference Series, 1447 (2020), 012002.

[14] Jean-Paul Berrut, Lloyd N. Trefethen, Barycentric Lagrange Interpolation, SIAM REVIEW, Society for Industrial and Applied Mathematics, 46 (3) (2004) 501–517.

[15] Nicholas J. Higham, The numerical stability of barycentric Lagrange interpolation, IMA Journal of Numerical Analysis, 24 (2004) 547-556.

[16] Zhang Xiao-yong, "Jacobi spectral method for the second-kind Volterra integral equations with a weakly singular kernel", Applied Mathematical Modelling, Volume 39, Issue 15, 1 August 2015, Pages 4421-4431.

[17] Can Huang and Martin Stynes, "Spectral Galerkin methods for a weakly singular Volterra integral equation of the second kind", IMA Journal of Numerical Analysis (2017) 37, 1411–1436.

[18] Lina Wang, Hongjiong Tian, Lijun Yi, "An hp-version of the discontinuous Galerkin time-stepping method for Volterra integral equations with weakly singular kernels" Applied Numerical Mathematics 161 (2021) 218–232.

[19] Jingjun Zhao, Teng Long and Yang Xu, "Super Implicit Multistep Collocation Methods for Weakly Singular Volterra Integral Equations", Numer. Math. Theor. Meth. Appl., Vol. 12, No. 4, pp. 1039-1065, 2019.

[20] Hossein BEYRAMI and Taher LOTFI," On the local super convergence of the fully discretized multiprojection method for weakly singular Volterra integral equations of the second kind", Turkish Journal of Mathematics, (2018) 42: 1400-1423.