

RESEARCH ARTICLE

Open Access



Computational method for solving weakly singular Fredholm integral equations of the second kind using an advanced barycentric Lagrange interpolation formula

E. S. Shoukralla¹, Nermin Saber² and Ahmed Y. Sayed^{3*} 

*Correspondence:

AHMED_BADR@m-eng.
helwan.edu.eg

³ Physics and Engineering
Mathematics Department,
Faculty of Engineering
at Mataria, Helwan University,
Cairo, Egypt

Full list of author information
is available at the end of the
article

Abstract

In this study, we applied an advanced barycentric Lagrange interpolation formula to find the interpolate solutions of weakly singular Fredholm integral equations of the second kind. The kernel is interpolated twice concerning both variables and then is transformed into the product of five matrices; two of them are monomial basis matrices. To isolate the singularity of the kernel, we developed two techniques based on a good choice of different two sets of nodes to be distributed over the integration domain. Each set is specific to one of the kernel arguments so that the kernel values never become zero or imaginary. The significant advantage of the two presented techniques is the ability to gain access to an algebraic linear system equivalent to the interpolant solution without applying the collocation method. Moreover, the convergence in the mean of the interpolant solution and the maximum error norm estimation are studied. The interpolate solutions of the illustrated four examples are found strongly converging uniformly to the exact solutions.

Keywords: Weakly singular integrals, Fredholm integral equations, Barycentric Lagrange interpolation, Electromagnetism, Scattering, Radiation

Introduction

The solutions of the initial, boundary, or mixed value problems have become common to be obtained through the integral equation method. This technique converts the solution of the value problems to the solutions of certain equivalent boundary integral equations of specified types and kinds. One of the common equivalent boundary integral equations is the weakly singular Fredholm integral equations of the second kind. These equations appear in many engineering fields, such as radiation, potential theory, scattering, electromagnetism, and other scientific fields [1–5]. The singularities of the integral equation are due to the singular kernel or the singular unknown function or the singularity of both. For example, the Dirichlet boundary value problems for the Laplace equation for an open arc in the plane is predominantly reduced to the solution of a weakly singular Fredholm integral equation of the first kind whose unknown function is singular at the

endpoints of the integration domain and has weakly logarithmic kernels [6–14]. Dmitriev et al. [6] provided an iterative method for solving the Fredholm integral equation of the first kind with a weakly singular logarithmic kernel and a nonsingular unknown function. Shoukralla [7, 8] presented two methods for the solution of the logarithmic singular kernel Fredholm integral equation with a singular unknown function. The kernel singularity is isolated analytically, depending on the Kantorovich technique, and the unknown functions are approximated on the basis of the Taylor and Chebyshev polynomials with an analytical treatment of the singularity. These techniques provided acceptable solutions at that time, regardless of the difficulty and the complexity of the two procedures. Shoukralla [9, 10] solved the same integral equation on the basis of Chebyshev polynomials of the second kind, thus providing high-accuracy results.

Shoukralla et al. [11–13] solved a certain class of singular Fredholm integral equation of the first kind with singular logarithmic kernels and singular unknown functions on the basis of monic and economized monic Chebyshev functions with different approaches for removing the singularities. In this study, we focus on the numerical solution of the second kind of Fredholm equations with another type of kernel's singularity, which needs another technique different from those for the singular logarithmic kernels of Fredholm equations of the first kind. Many methods for solving weakly singular Fredholm integral equations of the second kind have been published [14–25]. For example, Yin et al. [14], used the Jacobi–Gauss quadrature formula to approximate the integral operator in the numerical implementation of the spectral collocation method and established the spectral Chebyshev collocation method for solving Fredholm integral equations of the second kind with the weakly singular kernel. This method shows that the errors of the approximate solution decay exponentially in infinity and weighted norms. Behzadi et al. [15] developed some modifications on the generalization of the Euler–Maclaurin summation formula by using Bernoulli functions to construct such generalized quadrature and to construct a numerical method based on the trapezoidal rule for solving weakly singular integral equations.

In this study, we make progress toward the application of some advanced single and double barycentric Lagrange interpolation formulas and how to adapt them to be applicable to completely isolating the kernel singularity and find accurate solutions of the weakly singular Fredholm equations of the second kind. Shoukralla et al. [26–29] developed a new version of the traditional barycentric Lagrange interpolation [30] and applied it successfully to solve linear and nonsingular Volterra integral equations of the second kind. For weakly singular Fredholm equations of the second kind, the matter is more complicated due to some difficulties related to the singularity of the kernel.

This study focuses on the application of two advanced barycentric interpolation formulas to solve the weakly singular Fredholm integral equation of the second kind with two innovative techniques for the treatment of the kernel's singularity depending on the perfect choice of the node distribution rules. Naturally, our primary aim is to reduce computation complexity, but whether or how fast the interpolant solution converges to the exact one is at least as important.

One of the advantages of the presented techniques is the designed rules for the distribution of nodes so that they always remain within the domain of the integration and never be outside. Moreover, these rules are designed so that the difference between the

nodes subjected to the two variables of the kernel remains always positive. Thus, based on this idea, the numerical solutions become stable [31] on the whole interval even at the endpoints, as shown in the solved examples.

We begin by interpolating the unknown and data functions by using the advanced single matrix-form barycentric interpolate polynomials; each is expressed through four matrices, and one of which is the monomial basis functions matrix [32]. The weakly singular kernel is then interpolated twice concerning its two arguments; the first interpolation concerning the first variable of a positive sign with node distribution on the right half of the integration domain, and the second interpolation concerning the variable of negative sign and the distribution of the nodes will be held on the other left half of the integration domain. This generous scheme ensures that the difference between the kernel's two variables always remains positive, thus completely erasing the singularity of the kernel and expressing it through five matrices; two of which are monomial basis matrices.

Additional advantages of these techniques are not only to simplify the calculations but also to gain access to an equivalent linear system of algebraic equations without applying the collocation method. The implementation of this idea is achieved by substituting the interpolant unknown function on both sides of the integral equation. By solving the obtained algebraic linear system directly, the unknown coefficient matrix can be found, consequently finding the interpolant unknown function. The six examples are solved by using the two presented techniques; examples 1–4 are for weakly singular equations, while examples 5–6 are for linear nonsingular equations. The first three examples are also solved by a trapezoidal approach mentioned in [14], whereas the fourth boundary integral equation, which arises from the problem of radiation, potential theory, scattering theory, electromagnetism, and hydrodynamics, is solved in [15]. The solutions of examples 1–4 which are obtained by the two presented techniques are found to strongly converge to the exact solutions compared with [14, 15]. The solutions to examples 5–6 which are obtained by the two presented techniques are found to be equal to the exact solutions. The given tables and graphs demonstrate the originality, eligibility, and accuracy of the presented new method.

Advanced barycentric Lagrange interpolation formula

The question of expressing a given function by an interpolant is vital in approximation theory, as well as in computational methods. A remarkable advantage of Lagrange interpolation is its independence of the arrangement of the selected nodes, although the efficient results require a few nodes. By contrast, the increasing number of nodes leads to complicate the scheme and the instability of the numerical solution. That is, the barycentric Lagrange interpolation is a fantastic formula to increase the performance of traditional Lagrange interpolation. In this section, we provide a different mathematical formula in form and content that exceeds the well-known traditional barycentric Lagrange formula. The new formula is expressed through matrices; one of them is the monomial basis matrix, which is canceled in the solution's procedure. Thus, the steps of the solution are reduced, the round-off error is minimized, and high-precision solutions are provided.

Let the function $f(x)$ be defined on $[a, b]$ as the tabulated function $f(x_i) = f_i$; $i = \overline{0, n}$ for the set of $(n + 1)$ equally spaced distinct nodes $\{x_i\}_{i=0}^n$ such that $x_i = a + ih$, where the step size h is defined by $h = \frac{b-a}{n}$. Then, Berrut et al. [14] provided the barycentric Lagrange interpolating polynomial of degree n , $\tilde{f}_n(x)$, which interpolates the tabulated function $f(x_i) = f_i$ such that $\tilde{f}_n(x_i) = f(x_i) = f_i$ in the following form:

$$\tilde{f}_n(x) = \sum_{i=0}^n \frac{w_i}{x - x_i} f_i \bigg/ \sum_{i=0}^n \frac{w_i}{x - x_i}; \quad w_i = (-1)^i \binom{n}{i}. \quad (1)$$

Although Formula (1) is simpler than the traditional Lagrange interpolating polynomial [14], it is still difficult to apply for interpolating the unknown functions, as well as the kernels of integral equations of any types and kinds because it hinders the facilitation of the steps of the solution and causes some computational impediments. Therefore, we adapt this formula before using it so that it becomes easier to apply for solving integral equations. Using some operational matrix algebra, we can increase the computational efficiency of Formula (1) and achieve an improved matrix formula by expanding the numerator and distributing it on the denominator by separating the barycentric weights w_i . Thus, we obtain $\tilde{f}_n(x)$ in the modified matrix form

$$\tilde{f}_n(x) = \Psi(x)WF. \quad (2)$$

Here, $\Psi(x)$ is the $1 \times (n + 1)$ row matrix, $W = \text{diag}\{w_0, w_1, \dots, w_n\}$ is the $(n + 1) \times (n + 1)$ square diagonal matrix whose entries w_i are defined by (1), and $F^T = [f_i]_{i=0}^n$ is $(n + 1) \times 1$ column matrix whose entries f_i are the functional values of $f(x)$ such that

$$\Psi(x) = [\psi_i(x)]_{i=0}^n; \quad \psi_i(x) = \frac{\xi_i(x)}{\phi(x)}, \quad \phi(x) = \sum_{i=0}^n w_i \xi_i(x), \quad \xi_i(x) = \frac{1}{x - x_i}; \quad i = \overline{0, n}. \quad (3)$$

By studying the behavior of matrix Formula (2), we can perform analysis so that the so-called monomial matrix is separated. This idea can be implemented by extracting the coefficients of the barycentric functions of the matrix $\Psi(x)$. Moreover, the numerator and the denominator have common factors, and these factors annihilate each other. This idea gives us the incentive to rearrange the terms of each barycentric function in the ascending power of x or simply expand each function into a Maclaurin polynomial in a matrix form of multiplied two matrices; one of which is the monomial basis matrix, and the other is the Maclaurin coefficient. On the basis of this idea, Formula (2) is expressed via four matrices as follows:

$$\tilde{f}_n(x) = X(x)CWF, \quad (4)$$

where the $1 \times (n + 1)$ monomial basis row matrix $X(x)$ and the $(n + 1) \times (n + 1)$ square Maclaurin coefficient matrix C are defined by

$$X(x) = [x^i]_{i=0}^n, \quad C^T = [c_{ij}]_{i,j=0}^n; \quad c_{ij} = \frac{\psi_i^{(j)}(0)}{j!} \forall i, \quad j = \overline{0, n}. \quad (5)$$

Thus, we have derived a simple and magnificent matrix formula for raising the computational efficiency of the traditional barycentric Lagrange interpolation (1) that can be easily applicable to find the interpolant polynomial of any function $f(x)$ defined on the interval $[a, b]$. We name the right-hand side of (4) “advanced barycentric Lagrange single interpolation formula.” Applying Formula (4) for interpolating the data and unknown functions of integral equations remarkably reduces the solution steps due to some operational matrix abbreviations, considerably contributing to the reduction in round-off errors and saving time. Now, we apply the new Formula (4) to solve weakly singular Fredholm integral equations of the second kind.

Advanced barycentric interpolation formulas for solving weakly singular fredholm integral equations of the second kind

Here, we present two new techniques for solving weakly singular Fredholm integral equations of the second kind. This method starts by interpolating the unknown and data functions using Formula (4). As for the kernel, we use Formula (4) twice to obtain a double interpolant polynomial through five matrices. In this manner, we provide two techniques for choosing the distribution nodes of the two main variables x and t of the kernel. In the first technique, the x -nodes are distributed on the right half of the integration domain, whereas the t -nodes are distributed on the left half. The step sizes for the two sets of nodes depend on some real numbers $\delta_1, \delta_2 \geq 0$ that depend on the degree of the interpolation degrees. In the second technique, we present two different sets of node distributions corresponding to two variables, all of which are distributed on the entire integration domain. Consider the weakly singular Fredholm integral equation of the second kind.

$$u(x) = \varphi(x) + \int_a^b k(x, t)u(t)dt; \quad a \leq x \leq b, \quad (6)$$

where $\varphi(x)$ is a given function, and $u(x)$ is the unknown function defined on $L^2[a, b]$. Here, the given kernel $k(x, t)$ takes the form $k(x, t) = \frac{1}{|x-t|^\alpha}$; $0 < \alpha < 1$. Moreover, $\max_{x, t \in [a, b]} |k(x, t)| \leq N$, $\max_{x \in [a, b]} |\varphi(x)| \leq M$, $\max_{x \in [a, b]} |u(x)| \leq L$ for N, M, L are assumed to be real numbers.

The first technique

Let $\tilde{\varphi}_n(x)$ be the single interpolant polynomial that interpolates $\varphi(x)$ of (6) on the basis of Formula (4) such that $\tilde{\varphi}_n(x) \approx \varphi(x)$ and $\tilde{\varphi}_n(x_i) = \varphi(x_i)$ for the set of equidistant nodes $\{x_i\}_{i=0}^n$; $x_i = a + ih$, $h = \frac{b-a}{n}$. By using the new Formula (4), $\varphi(x)$ can be replaced by its interpolant polynomial $\tilde{\varphi}_n(x)$ of degree n in the matrix form

$$\tilde{\varphi}_n(x) = X(x)CW\Phi = X(x)P\Phi; \quad P = CW, \quad (7)$$

where $P = CW$ is the $(n+1) \times (n+1)$ square matrix, and Φ is the $(n+1) \times 1$ column matrix such that

$$P = CW, \quad P^T = [p_{ij}]_{i,j=0}^n; \quad p_{ij} = c_{ij}w_i; \quad i, j = \overline{0, n}, \quad \Phi^T = [\varphi_i]_{i=0}^n; \quad \varphi_i = \varphi(x_i); \quad i = \overline{0, n}, \quad (8)$$

and c_{ij} are calculated by (5). Similarly, the unknown function $u(x)$, as well as $\varphi(x)$, can be interpolated to obtain its unknown single interpolant polynomial $\tilde{u}_n(x)$ in the following matrix form:

$$\tilde{u}_n(x) = X(x)PU, \quad (9)$$

where $U = [u_i]_{i=0}^n$ is the $(n+1) \times 1$ unknown coefficient column matrix to be determined, where the entries $\{u_i\}_{i=0}^n$ are the undetermined coefficients of the unknown single interpolant polynomial.

Consequently, for the weakly singular kernel $k(x, t) = \frac{1}{|x-t|^\alpha}$, which is singular when $x \rightarrow t$, we interpolate it twice; the first interpolation is performed with respect to x , and the second is performed with respect to t so that we can obtain the double interpolant polynomial $\tilde{k}_{n,n}(x, t)$ of two variables x and t . The mathematical properties of the kernel force us to design an innovative new technique that has the potential to remove this singularity. This goal can only be achieved under the important and necessary condition that $x > t$. Thus, we adopt an approach based on the appropriate choice of two different sets of nodes; the first set $\{\tilde{x}_i\}_{i=0}^n$ is distributed on the right-half interval of the integration domain $\left[\frac{b-a}{2}, b\right]$, and the second set of nodes $\{\tilde{t}_i\}_{i=0}^n$ is distributed on the left-half interval $\left[a, \frac{b-a}{2}\right]$. This yields two barycentric function summations $\rho(x), \tilde{\rho}(x)$; the first summation $\rho(x)$ corresponds to the set of nodes $\{\tilde{x}_i\}_{i=0}^n$ and the barycentric functions $\varpi_i(x) = \frac{\zeta_i(x)}{\rho(x)}$ for $\zeta_i(x) = \frac{1}{x-\tilde{x}_i}$, whereas the second barycentric function summation $\tilde{\rho}(x)$ corresponds to the set of nodes $\{\tilde{t}_i\}_{i=0}^n$ and the barycentric functions $\tilde{\varpi}_i(t) = \frac{\tilde{\zeta}_i(t)}{\tilde{\rho}(t)}$ for $\tilde{\zeta}_i(t) = \frac{1}{t-\tilde{t}_i}$. We define \tilde{x}_i and \tilde{t}_i as follows:

$$\tilde{x}_i = a + 0.5 + ih_1; \quad h_1 = \frac{b-a-4\delta_1}{2n}, \quad \tilde{t}_i = a + ih_2; \quad h_2 = \frac{b-a-4\delta_2}{2n}; \quad i = \overline{0, n}. \quad (10)$$

We choose $\delta_1, \delta_2 \geq 0$ such that $\frac{b-a}{2} < h_1 < b$ and $a < h_2 < \frac{b-a}{2}$. Moreover, we put $h_2 = \frac{b-3a-4\delta_2}{n+0.1}$ for the kernel of the form $|1-t|^{-1/2}$, that is, if $x = 1$. The two summations $\rho(x), \tilde{\rho}(x)$ are defined by

$$\rho(x) = \sum_{i=0}^n w_i \zeta_i(x); \quad \tilde{\rho}(t) = \sum_{i=0}^n w_i \tilde{\zeta}_i(t); \quad \zeta_i(x) = \frac{1}{x-\tilde{x}_i}, \quad \tilde{\zeta}_i(t) = \frac{1}{t-\tilde{t}_i}.$$

By using the same strategy used to drive Formula (4), the kernel $k(x, t)$ can be interpolated using the set of nodes $\{\tilde{x}_i\}_{i=0}^n$ via four matrices as follows:

$$\tilde{k}_{n,n}(x, t) = X(x)C WK(\tilde{x}_i, t), \quad (11)$$

where $K(\tilde{x}_i, t)$ is the column matrix such that

$$K^T(\tilde{x}_i, t) = [k(\tilde{x}_0, t) \ k(\tilde{x}_1, t) \ k(\tilde{x}_2, t) \ \dots \ k(\tilde{x}_n, t)]. \quad (12)$$

In the same context, we again interpolate each function $k(\tilde{x}_i, t)$ for $i = \overline{0, n}$ by using the set of nodes $\{\tilde{t}_j\}_{j=0}^n$. After strenuous substitution and abbreviations, which are performed using some matrix operations, we obtain the kernel through five matrices; two of which are the monomial basis function matrices, that is, the row monomial

basis function matrix $X(x)$ subjected to x and the column monomial basis function matrix $X^T(t)$ subjected to t . Thus, we obtain the advanced barycentric double interpolant polynomial $\tilde{k}_{n,n}(x, t)$ via five matrices as follows:

$$\tilde{k}_{n,n}(x, t) = X(x)AKBX^T(t), \quad (13)$$

where the $(n+1) \times (n+1)$ square matrix K is calculated as follows:

$$K = [w_{ij}k_{ij}]_{i,j=0}^n; \quad k_{ij} = k(\tilde{x}_i, \tilde{t}_j); w_{ij} = w_i \times w_j; \quad i, j = \overline{0, n}. \quad (14)$$

Here, $A^T = [a_{ij}]_{i,j=0}^n$ and $B^T = [b_{ij}]_{i,j=0}^n$ are $(n+1) \times (n+1)$ square matrices whose entries a_{ij} and b_{ij} can be calculated by

$$a_{ij} = \frac{\varpi_i^{(j)}(0)}{j!}, b_{ij} = \frac{\tilde{\varpi}_i^{(j)}(0)}{j!} \quad \forall i, j = \overline{0, n}. \quad (15)$$

Moreover, substituting $\tilde{k}_{n,n}(x, t)$ given by (13) and $\tilde{u}_n(t)$ given by (9) into the right side of (6), we obtain $\tilde{u}_n(x)$ in the following matrix form:

$$\tilde{u}_n(x) = \varphi(x) + \int_a^b X(x)N\tilde{X}(t)PU dt, \quad (16)$$

where the $(n+1) \times (n+1)$ square matrix N and the $(n+1) \times (n+1)$ square matrix $\tilde{X}(t)$ are defined by

$$N = AKB, \tilde{X}(t) = X^T(t)X(t) = [t^{i+j}]_{i,j=0}^n. \quad (17)$$

By integrating the right side of (16), we obtain

$$\tilde{u}_n(x) = \varphi(x) + X(x)NHPU. \quad (18)$$

Here, the $(n+1) \times (n+1)$ square matrix H is given by

$$H = \int_a^b \tilde{X}(t) dt = [h_{ij}]_{i,j=0}^n; \quad h_{ij} = \int_a^b t^{i+j} dt = \frac{t^{i+j+1}}{i+j+1} \Big|_a^b = \frac{b^{i+j+1} - a^{i+j+1}}{i+j+1}; \quad i, j = \overline{0, n}. \quad (19)$$

Furthermore, by replacing $\tilde{u}_n(x)$ defined by (9) with $u(x)$ on the left side of (6) and replacing $\tilde{k}_{n,n}(x, t)\tilde{u}_n(t)$ with $u(t)k(x, t)$ on the right side, we obtain

$$X(x)NHPU - X(x)NHNHPU = X(x)NHP\Phi. \quad (20)$$

Simplifying (20) yields the linear algebraic system

$$(I - NH)PU = P\Phi. \quad (21)$$

By applying any direct method, we can solve system (21) to obtain the unknown coefficient column matrix U :

$$U = P^{-1}M^{-1}P\Phi; \quad M = (I - NH). \quad (22)$$

Accordingly, the interpolant solution that was given by (9) then takes the simple matrix form

$$\tilde{u}_n(x) = X(x)PP^{-1}M^{-1}P\Phi = X(x)\Omega, \quad (23)$$

where Ω is the $(n+1) \times 1$ column matrix

$$\Omega = M^{-1}P\Phi = [\gamma_i]_{i=0}^n. \quad (24)$$

The entries $\{\gamma_i\}_{i=0}^n$ of Ω can be easily calculated from the product of the multiplied three known coefficient matrices $M^{-1}P\Phi$. Hence, the interpolant polynomial solution of the considered integral Eq. (6) is given by

$$\tilde{u}_n(x) = \sum_{i=0}^n \gamma_i x^i; \quad a \leq x \leq b. \quad (25)$$

The second technique

We choose the two sets of nodes $\{\tilde{x}_i\}_{i=0}^n$ and $\{\tilde{t}_j\}_{j=0}^n$; each $(n+1)$ equally spaced distinct node corresponds to the two variables x, t . These sets of nodes are distributed on the whole domain $[a, b]$ and never come outside. Based on these two sets of nodes that depend on step sizes h_1, h_2 , which by extension depend on some positive numbers $\delta_1 \geq 0, \delta_2 \geq 0$, we define

$$h_1 = \frac{(b - \delta_1) - (a + \delta_1)}{n}, \quad h_2 = \frac{(b - \delta_2) - (a + \delta_2)}{n}, \quad (26)$$

and

$$x_i = (a + \delta_1) + ih_1, \quad t_j = (a + \delta_2) + jh_2; \quad i, j = \overline{0, n}. \quad (27)$$

Based on the modified matrix forms (2)–(5), we obtain $\tilde{u}_n(x)$ and $\tilde{f}_n(x)$ in the form

$$\tilde{u}_n(x) = \Psi(x)WU, \quad \tilde{f}_n(x) = \Psi(x)WF. \quad (28)$$

The kernel $k(x, t)$ is now interpolated twice; the first interpolation is performed with respect to the argument x , whereas the second interpolation is performed with respect to the argument t in inverse matrix orders. Thus, we obtain the modified barycentric double interpolant kernel $\tilde{k}_{n,n}(x, t)$ in the form

$$\tilde{k}_{n,n}(x, t) = \Psi(x)KN^T(t). \quad (29)$$

Here, $N^T(t) = [n_j(t)]_{j=0}^n$ is the $(n+1) \times 1$ column matrix of the barycentric functions $n_j(x)$, where

$$n_j(t) = \frac{\xi_i(t)}{\varphi(t)}, \quad \xi_j(t) = \frac{1}{t - t_j}, \quad \varphi(t) = \sum_{j=0}^n w_j \xi_j(x); \quad w_j = (-1)^j \binom{n}{j}, \quad (30)$$

and the known square matrix K is given by

$$K = [k_{ij}]_{i,j=0}^n; \quad k_{ij} = w_{ij} k(x_i, t_j); \quad w_{ij} = w_i \times w_j. \quad (31)$$

By virtue of Eqs. (28) and (29), the product of the single interpolant unknown function $\tilde{u}_n(t)$ by the double interpolated kernel $k_{n,n}(x, t)$ can be replaced by the following matrix form:

$$k_{n,n}(x, t)u_n(t) = \Psi(x)KN^T(t)\Psi(t)WU = \Psi(x)K\tilde{\Psi}(t)WU; \quad \tilde{\Psi}(t) = N^T(t)\Psi(t). \quad (32)$$

Now, replacing $k(x, t)u(t)$ in the right side of (6) with $k_{n,n}(x, t)u_n(t)$ given by (31), we obtain $\tilde{u}_n(x)$ in the form

$$\tilde{u}_n(x) = f(x) + \Psi(x)K\Phi WU; \quad \Phi = \int_a^b \tilde{\Psi}(t)dt. \quad (33)$$

Moreover, by replacing the matrix–vector single interpolant $\tilde{u}_n(x)$ that was given by (28) into both sides of (6), replacing the matrix–vector double interpolated kernel for $k_{n,n}(x, t)$ that was given by (29) with $k(x, t)$, and replacing $f(t)$ with $\tilde{f}_n(t)$ that was given by (28), we find that

$$\Psi(x)K\Phi WU - \int_a^b \Psi(x)K\tilde{\Psi}(t)K\Phi WUdt = \int_a^b \Psi(x)K\tilde{\Psi}(t)WFdt. \quad (34)$$

Simplifying Eq. (34) yields

$$\Psi(x)K\Phi WU - \Psi(x)K\Phi K\Phi WU = \Psi(x)K\Phi WF. \quad (35)$$

From this equation, we can find the required unknown coefficients matrix $U = (W - K\Phi W)^{-1}WF$; by substituting into (28), we obtain the matrix–vector single interpolant $\tilde{u}_n(x)$

$$\tilde{u}_n(x) = \Psi(x)W(W - K\Phi W)^{-1}WF. \quad (36)$$

Convergence and error analysis

In this section, we study the convergence in the mean [33, 34] of the interpolant unknown function $\tilde{u}_n(x)$ described in first technique to the exact solution $u(x)$.

Theorem 4.1 Assume $u(x) \in X$ is a sufficiently smooth exact solution of (6) such that $\max_{x \in [a,b]} |u(x)| \leq \varepsilon$. Then,

$$\lim_{n \rightarrow \infty} \|u(x) - \tilde{u}_n(x)\|_2 = 0. \quad (37)$$

Proof Let the exact solution $u(x)$ be expanded into a Maclaurin series

$$u(x) = \sum_{s=0}^{\infty} \alpha_s x^s. \quad (38)$$

Then, we have

$$\|u(x) - \tilde{u}_n(x)\|_2 = \int_a^b |u(x) - \tilde{u}_n(x)|^2 dx = \int_a^b |u(x)|^2 dx + \int_a^b |\tilde{u}_n(x)|^2 dx - 2 \int_a^b |u(x)\tilde{u}_n(x)| dx. \quad (39)$$

Here, we find from Eq. (25) that

$$\begin{aligned} \int_a^b |\tilde{u}_n(x)|^2 dx &= \int_a^b \left| \sum_{i=0}^n \gamma_i x^i \sum_{j=0}^n \gamma_j x^j \right| dx = \sum_{i=0}^n \sum_{j=0}^n \gamma_i \gamma_j \int_a^b x^{i+j} dx \\ &= \sum_{i=0}^n \sum_{j=0}^n \frac{\gamma_i \gamma_j}{i+j+1} (b^{i+j+1} - a^{i+j+1}). \end{aligned} \quad (40)$$

From (38), we obtain

$$\begin{aligned} \int_a^b |u(x)|^2 dx &= \int_a^b \left| \sum_{s=0}^{\infty} \alpha_s x^s \sum_{k=0}^{\infty} \alpha_k x^k \right| dx = \sum_{s=0}^{\infty} \sum_{k=0}^{\infty} \alpha_s \alpha_k \int_a^b x^{s+k} dx \\ &= \sum_{s=0}^{\infty} \sum_{k=0}^{\infty} \frac{\alpha_s \alpha_k}{s+k+1} (b^{s+k+1} - a^{s+k+1}). \end{aligned} \quad (41)$$

From the Schwarz inequity, we have

$$\begin{aligned} \int_a^b |\tilde{u}_n(x)u(x)| dx &\leq \left[\int_a^b |\tilde{u}_n(x)|^2 dx \right]^{\frac{1}{2}} \times \left[\int_a^b |u(x)|^2 dx \right]^{\frac{1}{2}} \\ &= \left[\sum_{i=0}^n \sum_{j=0}^n \frac{\gamma_i \gamma_j}{i+j+1} (b^{i+j+1} - a^{i+j+1}) \right]^{\frac{1}{2}} \\ &\quad \times \left[\sum_{s=0}^{\infty} \sum_{k=0}^{\infty} \frac{\alpha_s \alpha_k}{s+k+1} (b^{s+k+1} - a^{s+k+1}) \right]^{\frac{1}{2}}. \end{aligned} \quad (42)$$

By substituting Eqs. (32), (33), and (34) into Eq. (31) on the basis of $s, k \rightarrow \infty$, we have proven that

$$\begin{aligned} \lim_{n \rightarrow \infty} \|u(x) - \tilde{u}_n(x)\|_2 &= \left[\sum_{s=0}^{\infty} \sum_{k=0}^{\infty} \frac{\alpha_s \alpha_k}{s+k+1} + \sum_{i=0}^n \sum_{j=0}^n \frac{\gamma_i \gamma_j}{i+j+1} \right] \\ &\quad (b^{i+j+1} - a^{i+j+1}) \\ &\quad - 2 \left[\sum_{i=0}^n \sum_{j=0}^n \frac{\gamma_i \gamma_j}{i+j+1} (b^{i+j+1} - a^{i+j+1}) \right]^{\frac{1}{2}} \\ &\quad \times \left[\sum_{s=0}^{\infty} \sum_{k=0}^{\infty} \frac{\alpha_s \alpha_k}{s+k+1} (b^{s+k+1} - a^{s+k+1}) \right]^{\frac{1}{2}} = 0. \end{aligned} \quad (43)$$

The goal now is to estimate the error of interpolation. We Sample the total error of the approximation by $\varepsilon_n(x)$. Thus, $\varepsilon_n(x) = \|u(x) - \tilde{u}_n(x)\|_2$, where $\|\cdot\|_2$ denotes the Euclidean norm in \mathbb{R}^2 . \square

Theorem 4.2 Let \mathbb{L} be a compact linear bounded operator with a weakly singular kernel defined on the Banach space $X \rightarrow X$, where $\mathbb{X} = L^2[a, b]$, such that

$$\mathbb{L}(u(t)) = \int_a^b \frac{u(t)}{|x-t|^\alpha} dt; \quad 0 < \alpha < 1, a \leq x \leq b, \quad (44)$$

and

$$\mathbb{L}(\tilde{u}_n(t)) = \int_a^b \frac{\tilde{u}_n(t)}{|x-t|^\alpha} dt; \quad 0 < \alpha < 1, a \leq x \leq b. \quad (45)$$

Then,

$$\varepsilon_n = \|u(x) - \tilde{u}_n(x)\| = 0. \quad (46)$$

Proof Substituting Eqs. (44) and (45) into Eq. (6) we get

$$u(x) = \varphi(x) + \mathbb{L}(u(t)), \tilde{u}_n(x) = \varphi(x) + \mathbb{L}(\tilde{u}_n(t)). \quad (47)$$

Thus, we have

$$\begin{aligned} \varepsilon_n &= \|u(x) - \tilde{u}_n(x)\|_\Theta = \|\mathbb{L}(u(t)) - \mathbb{L}(\tilde{u}_n(t))\| \\ &= \left\| \int_a^b \frac{1}{|x-t|^\alpha} u(t) dt - \int_a^b \frac{1}{|x-t|^\alpha} \tilde{u}_n(t) dt \right\| \\ &= \left[\int_a^b \left| \int_a^b \frac{1}{|x-t|^\alpha} u(t) dt - \int_a^b \frac{1}{|x-t|^\alpha} \tilde{u}_n(t) dt \right|^2 dx \right]^{\frac{1}{2}} \\ &= \left[\int_a^b \left| \int_a^b \frac{u(t)}{|x-t|^\alpha} dt - \sum_{i=0}^n \tilde{\gamma}_i x^i \right|^2 dx \right]^{\frac{1}{2}}. \end{aligned} \quad (48)$$

Here, we have

$$\int_a^b \frac{1}{|x-t|^\alpha} \tilde{u}_n(t) dt = \int_a^b X(x) A K B X^T(t) X(t) \Omega dt = X(x) A K B H \Omega = X(x) \tilde{\Omega}, \quad (49)$$

where

$$\tilde{\Omega} = A K B H \Omega = [\tilde{\gamma}_i]_{i=0}^n, \quad (50)$$

and

$$\int_a^b \frac{1}{|x-t|^\alpha} u(t) dt = \int_a^b X(x) F X^T(t) \tilde{X}(t) \tilde{\Delta} dt = \int_a^b X(x) F Z(t) \tilde{\Delta} dt = X(x) F \tilde{Z} \tilde{\Delta} = X(x) E, \quad (51)$$

$$F = AKB = [f_{ij}]_{i,j=0}^n, \quad E = \tilde{Z} \tilde{\Delta}, \quad Z(t) = X^T(t) \tilde{X}(t); \quad \tilde{X}(t) = \lim_{m \rightarrow \infty} [t^q]_{q=0}^m, \quad (52)$$

$$\tilde{\Delta} = \lim_{m \rightarrow \infty} [\alpha_q]_{q=0}^m,$$

and

$$\begin{aligned} \tilde{Z} &= \int_a^b Z(t) dt = \lim_{m \rightarrow \infty} [z_{iq}]_{i,q=0}^{n,m}; \quad z_{iq} = \int_a^b t^{i+q} dt \\ &= \frac{b^{i+q+1} - a^{i+q+1}}{i+q+1}; \quad i = \overline{0, n}, \quad q = 0, 1, 2, \dots \end{aligned} \quad (53)$$

Let $E - \tilde{\Omega}$ be a column matrix with entries denoted by λ_i . Then, we have proven that

$$\lim_{n \rightarrow \infty} \{i=0\}^n \{\gamma_i\} x^i$$

$$\begin{aligned} \varepsilon_n &= \|u(x) - \tilde{u}_n(x)\|_{\Theta} = \left\| \int_a^b \frac{1}{|x-t|^\alpha} u(t) dt - \int_a^b \frac{1}{|x-t|^\alpha} \tilde{u}_n(t) dt \right\|^2 dx \Bigg|^{\frac{1}{2}} \\ &= \left[\int_a^b |X(x)(E - \tilde{\Omega})|^2 dx \right]^{\frac{1}{2}} = \lim_{m \rightarrow \infty} \left[\sum_{i=0}^n \sum_{q=0}^m \frac{\lambda_i \lambda_j}{i+q+1} (b^{i+q+1} - a^{i+q+1}) \right]^{\frac{1}{2}} = 0. \end{aligned} \quad (54)$$

Computational results and discussions

Based on the two presented techniques, we designed two MATLAB R2019b codes for the solution of four weakly singular Fredholm integral equations of the second kind. We find the interpolant solutions for the six examples by applying the two given techniques and compare the obtained results with the exact solutions. The obtained interpolated solutions to examples 1–4 strongly converge to the exact ones faster than the methods mentioned in [20, 21]. Moreover, the interpolation is found easily and uniformly rather than these complicated results that are mentioned in [20, 21], as shown in the given tables and figures that are superior. The obtained interpolate solutions to the examples 5–6 for nonsingular equations are found equal to the exact solution. We denoted the exact solution by $u_{ex}(x)$ and the interpolant solutions obtained by using the first and second techniques are denoted by $\tilde{u}_n^1(x)$ and $\tilde{u}_n^2(x)$, respectively, where n denotes the interpolant degree.

Example 1 Consider the integral equation,

Table 1 The absolute errors $R_n^1(x_i)$ for $n = 2, 5$

x_i	$R_2^1(x_i)$	$R_5^1(x_i)$
0	0.00359	0.0021782
0.1	0.00359	0.0021782
0.2	0.00359	0.0021782
0.3	0.00359	0.0021782
0.4	0.00359	0.0021782
0.5	0.00359	0.0021782
0.6	0.00359	0.0021782
0.7	0.00359	0.0021782
0.8	0.00359	0.0021782
0.9	0.00359	0.0021782
1.0	0.0036	0.0021782

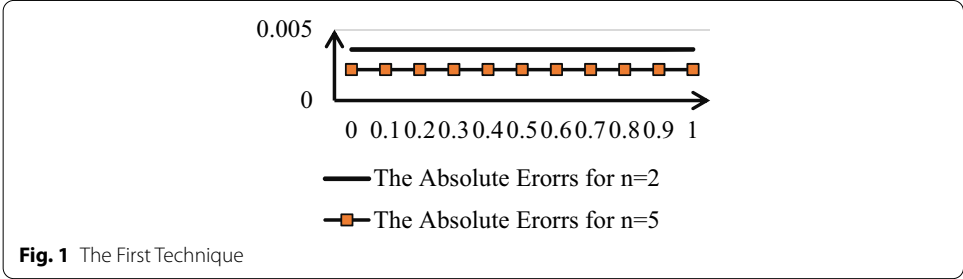


Table 2 The absolute errors $R_n^2(x_i)$ for $n = 4, 6$

x_i	$R_4^2(x_i)$	$R_6^2(x_i)$
0	0.10881	0.004659
0.1	0.10881	0.004659
0.2	0.10881	0.004659
0.3	0.10881	0.004659
0.4	0.10881	0.00466
0.5	0.10881	0.00466
0.6	0.10881	0.00466
0.7	0.10881	0.00466
0.8	0.10881	0.00466
0.9	0.10881	0.00466
1.0	0.10881	0.00466

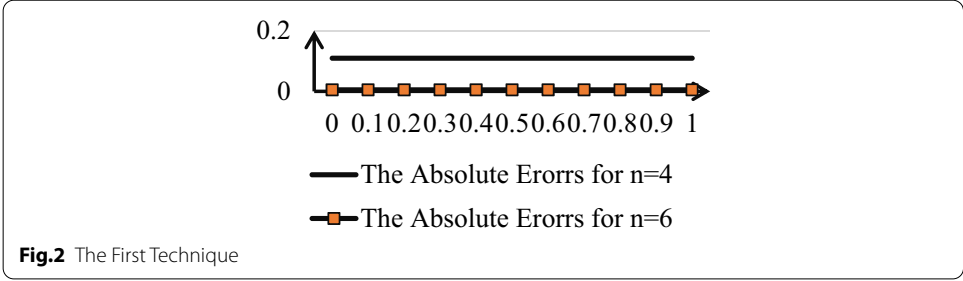
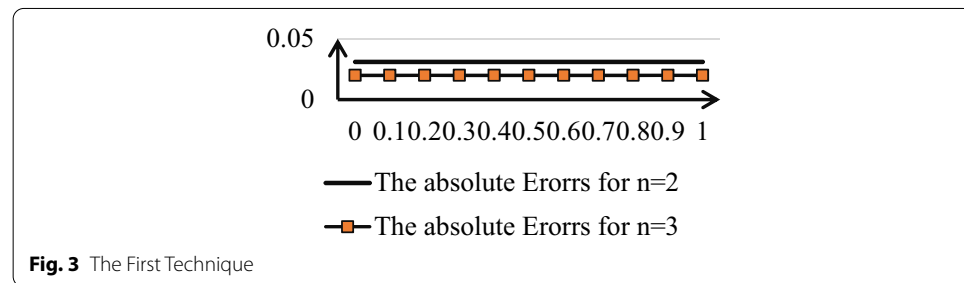


Table 3 The absolute errors $R_n^1(x_i)$ for $n = 2, 3$

x_i	$R_2^1(x_i)$	$R_3^1(x_i)$
0	0.030966	0.019942
0.1	0.030966	0.01994
0.2	0.030966	0.01995
0.3	0.030966	0.01994
0.4	0.030966	0.01994
0.5	0.030966	0.01994
0.6	0.030966	0.01994
0.7	0.030966	0.01994
0.8	0.030966	0.01994
0.9	0.030966	0.01995
1	0.030966	0.0199



$$u(x) = x^2 - \frac{16}{15} + \int_0^1 \frac{u(t)}{\sqrt{1-t}} dt; \quad 0 \leq x \leq 1, \quad (55)$$

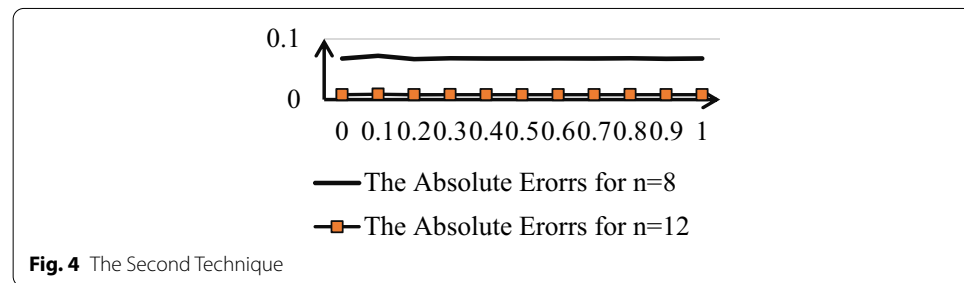
whose exact solution is given by $u_{ex}(x) = x^2$ [20]. Using the first technique for $\delta_1 = \delta_2 = 0$ with $x = 1$ in the kernel, we obtained uniformly interpolated polynomials $\tilde{u}_n^1(x)$ for $n = 2, 5$. The CPU time for $n = 2$ was 8.294 s and for $n = 5$ was 12.893 s. We evaluated the exact solution values $u_{ex}(x_i)$ and $\tilde{u}_n^1(x_i)$ for $n = 2, 5$ at the set of nodes $x_i = 0 : 0.1 : 1.0$ and then estimated the absolute errors $R_n^1(x_i) = |u_{ex}(x_i) - \tilde{u}_n^1(x_i)|$ for $n = 2, 5$ as shown in Table 1. In Fig. 1, plotted are the graphs of absolute errors $R_n^1(x_i)$ for $n = 2, 5$. Using the second technique for $n = 4, 6$ and $\delta_1 = 0$, $\delta_2 = 1/150$, we obtained the uniformly interpolated polynomials $\tilde{u}_4^2(x), \tilde{u}_6^2(x)$. The total CPU time for $n = 4$ was 12.356 s and for $n = 6$ was 18.000 s. We evaluated $u_{ex}(x_i)$ and $\tilde{u}_n^2(x_i)$ for $n = 4, 6$ at the set of nodes $x_i = 0 : 0.1 : 1.0$ and hence estimated the absolute errors $R_n^2(x_i) = |u_{ex}(x_i) - \tilde{u}_n^2(x_i)|$ for $n = 4, 6$ as shown in Table 2. In Fig. 2, plotted are the graphs of the absolute errors $R_n^2(x_i)$ for $n = 4, 6$.

Example 2 Consider the integral equation,

$$u(x) = \sqrt{x} - \frac{\pi}{2} + \int_0^1 \frac{u(t)}{\sqrt{1-t}} dt; \quad 0 \leq x \leq 1, \quad (56)$$

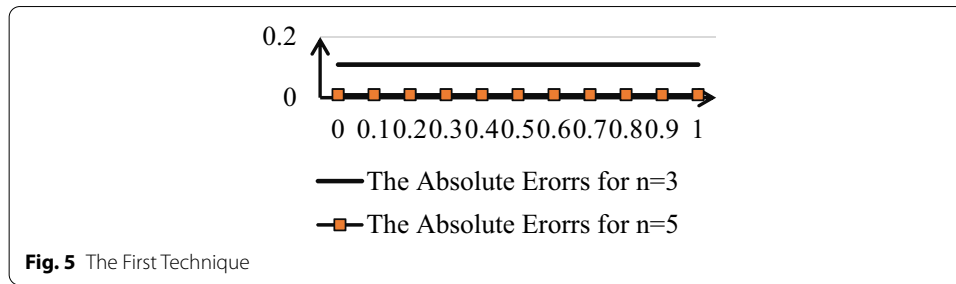
Table 4 The absolute errors $R_n^2(x_i)$ for $n = 8, 12$

x_i	$R_8^2(x_i)$	$R_{12}^2(x_i)$
0	0.067624	0.00799
0.1	0.072018	0.00876
0.2	0.066423	0.00789
0.3	0.06801	0.00801
0.4	0.067507	0.00799
0.5	0.067624	0.00799
0.6	0.067704	0.00800
0.7	0.067452	0.00798
0.8	0.067949	0.00802
0.9	0.067051	0.00790
1	0.067624	0.00799

**Table 5** The absolute errors $R_n^1(x_i)$ for $n = 3, 5$

x_i	$R_3^1(x_i)$	$R_5^1(x_i)$
0	0.10872	0.0095
0.1	0.10872	0.0095
0.2	0.10872	0.0095
0.3	0.10872	0.0095
0.4	0.10872	0.0095
0.5	0.10872	0.0095
0.6	0.10872	0.0095
0.7	0.10872	0.0095
0.8	0.10872	0.0095
0.9	0.10872	0.0095
1.0	0.10872	0.0095

whose exact solution is given by $u_{ex}(x) = \sqrt{x}$ [20]. Using the first technique for $n = 2, 3$, and $\delta_1 = \delta_2 = 0$ while $x = 1$ in the kernel, we obtained the uniformly interpolated polynomials $\tilde{u}_2^1(x), \tilde{u}_3^1(x)$. The CPU total time for $n = 2$ was 8.282 s and for $n = 3$ was 9.013 s. We evaluated the exact solution values $u_{ex}(x_i)$ and the interpolated polynomials $\tilde{u}_2^1(x_i), \tilde{u}_3^1(x_i)$ at the set of nodes $x_i = 0 : 0.1 : 1.0$ and hence estimated the absolute errors $R_n^1(x_i) = |u_{ex}(x_i) - \tilde{u}_n^1(x_i)|$ for $n = 2, 3$ as shown in Table 3. In Fig. 3, plotted are the graphs of the absolute errors $R_n^1(x_i)$ for $n = 2, 3$. Using the second technique for $n = 8, 12$ and $\delta_1 = 0, \delta_2 = 1/300$, we obtained the uniformly interpolated polynomials



$\tilde{u}_8^2(x)$, $\tilde{u}_{12}^2(x)$. The CPU total time for $n = 8$ was 24.775 s and for $n = 12$ was 44.715 s. We evaluated the exact solution values $u_{ex}(x_i)$ and $\tilde{u}_8^2(x_i)$, $\tilde{u}_{12}^2(x_i)$ at the set of nodes $x_i = 0 : 0.1 : 1.0$ and hence estimated the absolute errors $R_n^2(x_i) = |u_{ex}(x_i) - \tilde{u}_n^2(x_i)|$ for $n = 8, 12$ as shown in Table 4. In Fig. 4, plotted are the graphs the absolute errors $R_n^2(x_i)$ for $n = 8, 12$.

Example 3 Consider the integral equation,

$$u(x) = e^x - 4.0602 + \int_0^1 \frac{u(t)}{\sqrt{1-t}} dt; \quad 0 \leq x \leq 1, \quad (57)$$

whose exact solution is given by $u_{ex}(x) = e^x$ [20]. Using the first technique for $\delta_1 = \delta_2 = 0$ with $x = 1$ in the kernel, we obtain $\tilde{u}_n^1(x)$ for $n = 3, 5$. The CPU total time for $n = 3$ was 8.882 s and for $n = 5$ was 11.830 s. We evaluated the exact solution values

Table 6 The absolute errors $R_n^2(x_i)$ for $n = 6, 8$

x_i	$R_6^2(x_i)$	$R_8^2(x_i)$
0	0.28768	0.095222
0.1	0.28768	0.095222
0.2	0.28768	0.095222
0.3	0.28768	0.095222
0.4	0.28768	0.095222
0.5	0.28768	0.095222
0.6	0.28768	0.095222
0.7	0.28768	0.095222
0.8	0.28768	0.095222
0.9	0.28768	0.095222
1.0	0.28768	0.095222

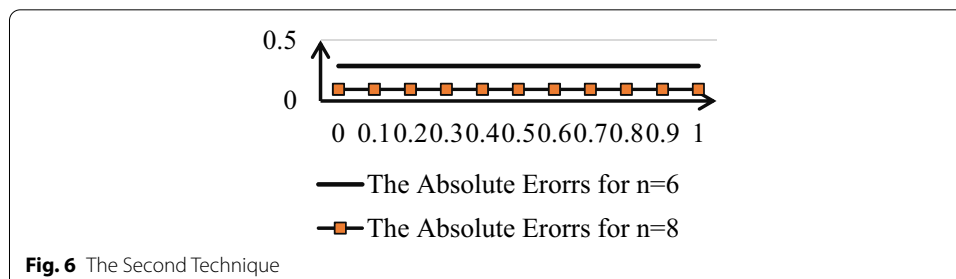
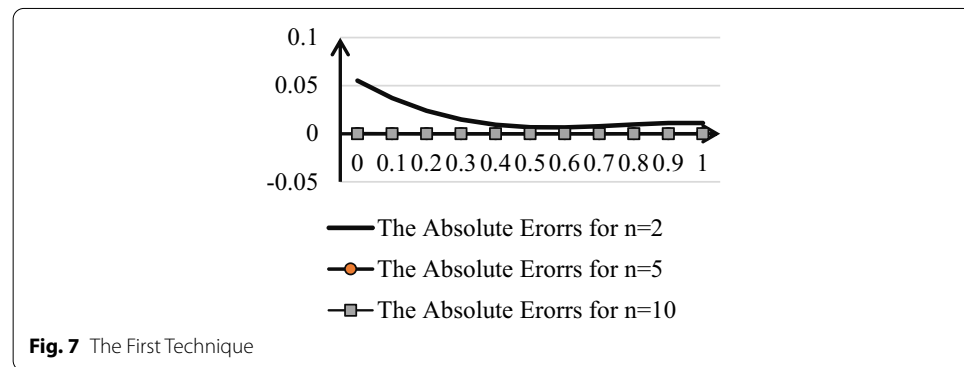


Table 7 The absolute errors $R_2^1(x_i)$ and $R_5^1(x_i)$

x_i	$R_2^1(x_i)$	$R_5^1(x_i)$
0	0.057185	0.79416
0.1	0.053333	0.44121
0.2	0.050139	0.23172
0.3	0.047339	0.11599
0.4	0.04469	0.05745
0.5	0.04202	0.030668
0.6	0.039271	0.019345
0.7	0.036508	0.014296
0.8	0.033892	0.011388
0.9	0.03167	0.009474
1	0.030107	0.008288



$u_{ex}(x_i)$ and the uniformly interpolated polynomials $\tilde{u}_3^1(x_i)$, $\tilde{u}_5^1(x_i)$ at the set of nodes $x_i = 0 : 0.1 : 1.0$ and hence estimated the absolute errors $R_n^1(x_i) = |u_{ex}(x_i) - \tilde{u}_n^1(x_i)|$ for $n = 3, 5$ as shown in Table 5. In Fig. 5, plotted are the graphs of the absolute errors $R_n^1(x_i)$ for $n = 3, 5$. Using the second technique for $\delta_1 = 0, \delta_2 = 1/250$, we obtain the uniformly interpolated polynomials $\tilde{u}_n^2(x)$ for $n = 6, 8$. The CPU total time for $n = 6$ was 15.317 s and for $n = 8$ 22.591 s. We evaluated the exact solution values $u_{ex}(x_i)$ and $\tilde{u}_6^2(x_i)$, $\tilde{u}_8^2(x_i)$ at the set of nodes $x_i = 0 : 0.1 : 1.0$ and hence estimated the absolute errors $R_n^2(x_i) = |u_{ex}(x_i) - \tilde{u}_n^2(x_i)|$ for $n = 6, 8$ as shown in Table 6. In Fig. 6, plotted are the graphs of the absolute errors $R_n^2(x_i)$ for $n = 6, 8$.

Example 4 Consider the integral equation,

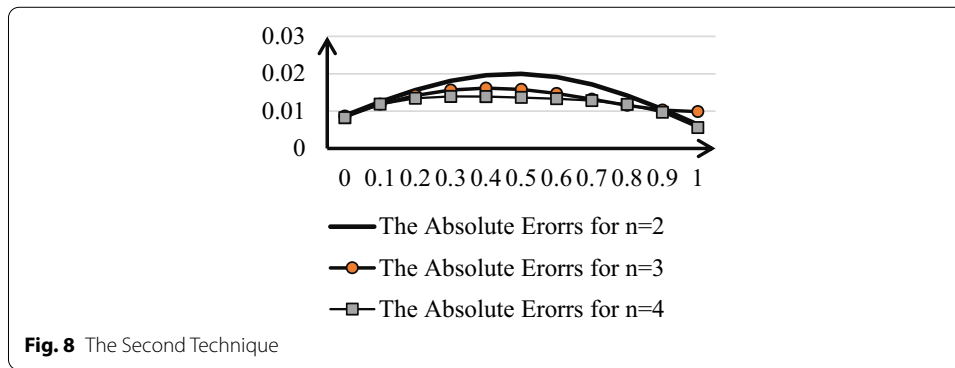
$$u(x) = f(x) + \frac{1}{10} \int_0^1 |x - t|^{-\frac{1}{3}} u(t) dt; \quad 0 \leq x \leq 1, \quad (58)$$

where

$$f(x) = x^2(1 - x^2) - \frac{27}{30800} \left[x^{8/3} (54x^2 - 126x + 77) + (1 - x)^{8/3} (54x^2 + 18x + 5) \right], \quad (59)$$

Table 8 The absolute errors $R_n^2(x_i)$ for $n = 2, 3, 4$

x_i	$R_2^2(x_i)$	$R_3^2(x_i)$	$R_4^2(x_i)$
0	0.008776	0.008654	8.22E-03
0.1	0.012442	0.011837	1.18E-02
0.2	0.015643	0.014181	1.34E-02
0.3	0.018116	0.015636	1.39E-02
0.4	0.019614	0.016163	1.39E-02
0.5	0.019973	0.015809	1.36E-02
0.6	0.019133	0.014724	1.33E-02
0.7	0.017153	0.013177	1.28E-02
0.8	0.014199	0.011545	1.18E-02
0.9	0.010516	0.010283	9.68E-03
1	0.00637	0.009866	5.57E-03



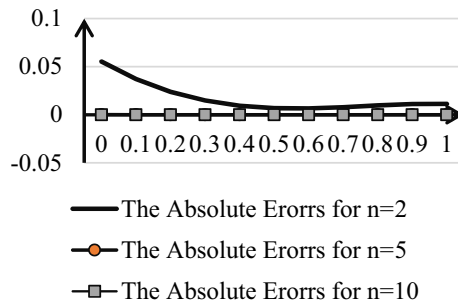
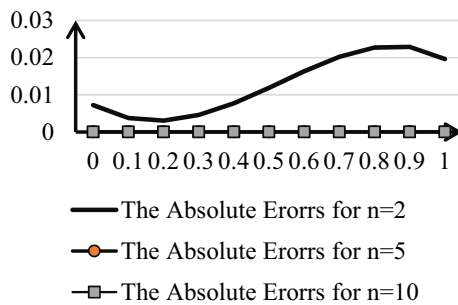
whose exact solution is given by $u_{ex}(x) = x^2(1-x)^2$ [21]. Using the first technique for $n = 2, 5$ and $\delta_1 = 0, \delta_2 = 1/5$, we obtain $\tilde{u}_2^1(x), \tilde{u}_5^1(x)$. The CPU total time for $n = 2$ was 9.778s and for $n = 5$ was 15.501s. We evaluated the exact solution values $u_{ex}(x_i)$ and the interpolated solution values $\tilde{u}_2^1(x_i), \tilde{u}_5^1(x_i)$ at the set of nodes $x_i = 0 : 0.1 : 1.0$ and hence estimated the absolute errors $R_n^1(x_i) = |u_{ex}(x_i) - \tilde{u}_n^1(x_i)|$ for $n = 2, 5$ as shown in Table 7. In Fig. 7, plotted are the graphs of the absolute errors $R_n^1(x_i)$ for $n = 2, 5$. Using the second technique for $n = 2, 3, 4$ and $\delta_1 = 1/15, \delta_2 = 0$, we obtained $\tilde{u}_2^2(x_i), \tilde{u}_3^2(x_i), \tilde{u}_4^2(x_i)$ at the set of nodes $x_i = 0 : 0.1 : 1.0$. The CPU total time for $n = 2$ was 10.331 s, for $n = 3$ was 11.713 s, and for $n = 4$ was 15.699 s. Table 8, contains the absolute errors $R_n^2(x_i) = |u_{ex}(x_i) - \tilde{u}_n^2(x_i)|$ for $n = 2, 3, 4$ at the set of nodes $x_i = 0 : 0.1 : 1.0$. In Fig. 8, plotted are the graphs of the absolute errors $R_2^2(x_i), R_3^2(x_i)$ and $R_4^2(x_i)$ at $x_i = 0 : 0.1 : 1.0$.

Example 5 Consider the nonsingular Fredholm integral equation of the second kind.

$$u(x) = e^{-x} + \int_0^1 e^{x+t} u(t) dt; \quad 0 \leq x \leq 1 \quad (60)$$

Table 9 The absolute errors $R_n^1(x_i)$ for $n = 2, 5, 10$

x_i	$R_2^1(x_i)$	$R_5^1(x_i)$	$R_{10}^1(x_i)$
0	0.00359	0.000164	9.54E-06
0.1	0.00359	5.27E-05	8.34E-07
0.2	0.00359	1.31E-06	-6.85E-07
0.3	0.00359	1.99E-05	4.10E-08
0.4	0.00359	2.81E-05	5.85E-08
0.5	0.00359	3.21E-05	1.37E-07
0.6	0.00359	3.54E-05	2.68E-07
0.7	0.00359	3.92E-05	4.43E-07
0.8	0.00359	4.33E-05	6.28E-07
0.9	0.00359	4.78E-05	7.53E-07
1.0	0.0036	5.29E-05	7.20E-07

**Fig. 9** The First Technique**Fig. 10** The Second Technique

whose exact solution is given by $u_{ex}(x) = e^{-x} + \frac{2e^x}{3-e^2}$ [35]. Using the first technique for $\delta_1 = \delta_2 = 0$, we obtained uniformly interpolated polynomials $\tilde{u}_n^1(x)$ for $n = 2, 5, 10$. The CPU time for $n = 2$ was 7.987 s and for $n = 5$ was 12.859 s, and for $n = 10$ was 36.077 s. We evaluated the exact solution values $u_{ex}(x_i)$ and $\tilde{u}_n^1(x_i)$ for $n = 2, 5, 10$ at the set of nodes $x_i = 0 : 0.1 : 1.0$ and then estimated the absolute errors $R_n^1(x_i) = |u_{ex}(x_i) - \tilde{u}_n^1(x_i)|$ for $n = 2, 5, 10$ as shown in Table 9. In Fig. 9, plotted are the graphs of absolute errors $R_n^1(x_i)$ for $n = 2, 5, 10$. Using the second technique for $n = 2, 5, 10$ and $\delta_1 = \delta_2 = 0$, we obtained the uniformly interpolated polynomials $\tilde{u}_n^2(x)$ for $n = 2, 5, 10$. The CPU total

Table 10 The absolute errors $R_n^2(x_i)$ for $n = 4, 6$

x_i	$R_2^2(x_i)$	$R_{10}^2(x_i)$	$R_{15}^2(x_i)$
0	0.007216	2.4E-05	1.61E-11
0.1	0.003766	2.75E-05	1.77E-11
0.2	0.003067	2.93E-05	2.01E-11
0.3	0.00459	3.21E-05	2.18E-11
0.4	0.007749	3.58E-05	2.74E-11
0.5	0.011897	3.98E-05	3.69E-11
0.6	0.016318	4.37E-05	5.78E-11
0.7	0.020221	4.8E-05	1.04E-10
0.8	0.022733	5.34E-05	2.02E-10
0.9	0.022887	6.01E-05	4.03E-10
1.0	0.019615	6.52E-05	8.01E-10

time for $n = 2$ was 8.086 s, for $n = 5$ was 13.579 s and for $n = 10$ was 31.443 s. We evaluated $u_{ex}(x_i)$ and $\tilde{u}_n^2(x_i)$ for $n = 4, 6$ at the set of nodes $x_i = 0 : 0.1 : 1.0$ and hence estimated the absolute errors $R_n^2(x_i) = |u_{ex}(x_i) - \tilde{u}_n^2(x_i)|$ for $n = 4, 6$ as shown in Table 2. In Fig. 2, plotted are the graphs of the absolute errors $R_n^2(x_i)$ for $n = 4, 6$. The generated interpolated solutions strongly converge to the exact ones when employing either the first or second technique (Table 10, Fig. 10).

Example 6 Consider the nonsingular Fredholm integral equation of the second kind.

$$u(x) = x + \int_{-1}^1 (x^4 - t^4) u(t) dt; \quad -1 \leq x \leq 1 \quad (61)$$

whose exact solution is given by $u_{ex}(x) = x$ [36]. Using formula (10) of the first technique for $\delta_1 = \delta_2 = 0$, we obtained the interpolated polynomials $\tilde{u}_n^1(x)$ for $n = 5, 6$. The CPU time for $n = 5$ was 11.474 s and for $n = 6$ was 11.777 s. We evaluated the exact solution values $u_{ex}(x_i)$ and $\tilde{u}_n^1(x_i)$ for $n = 5, 6$ at the set of nodes $x_i = -1 : 0.2 : 1.0$ and then estimated the absolute errors. It turns out that $R_n^1(x_i) = |u_{ex}(x_i) - \tilde{u}_n^1(x_i)| = 0$ for $n = 5, 6$. Using the formulas (26) and (27) of the second technique for $\delta_1 = \delta_2 = 0$, we obtained the interpolated polynomials $\tilde{u}_n^2(x)$ for $n \geq 2$ exactly equal to the exact solution. The CPU total time for $n = 2$ was 9.659 s.

Conclusion

We modified the traditional barycentric Lagrange interpolation formula and expressed it as a product of four matrices; one of which is the monomial basis function matrix. Based on this advanced formula, we presented two techniques for finding the interpolate solutions of weakly singular Fredholm integral equations of the second kind. The kernel is interpolated twice with respect to both variables and thus has been expressed via five matrices. The advantage of the presented techniques is that we can isolate the singularity of the kernel and easily find the interpolant solution in matrix form without applying the collocation method. The most important advantage lies in the idea of the given rules

for choosing two different sets of interpolation nodes associated with the kernel's two variables so that the square root of the kernel remains greater than zero and has non-imaginary values. Thus, the singularity is completely removed. The convergence in the mean and the error norm estimation are studied. The interpolate solutions of the first four illustrated examples to weakly singular equations are found strongly converge uniformly to the exact ones. The convergence of the obtained solutions is faster than those obtained by other cited methods. The interpolate solutions of the fifth and sixth examples to nonsingular equations are found equal to the exact ones. Thus, the efficiency and genuineness of the given method are confirmed.

Acknowledgements

We express our gratitude to the anonymous referees for their constructive reviews of the manuscript and for helpful comments.

Authors' contributions

SES developed the theoretical formalism, Both NS and AYS performed the analytic calculations and performed the numerical simulations and contributed to the final version of the manuscript. All authors read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

Not applicable.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

¹Faculty of Electronic Engineering, Menofia University, Cairo, Egypt. ²Faculty of Energy and Environmental Engineering, British University, Cairo, Egypt. ³Physics and Engineering Mathematics Department, Faculty of Engineering at Mataria, Helwan University, Cairo, Egypt.

Received: 22 June 2021 Accepted: 21 November 2021

Published online: 08 December 2021

References

- Kendall EA. The Numerical Solution of Integral Equations of the Second Kind. Cambridge: Cambridge University Press; 2010.
- Farhad SHL, Reza G, Vladimir IO. Novel single-source surface integral equation for scattering problems by 3-D dielectric objects. *IEEE Trans Antennas Propagation*. 2018;66(2):797–807.
- Adrian SB, Andriulli FP, Eibert TF. On a refinement-free Calderón multiplicative preconditioner for the electric field integral equation. *Physics*. 2019;376:1232–52.
- Shoukralla ES. Numerical Solution of Helmholtz Equation for an Open Boundary in Space. *J Appl Math Modeling*. 1997;21:231–2.
- Qin SL, Sheng S, Weng CC. A Potential-Based Integral Equation Method for Low-Frequency Electromagnetic Problems. *IEEE Trans Antennas Propag*. 2018;66(3):1413–26.
- Dmitriev VI, Dmitrieva IV, Osokin NA. Solution of an Integral Equation of the first Kind with Logarithmic Kernel. *Comput Math Model*. 2018;29(3):307–18.
- Shoukralla ES. A Technique for the Solution of a Certain Singular Integral Equation of The First Kind. *Int J Computer Math*. 1998;69:165–73.
- Shoukralla ES. Approximate solution to weakly singular integral equations. *J Appl Math Modeling*. 1996;20:800–3.
- Shoukralla ES. A Numerical Method for Solving Fredholm Integral Equations of the First Kind with Logarithmic Kernels and Singular Unknown Functions. *J Appl Comput Math*. 2020;6:172.
- Shoukralla ES. Application of Chebyshev Polynomials of the Second Kind to the Numerical Solution of Weakly Singular Fredholm Integral Equations of the First Kind. *IAENG Int J Appl Math*. 2021;51:8.
- Shoukralla ES, Markos MA. The economized monic Chebyshev polynomials for solving weakly singular Fredholm integral equations of the first kind. *Asian-Eur J Math*. 2020;12:1.
- Shoukralla ES, Markos MA. Numerical Solution of a Certain Class of Singular Fredholm Integral Equations of the First Kind via the Vandermonde Matrix. *Int J Math Models Methods Appl Sci*. 2020;14:48–53.

13. Shoukralla ES, Kamel M, Markos MA. A new computational method for solving weakly singular Fredholm integral equations of the first kind. In: 13th IEEE International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt, IEEE Xplore; 2018.
14. Reza B, Emran T, Faezeh T. Numerical solution of weakly singular Fredholm integral equations via generalization of the Euler-Maclaurin summation Formula. *J Taibah Univ Sci.* 2014;8:199–205.
15. Yang Y, Tang Z, Huang Y. Numerical Solutions for Fredholm Integral Equations of The Second Kind with Weakly Singular Kernel Using Spectral Collocation Method. *Appl Math Computation.* 2019;349:314–24.
16. Azizallah A, Mahmoud P. Reproducing kernel method for a class of weakly singular Fredholm integral equations". *J Taibah Univ Sci.* 2018;12(4):409–14.
17. Bijaya LP, Moumita M, Gnaneshwar N. Legendre Multi-Galerkin Methods for Fredholm Integral Equations with Weakly Singular Kernel and the Corresponding Eigenvalue Problem *J. Comput Appl Math.* 2019;346:224–36.
18. Guebbah H. "Regularization and Fourier Series for Fredholm Integral Equations of the Second Kind with a Weakly Singular Kernel. *Numer Funct Anal Optim.* 2018;39(1):1–10.
19. Filmomena D, Rosario F. Projection Methods Based on Grids for Weakly Singular Integral Equations. *Appl Numer Math.* 2017;114:47–54.
20. Boichuk OA, Feruk VA. Linear Boundary-Value Problems for Weakly Singular Integral Equations. *J Math Sci.* 2020;247:2.
21. Behera S, Saha Ray S. Euler wavelets method for solving fractional-order linear Volterra-Fredholm integro-differential equations with weakly singular kernels. *Comput Appl Math.* 2021;40:192.
22. Ali S, Taher L, Tofigh A, Mahmoud P. an effective collocation technique to solve the singular Fredholm integral equations with Cauchy kernel. *Adv Diff Eq.* 2017;280:2.
23. Rehman S, Pedas A, Vainikko G. Fast solvers of weakly singular integral equations of the second kind. *Math Model Anal.* 2018;23(4):639–64.
24. Shoukralla ES. Interpolation method for solving weakly singular integral equations of the second kind. *Appl Comput Math.* 2021;10(3):76–85.
25. Shoukralla ES. Interpolation Method for Evaluating Weakly Singular Kernels. *J Math Comput Sci.* 2021;11(6):7487–510.
26. Shoukralla ES, Elgohary H, Ahmed BM. Barycentric Lagrange interpolation for solving Volterra integral equations of the second kind. *J Phys England.* 2020;1447:012002.
27. Shoukralla ES, Ahmed BM. Numerical Solutions of Volterra Integral Equations of the Second Kind using Lagrange interpolation via the Vandermonde matrix. *J Phys.* 2020;1447:012003.
28. Shoukralla ES, Ahmed BM. Multi-techniques method for Solving Volterra Integral Equations of the Second Kind. In: 14th International Conference on Computer Engineering and Systems (ICCES). IEEE. 2019.
29. The Barycentric Lagrange Interpolation via Maclaurin Polynomials for Solving the Second Kind Volterra Integral Equations. In: IEEE 15th International Conf. on Computer Engineering and Systems (ICCES 2020), Cairo, Egypt.
30. Berrut J-P, Trefethen LN. Barycentric Lagrange Interpolation. *Soc Indus Appl Math.* 2004;46(3):501–17.
31. Nicholas J. The numerical stability of Barycentric Lagrange interpolation. *IMA J Numer Anal.* 2004;24:547–56.
32. Gander W. Change of basis in polynomial interpolation. *Numer Linear Algebra Appl.* 2005;12:769–78.
33. Cvetkovski Z. Inequalities, theorems, techniques, and selected problems. Berlin: Springer; 2012.
34. Fabio SB. Real Analysis and Application. Berlin: Springer; 2018.
35. Shoukralla ES, Elgohary H, Morgan M. Shifted Legendre Polynomials for Solving Second Kind Fredholm Integral Equations. *Menoufia J Elect Eng Res (MJEER).* 2021;30(1):76–83.
36. Shoukralla ES, El-Serafi SA, Elgohary H, Morgan M. A Computational Method for Solving Fredholm Integral Equations of the Second Kind. *Comput Methods.* 2019;28:280–5.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)