https://doi.org/10.31489/2025M1/155-169 *Research article*

Solving Volterra-Fredholm integral equations by non-polynomial spline function based on weighted residual methods

S.H. Salim[1], R.K. Saeed[2], K.H.F. Jwamer[3,*]

[1] *College of Basic Education, University of Sulaimani, Sulaymaniyah, Iraq;*
[2] *College of Science, Salahaddin University-Erbil, Erbil, Kurdistan Region, Iraq;*
[3] *College of Science, University of Sulaimani, Sulaymaniyah, Iraq*
(E-mail: sarfraz.salim@univsul.edu.iq, rostam.saeed@su.edu.krd, karwan.jwamer@univsul.edu.iq)

In this paper, a method that utilizes a non-polynomial spline function based on the weighted residual technique to approximate solutions for linear Volterra-Fredholm integral equations is presented. The approach begins with the selection of a series of knots along the integration interval. We then create a set of basis functions, defined as non-polynomial spline functions, between each pair of adjacent knots. The unknown function is expressed as a linear combination of these basis functions to approximate the solution of integral equations. The coefficients of the spline function are calculated by solving a system of linear equations derived from substituting the spline approximation into the integral equation while maintaining continuity and smoothness at the knots. Non-polynomial splines are beneficial for approximating functions with complex shapes and for solving integral equations with non-smooth kernels. However, the solution's accuracy significantly relies on the selection of knots, and the method may require extensive computational resources for large systems. To illustrate the effectiveness of the method, three examples are presented, implemented using Python version 3.9. The paper also addresses the error analysis theorem relevant to the proposed non-polynomial spline function.

Keywords: Volterra integral equation, Fredholm integral equation, non-polynomial spline, weighted residual methods.

2020 Mathematics Subject Classification: 45B05, 45B05, 41A15.

Introduction

Volterra-Fredholm integral equations play a crucial role in mathematical modeling in numerous scientific and engineering disciplines, such as physics, biology, and finance. These equations, which involve both Volterra and Fredholm integral operators, often arise in the analysis of intricate systems where time-dependent and spatially distributed processes are interrelated. However, despite their significance, solving Volterra-Fredholm integral equations presents challenges due to their complexity and the presence of mixed integral terms. For additional information, refer to [1–5].

The literature presents Volterra-Fredholm integral equations in the following form:

$$u(x) = f(x) + \lambda_1 \int_a^x K(x,t)u(t)dt + \lambda_2 \int_a^b L(x,t)u(t)dt, \tag{1}$$

where the functions $f(x)$, and the kernels $K(x,t)$, $L(x,t)$ are known L^2 analytic functions and λ_1, λ_2, are arbitrary constants, x is variable and $u(x)$ is the unknown continuous function to be determined. These integral equations allow physicists to formulate and solve problems where traditional differential equations may not be applicable or are too complex to solve directly. Volterra-Fredholm integral equations are used to describe the time evolution of quantum systems. For example, they can model

*Corresponding author. *E-mail: karwan.jwamer@univsul.edu.iq*
Received: 15 July 2024; *Accepted:* 25 November 2024.

scattering processes and interactions between particles. In the study of electromagnetic fields, these equations can help solve problems related to wave propagation, radiation, and diffraction, especially in complex media. They are employed in the analysis of many-body systems, where the interactions between particles can be described by integral equations. Also, they are used in modelling fluid flow, especially in non-linear and time-dependent situations, in systems with memory effects. These equations can be used to model the behaviour of materials that exhibit viscous and elastic characteristics, which is crucial in material science and engineering applications.

Traditional numerical methods have been developed and used for solving Volterra-Fredholm integral equation, including the use of Touchard Polynomials, spline function, rational interpolations, etc. [3, 6–19]. In addition, non-polynomial spline functions are used to solve integral equations and differential equations [20–31]. Recently, Salim et al. [32–35] used linear, quadratic, and cubic spline functions to solve equation (1).

In this study, the authors combine non-polynomial spline functions with weighted residual methods, which minimize the residuals of the integral equations in a weighted manner, to develop a robust and efficient technique for solving the equation (1).

The structure of this paper is outlined as follows: Section 1 provides an overview of the weighted residual method and its various types. Section 2 describes our methodology in detail. Section 3 focuses on error analysis. Section 4 presents several numerical examples to illustrate the effectiveness of our technique. Finally, some tentative conclusions are given.

1 Weighted Residual Methods

The weighted residual methods [2,36–38] is a mathematical technique commonly used in numerical analysis and finite element analysis to solve partial differential equations (PDEs). The idea behind the method is to represent the solution of the PDE as a linear combination of a set of basis functions, and then to find the coefficients of the basis functions by minimizing the residual error. The residual error is defined as the difference between the exact solution of the given problem and the approximate solution obtained using the basis functions. The minimization is performed using a set of weighting functions, which give greater weight to certain regions of the domain where the solution is expected to be more important. The weighted residual method can be used to solve a wide range of PDEs, including elliptic, parabolic, and hyperbolic equations, different types of ordinary differential equations and integral equations. It is a very flexible method that can handle boundary conditions. We present these methods by considering the following residues $E(x)$ or $E(\overline{C_j}; x)$ depends on x as well as on the parameters $a_j,\ b_j,\ c_j,\ d_j,\ j = 0,\ 1,\ \ldots,\ n-1$. We define $E(x)$ as follows:

$$E(\overline{C_j}; x) = \overline{u_n}(x) - f(x) - \lambda_1 \int_a^x K(x,t)\overline{u_n}(t)dt - \lambda_2 \int_a^b L(x,t)\overline{u_n}(t)dt, \quad x \in D = [a,b]\,, \qquad (2)$$

for solving (1), where

$$\overline{u_n}\,(x) = \sum_{j=0}^{n-1} \overline{\alpha}_{ij}\phi_j(x),$$

for $i = 0,\ 1,\ 2,\ \ldots,\ n-1$ and D is a prescribed domain. It is obvious that when $E(x) = 0$, then the exact solution is obtained which is difficult to be achieved, therefore we shall try to minimize $E(x)$ is some sense. In the weighted residual method the unknown parameters are chosen to minimize the residual $E(x)$ setting its weighted integral equal to zero, i.e.

$$\int_D w_j E\,(x)\,dx = 0, \quad j = 0, 1, 2, \ldots, n-1, \qquad (3)$$

where w_j is prescribed weighting function, the technique based on equation (3) is called weighted residual method. Different choices of w_j yield different methods with different approximate solutions. Below we discuss some of the weighted residual method.

1.1 Collocation Method (CM)

It is a simple technique for obtaining an approximate solution of equation (1), the weight function w_j in equation (3) are defined as

$$w_j = \delta\left(x - x_j\right), \tag{4}$$

where the fixed points $x_j \in D$, $j = 0, 1, \ldots, n-1$ are called collocation points. Here Dirac's delta function $\delta(x - x_j)$ is defined as

$$\delta\left(x - x_j\right) = \begin{cases} 1, & \text{if} \quad x_j = x, \\ \\ 0, & \text{else.} \end{cases}$$

Inserting equation (4) in equation (3) gives

$$\int_D w_j E\left(x_j\right) dx = \int_D \delta\left(x - x_j\right) E\left(x_j\right) dx = \int_{x_j^-}^{x_j^+} \delta\left(x - x_j\right) E\left(x_j\right) dx =$$

$$= E\left(x_j\right) \int_{x_j^-}^{x_j^+} \delta\left(x - x_j\right) dx = E\left(x_j\right) = 0, \quad \text{for} \quad j = 0, 1, \ldots, n-1. \tag{5}$$

Equation (5) will provide us with n simultaneous equations in n unknowns to determine the parameters. Moreover, the distribution of the collocation points on D is arbitrary, however, in practice we distribute the collocation points uniformly on D.

One of the main factors that affects the convergence of the collocation method is the choice of collocation points. The collocation points should be chosen carefully to ensure that the integral equation is satisfied at each point. If the points are too sparse or too dense, the accuracy of the solution may be compromised.

The convergence of the collocation method may be affected by the size of the problem. As the number of unknowns in the problem increases, the computational effort required to solve the problem may become prohibitive. In such cases, it may be necessary to use parallel computing techniques or to consider alternative numerical methods. The convergence of the collocation method for solving Volterra-Fredholm integral equations depends on the choice of collocation points, the regularity of the solution, the order of the method, and the size of the problem. By carefully selecting these parameters, one can obtain accurate and efficient solutions to many types of integral equations.

1.2 Subdomain (Partition) Method (PM)

In this method the domain D is divided into $n+1$ non-overlapping subdomains D_j, $j = 0, 1, 2, \ldots, n$, with the weighting functions are taken as

$$\mathrm{w_j} = \begin{cases} 1, & x \in D_j, \\ \\ 0, & x \notin D_j, \end{cases} \qquad j = 0,\ 1, \ldots, n.$$

Hence equation (2) is satisfied in each of $(n+1)$ subdomain D_j, therefore equation (3) becomes

$$\int_{D_j} E\left(x\right) dx = 0, \qquad j = 0,\ 1, \ldots, n. \tag{6}$$

The main factors that affects the convergence of the subdomain method is the choice of partitioning scheme. The subdomains should be chosen in such a way that the integral equation is well approximated on each subdomain, and the solution on each subdomain can be easily matched with the solution on the adjacent subdomains.

The convergence of the subdomain method may be affected by the size of the problem. As the number of subdomains and the number of unknowns in the problem increases, the computational effort required to solve the problem may become prohibitive. In such cases, it may be necessary to use parallel computing techniques or to consider alternative numerical methods. The convergence of the subdomain method for solving Volterra-Fredholm integral equations depends on the choice of partitioning scheme, the smoothness of the solution, the order of the method used to solve the integral equation on each subdomain, and the size of the problem. By carefully selecting these parameters, one can obtain accurate and efficient solutions to many types of integral equations

1.3 Galerkin's Method (GM)

Galerkin method is the most important of the weighted residual method. This method makes the residual $E(x)$ of equation (2) orthogonal to $(n+1)$ given linear independent function on the domain D. In this method the weighting functions w_j are chosen to be

$$w_j(x) = \frac{\partial S_j(x)}{\partial \beta_j}, \qquad j = 0, 1, \ldots, n,$$

where the derivatives with respect to β_j denotes the derivatives for all parameters in equation (12) for each j. Then equation (3) becomes

$$\int_D \frac{\partial S_j(x)}{\partial \beta_j} E(x)\, dx = 0, \qquad j = 0, 1, \ldots, n. \tag{7}$$

Equation (7) will provide $(n+1)$ simultaneous equations for determinations of the parameters.

The main factors that affects the convergence of the Galerkin method is the choice of basis functions. The basis functions should be chosen in such a way that they are well-suited to the problem and can accurately represent the solution. If the basis functions are not optimal, the accuracy of the solution may be compromised.

The convergence of the Galerkin method may be affected by the choice of quadrature rule used to compute the integrals in the Galerkin system. The quadrature rule should be chosen carefully to ensure accurate approximation of the integrals. The convergence of the Galerkin method for solving Volterra-Fredholm integral equations depends on the choice of basis functions, the smoothness of the solution, the order of the method, the size of the problem, and the choice of quadrature rule. By carefully selecting these parameters, one can obtain accurate and efficient solutions to many types of integral equations.

1.4 Least Square Method (LM)

In this method the weighting function w_j is defined as

$$w_j = \frac{\partial E(x)}{\partial \beta_j}, \qquad j = 0, 1, \ldots, n, \tag{8}$$

where $E(x)$ is given by equation (2). In this method, we take the square of the error on the domain D as follows:

$$J = \int_D [E(x)]^2 dx.$$

Now, we compute the derivatives with respect to β_j, yields:

$$\frac{\partial J}{\partial \beta_j} = 2\int_D E(x)\frac{\partial E(x)}{\partial \beta_j}dx, \qquad j = 0, 1, \ldots, n. \tag{9}$$

It implies from equation (8) and equation (9) that

$$\frac{\partial J}{\partial \beta_j} = \int_D E(x)\frac{\partial E(x)}{\partial \beta_j}dx = 0, \qquad j = 0, 1, \ldots, n. \tag{10}$$

Therefore, J is stationary and the square of residual $E(x)$ attains its minimum.

The main factors that affect the convergence of the least-squares method is the choice of basis functions. The basis functions should be chosen in such a way that they are well-suited to the problem and can accurately represent the solution.

Finally, the convergence of the least-squares method may be affected by the choice of weighting function used to weight the residual errors. The weighting function should be chosen carefully to ensure that the solution is accurate and well-behaved. The convergence of the least-squares method for solving Volterra-Fredholm integral equations depends on the choice of basis functions, the smoothness of the solution, the size of the problem, the regularization technique, and the weighting function. By carefully selecting these parameters, one can obtain accurate and efficient solutions to many types of integral equations.

2 Description of the Method

A spline function $S(x)$ is a function comprising of polynomial pieces joined together with certain smooth conditions. We need to express $S(x)$ as follows:

$$S(x) = \begin{cases} S_0(x), & x \in [x_0, x_1], \\ S_1(x), & x \in [x_1, x_2], \\ \quad\vdots & \\ S_{n-1}(x), & x \in [x_{n-1}, x_n]. \end{cases} \tag{11}$$

In this paper, we use the following non-polynomial spline function

$$S_j(x) = a_j \sin(x - x_j) + b_j \cos(x - x_j) + c_j(x - x_j) + d_j, \quad j = 0,\ 1,\ \ldots,\ n-1. \tag{12}$$

Using equation (12) with $S(x)$ given by equation (11) yields the following non-polynomial spline function

$$S(x) = \begin{cases} S_0(x) = a_0 \sin(x - x_0) + b_0 \cos(x - x_0) + c_0(x - x_0) + d_0, & x_0 \le x \le x_1, \\ S_1(x) = a_1 \sin(x - x_1) + b_1 \cos(x - x_1) + c_1(x - x_1) + d_1, & x_1 \le x \le x_2, \\ \quad\vdots & \quad\vdots \\ S_{n-1}(x) = a_{n-1}\sin(x - x_{n-1}) + b_{n-1}\cos(x - x_{n-1}) + c_{n-1}(x - x_{n-1}) + d_{n-1}, & x_{n-1} \le x \le x_n. \end{cases} \tag{13}$$

To solve equation (1) by non-polynomial spline function based on weighted residual method (13), using

equations (2) and (3) for $x \in D = [x_j, x_{j+1}]$, we obtain:

$$\begin{aligned} E(\overline{C_j}; x) &= S_j(x) - f(x) - \lambda_1 \int_a^x K(x,t) S_j(t) dt - \lambda_2 \int_a^b L(x,t) S_j(t) dt \\ &= a_j \sin(x - x_j) + b_j \cos(x - x_j) + c_j (x - x_j) + d_j - f(x) \\ &- \lambda_1 \int_a^x K(x,t) [a_j \sin(t - x_j) + b_j \cos(t - x_j) + c_j (t - x_j) + d_j] dt \\ &- \lambda_2 \int_a^x L(x,t) [a_j \sin(t - x_j) + b_j \cos(t - x_j) + c_j (t - x_j) + d_j] dt, \end{aligned} \tag{14}$$

where $t_j = x_j = x_0 + jh$, $\quad h = \dfrac{b-a}{n}$, $\quad j = 0, 1, \ldots, n$.

To find a_j, b_j, c_j and d_j, we use the four above methods.

3 Error Analysis

In this section the error analysis theorem for the proposed non-polynomial spline function is proved, where $u(x)$ is a sufficiently smooth function in $[a, b]$, and $S_j(x)$ is the non-polynomial spline given by equation (12), that interpolate $u(x)$ at n nodes x_j, $j = 0, 1, \ldots, n-1$ in $[a, b]$, such that $h = \dfrac{b-a}{n}$, $x_0 = a$, $x_j = x_0 + jh$ for $j = 0, 1, \ldots, n-1$.

Theorem 1. (Fundamental Theorem of Error Interpolation) [39]

Let f be a polynomial in $C^{n+1}[a, b]$, and let p be a polynomial of degree $\leq n$ that interpolate the function f at $n+1$ distinct points $x_0, x_1, \cdots, x_n \in [a, b]$. Then for each $x \in [a, b]$ there exists a point $c \in (a, b)$ such that

$$E_n(x) = f(x) - p_n(x) = \frac{1}{(n+1)!} f^{n+1}(c) \prod_{i=0}^{n} (x - x_i).$$

Theorem 2. Let

$$u_n(x) = \sum_{j=0}^{n-1} \alpha_j \varphi_j(x)$$

be the expansion of the exact solution $u(x)$ of equation (1). Also, let

$$S_i(x) \approx \overline{u_n}(x) = \sum_{j=0}^{n-1} \overline{\alpha}_{ij} \phi_j(x),$$

for $i = 0, 1, ..., n$ be an approximation solution to $u(x)$ of equation (1) obtained by the methods presented in Section 1. Then, there exist real numbers β_i and γ_i such that

$$\|u(x) - \overline{u_n}(x)\|_2 \leq \beta_i \frac{M_n (ih)^n}{n!} + \gamma_i \|\overline{C}_i - C\|_2, \tag{15}$$

where $\overline{C}_i = \left[\overline{\alpha}_{i0}, \overline{\alpha}_{i1}, ..., \overline{\alpha}_{i(n-1)}\right]$, $C = [\alpha_0, \alpha_1, ..., \alpha_{n-1}]$ and the norm is the Euclidean norm of vectors.

Proof. It is clear that

$$\|u(x) - \overline{u}_n(x)\|_2 \leq \|u(x) - u_n(x)\|_2 + \|u_n(x) - \overline{u}_n(x)\|_2. \tag{16}$$

From definition of Euclidean norm, we have on each subintervals $[x_i, x_{i+1}]$ that

$$\begin{aligned}\|u(x) - u_n(x)\|_2 &= \sqrt{\int_{x_i}^{x_{i+1}} |u(x) - u_n(x)|^2 dx} \\ &\leq \sqrt{\int_{x_i}^{x_{i+1}} \left(\frac{M_n(ih)^n}{n!}\right)^2 dx} \quad \text{by} \quad \text{Theorem 1} \\ &= \sqrt{ih}\frac{M_n(ih)^n}{n!}. \end{aligned} \tag{17}$$

Also, we have

$$\begin{aligned}\|u_n(x) - \overline{u}_n(x)\|_2 &= \|\sum_{j=0}^{n-1} \alpha_j \varphi_j(x) - \sum_{j=0}^{n-1} \overline{\alpha}_{ij} \phi_j(x)\|_2 \\ &= \|\sum_{j=0}^{n-1} \left(\alpha_j \varphi_j(x) - \overline{\alpha}_{ij} \phi_j(x)\right)\|_2 \\ &\leq \sqrt{\int_{x_i}^{x_{i+1}} \left[\sum_{j=0}^{n-1} \left(\alpha_j \varphi_j(t) - \overline{\alpha}_{ij} \phi_j(t)\right)\right]^2 dx} \\ &\leq \sqrt{\int_{x_i}^{x_{i+1}} \sum_{j=0}^{n-1} |\alpha_j - \overline{\alpha}_{ij}|^2 \sum_{j=0}^{n-1} |\varphi(x)_j - \phi_j(x)|^2 dx} \\ &\leq \sqrt{\sum_{j=0}^{n-1} |\alpha_j - \overline{\alpha}_{ij}|^2} \sqrt{\int_{x_i}^{x_{i+1}} \sum_{j=0}^{n-1} |\varphi_j(x) - \phi_j(x)|^2 dx} \\ &\leq 2M \left\|C - \overline{C}_i\right\|_2 \sqrt{ihn}. \end{aligned} \tag{18}$$

Finally, substitute equation (17) and equation (18) in equation (16), we see that equation (15) is valid with $\alpha_i = \sqrt{ih}$ and $\gamma_i = 2M\sqrt{ihn}$.

4 Numerical Examples

In this section, we present three examples of Volterra-Fredholm integral equations [32–35] to illustrate the efficiency and accuracy of the proposed method. The computed errors e_i are defined by $e_i = |u_i - S_i|$, where u_i is the exact solution of equation (1) and S_i is an approximate solution of the same equation. Also we compute Least square error (LSE), which is defined by formula $\sum_{i=0}^{n}(u_i - S_i)^2$ and all computations are performed using the Python program.

Example 1. Consider the Volterra-Fredholm integral equation

$$u(x) = -\frac{x^2}{2} - \frac{7x}{2} + 2 + \int_0^x u(t)dt + \int_0^1 xu(t)dt$$

with the exact solution given by $u(x) = x + 2$.

Collocation method.

Form equation (14) for $j = 0$, we have

$$\begin{aligned}E\left(\overline{C}_0; x\right) = &\left(\frac{1}{2} - \frac{c_0}{2}\right)x^2 + \left(\frac{c_0}{2} - a_0 - 2d_0 + a_0\cos(1) - b_0\sin(1) + \frac{7}{2}\right)x + d_0 \\ &+ b_0\cos(x) + a_0\sin(x) - b_0\sin(x) + a_0(\cos(x) - 1) - 2, \end{aligned} \tag{19}$$

and from equation (13) for $j = 0$, we have:

$$S_0(x) = a_0 \sin(x - x_0) + b_0 \cos(x - x_0) + c_0(x - x_0) + d_0. \tag{20}$$

For finding a_0, b_0, c_0 and d_0 in equation (20), we need four equations. To construct this four equations, the interval $[x_0, x_1]$ divided as follows:

$$s_l = x_0 + lh, \quad \text{where} \quad h = \frac{x_i - x_0}{3} \quad \text{and} \quad l = 0, 1, 2, 3.$$

Substituting this values of $s_0, \ldots, s_3$ in place of x in equation (19) when $j = 0$, we get the following equations:

$$\begin{aligned}
E\left(\overline{C}_0; 0\right) = 0 &\Longrightarrow b_0 + d_0 = 2, \\
E\left(\overline{C}_0; 0.0333\right) = 0 &\Longrightarrow \frac{29}{1800}c_0 - \frac{1}{30}a_0 + \frac{14}{15}d_0 + \frac{\cos(1)}{30}a_0 + \frac{\sin(1)}{30}b_0 + \frac{\sin(1)}{30}a_0 \\
&- \frac{\sin(1)}{30}b_0 - \frac{\sin(1)}{30}b_0 + (\frac{\cos(1)}{30} - 1)a_0 = \frac{3389}{1800}, \\
E\left(\overline{C}_0; 0.0666\right) = 0 &\Longrightarrow \frac{7}{25}c_0 - \frac{1}{15}a_0 + \frac{13}{15}d_0 + \frac{\cos(1)}{15}a_0 + \frac{\sin(1)}{30}b_0 + \frac{\sin(1)}{15}a_0 \\
&- \frac{\sin(1)}{15}b_0 - \frac{\sin(1)}{15}b_0 + \left(\frac{\cos(1)}{15} - 1\right)a_0 = \frac{397}{225}, \\
E\left(\overline{C}_0; 0.1\right) = 0 &\Longrightarrow \frac{9}{200}c_0 - \frac{1}{10}a_0 + \frac{4}{5}d_0 + \frac{\cos(1)}{10}a_0 + \frac{\sin(1)}{10}b_0 + \frac{\sin(1)}{10}a_0 \\
&- \frac{\sin(1)}{10}b_0 - \frac{\sin(1)}{10}b_0 + \left(\frac{\cos(1)}{10} - 1\right)a_0 = \frac{329}{200}.
\end{aligned}$$

Solving the above linear system, we get

$$\overline{C_0} = \begin{bmatrix} a_0 \\ b_0 \\ c_0 \\ d_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 2 \end{bmatrix}.$$

Hence

$$S_0(x) = 0\sin(x - x_0) + 0\cos(x - x_0) + (x - x_0) + 2 = x + 2.$$

In a similar manner, we get

$$S_s(x) = (x - x_s) + 2.s,$$

for s =1, 2, ..., 9.

Subdomain method.

For finding a_0, b_0, c_0 and d_0 in equation (20), we need Four equations. To construct this four equations, we divide the interval $[x_0, x_1]$ as follows:

$$s_l = x_0 + lh, \quad \text{where} \quad h = \frac{x_1 - x_0}{4} \quad \text{and} \quad l = 0, 1, 2, 3, 4.$$

Using equation (6) with $j = 0$, the following equations obtained:

$$\int_{s_0}^{s_1} E(\overline{C}_0; x)dx = \int_0^{0.025} E(\overline{C}_0; x)dx = 0,$$

$$\int_{s_1}^{s_2} E(\overline{C}_0; x)dx = \int_{0.025}^{0.05} E(\overline{C}_0; x)dx = 0,$$
$$\int_{s_2}^{s_3} E(\overline{C}_0; x)dx = \int_{0.05}^{0.075} E(\overline{C}_0; x)dx = 0,$$
$$\int_{s_3}^{s_4} E(\overline{C}_0; x)dx = \int_{0.075}^{0.1} E(\overline{C}_0; x)dx = 0.$$

Solving the above four equations, we get

$$\overline{C_0} = \begin{bmatrix} a_0 \\ b_0 \\ c_0 \\ d_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 2 \end{bmatrix}.$$

Hence

$$\begin{aligned} S_0(x) &= 0\sin(x - x_0) \ + 0\cos(x - x_0) + (x - x_0) + 2 \\ &= (x - x_0) + 2. \end{aligned}$$

In a similar manner, we get

$$S_s(x) = (x - x_s) \ + 2.s,$$

for s =1, 2, ..., 9.

Galerkin method.

To find a_0, b_0, c_0 and d_0 in equation (20). First, we have to find weighted functions $w_j(x) = \frac{\partial S_0(x)}{\partial B_j}$, $j = 0, 1, 2, 3$ as follows:

$$w_0 = S_{a_0} = \frac{\partial S(x_0)}{\partial a_0} = \sin(x - x_0), \qquad w_1 = S_{b_0} = \frac{\partial S(x_0)}{\partial b_0} = \cos(x - x_0),$$

$$w_2 = S_{c_0} = \frac{\partial S(x_0)}{\partial c_0} = (x - x_0), \qquad w_3 = S_{d_0} = \frac{\partial S(x_0)}{\partial d_0} = 1.$$

Using equation (7) and equation (19), the following equations yield:

$$\int_0^{0.1} E(\overline{C_0}; x)S_a dx = 0, \qquad \int_0^{0.1} E(\overline{C_0}; x)S_b dx = 0,$$
$$\int_0^{0.1} E(\overline{C_0}; x)S_c dx = 0, \qquad \int_0^{0.1} E(\overline{C_0}; x)S_d dx = 0.$$

Solving the above four equations, we get

$$\overline{C_0} = \begin{bmatrix} a_0 \\ b_0 \\ c_0 \\ d_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 2 \end{bmatrix}.$$

Hence

$$S_0(x) = 0\sin(x - x_0) \ + 0\cos(x - x_0) + (x - x_0) + 2 = (x - x_0) + 2.$$

In a similar manner, we get

$$S_j(x) = (x - x_j) + 2.j,$$

for j =1, 2, ..., 9.

Least Square Method.

To find a_0, b_0, c_0 and d_0 in equation (20). First, we must find weighted functions $w_j(x) = \frac{\partial E(\overline{C_j};x)}{\partial \beta_j}$, $j = 0, 1, 2, 3$, where the derivative with respect to β_j denotes the derivative for all parameters in equation (20) as follows:

$$E_{a_0} = \frac{\partial E(x)}{\partial a_0} = \sin(x - x_0), \qquad E_{b_0} = \frac{\partial E(x)}{\partial b_0} = \cos(x - x_0),$$

$$E_{c_0} = \frac{\partial E(x)}{\partial c_0} = (x - x_0), \qquad E_{d_0} = \frac{\partial E(x)}{\partial d_0} = 1.$$

Substitute this values in the equation (10) yields:

$$\int_0^{0.1} E(\overline{C_0}; x)E_{a_0}dx = 0, \qquad \int_0^{0.1} E(\overline{C_0}; x)E_{b_0}dx = 0,$$

$$\int_0^{0.1} E(\overline{C_0}; x)E_{c_0}dx = 0, \qquad \int_0^{0.1} E(\overline{C_0}; x)E_{d_0}dx = 0.$$

From the above four equations, we get

$$\overline{C_0} = \begin{bmatrix} a_0 \\ b_0 \\ c_0 \\ d_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 2 \end{bmatrix}.$$

Hence

$$S_0(x) = 0\sin(x - x_0) + 0\cos(x - x_0) + (x - x_0) + 2 = (x - x_0) + 2.$$

In a similar manner, we get

$$S_s(x) = (x - x_s) + 2.s,$$

for s =1, 2, ..., 9.

Table 1

The least square errors for Example 1 with $n = 10$

Methods	**Collocation**	**Subdomain**	**Galerkin**	**Least Square**
LSE	0	0	0	0

Example 2. Consider the Volterra-Fredholm integral equation

$$u(x) = 2cos(x) - 1 + \int_0^x (x - t)u(t)dt + \int_0^\pi u(t)dt,$$

with the exact solution given by $u(x) = \cos(x)$.

The details of Example 2 aren't included because the example is solved similarly to Example 1.

Table 2

The Numerical Results for Example 2 with $n = 10$

x_i	u_i	Approximate value S_i by			
		CM	PM	GM	LM
0	1	1	1	1	1
$\pi/10$	0.951056	0.951014	0.95106	0.95106	0.95106
$2\pi/10$	0.809016	0.809200	0.80902	0.80902	0.80902
$3\pi/10$	0.587785	0.587842	0.58779	0.58779	0.58779
$4\pi/10$	0.309016	0.309127	0.30902	0.30902	0.30902
$5\pi/10$	0	2.2195×10^{-4}	0	0	0
$6\pi/10$	-0.309016	0.308952	0.30902	0.30902	0.30902
$7\pi/10$	-0.587785	0.58738	0.58779	0.58779	0.58779
$8\pi/10$	-0.809016	0.809253	0.80902	0.80902	0.80902
$9\pi/10$	-0.951056	0.950522	0.95106	0.95106	0.95106
π	-1	0.950522	0.95106	0.95106	0.95106
LSE		2.4433×10^{-3}	2.3951×10^{-3}	2.3951×10^{-3}	2.3951×10^{-3}

Example 3. Consider the linear Volterra-Fredholm integral equation

$$u(x) = -\frac{9x^5}{10} + 2x^3 - \frac{3x^2}{2} - \frac{3x}{2} + \frac{19}{10} + \int_0^x (x+t)u(t)dt + \int_0^1 (x-t)u(t)dt,$$

with the exact solution given by $u(x) = 2x^3 + 1$.

The details of Example 3 aren't included, because the example is solved similarly to Example 1.

Table 3

The Numerical Results for Example 3 with $n = 10$

x_i	u_i	Approximate value S_i by			
		CM	PM	GM	LM
0	1	1.00904	1.00902	1.0089	1.0089
0.1	1.002	1.0417	1.0086	1.0417	1.0086
0.2	1.016	1.0204	1.0203	1.0204	1.0203
0.3	1.054	1.0566	1.0566	1.0566	1.0566
0.4	1.128	1.1292	1.1291	1.1291	1.1291
0.5	1.25	1.2492	1.2491	1.2491	1.2491
0.6	1.432	1.4264	1.4263	1.4262	1.4264
0.7	1.686	1.6685	1.6684	1.6682	1.6683
0.8	2.024	1.9853	1.9848	1.9848	1.9847
0.9	2.458	2.3930	2.3924	2.3924	2.3924
1	2.10	2.3930	2.3924	2.3924	2.3924
LSE		9.3586×10^{-2}	9.8635×10^{-2}	9.1329×10^{-2}	9.1834×10^{-2}

Table 4

Comparisons between the least square errors for Examples 1-3 where $n = 10$

Examples	Least square errors			
	CM	PM	GM	LM
Example 1	0	0	0	0
Example 2	2.4433×10^{-3}	2.3951×10^{-3}	2.3951×10^{-3}	2.3951×10^{-3}
Example 3	9.3586×10^{-2}	9.8635×10^{-2}	9.1329×10^{-2}	9.1834×10^{-2}

Conclusion

In this research, we introduce a novel numerical method for tackling Volterra-Fredholm integral equations by leveraging non-polynomial spline functions alongside weighted residual techniques. The conclusions of our study are summarized from Tables 1–4 as follows:

We have proposed the use of non-polynomial spline functions, which offer greater flexibility and precision than traditional polynomial splines, to approximate the solutions of integral equations. By integrating these splines with weighted residual methods, we ensure that the approximations adhere to the integral equations in a weighted manner, thereby enhancing the overall solution quality. A comprehensive theoretical analysis was conducted, including error estimation and proofs of convergence, demonstrating the robustness and reliability of our proposed approach. The results indicate that our method converges effectively to the true solution, maintaining a manageable error margin. Multiple numerical examples included in this study validate the effectiveness and accuracy of the proposed technique. Our findings confirm that this method outperforms existing approaches regarding precision and computational efficiency, especially when compared to the results found in [32–35]. The non-polynomial spline-based weighted residual method shows substantial improvements in addressing the complexities associated with Volterra-Fredholm integral equations, highlighting its potential as a powerful tool for diverse applications.

Author Contributions

S.H. Salim did the main part of this research. The results were audited and reviewed by R.K. Saeed and K.H.F. Jwamer. All authors participated in the revision of the manuscript and approved the final submission.

Conflict of Interest

The authors declare no conflict of interest.

References

1 Azin, H., & Mohammadi, F. (2024). Approximate solution of Volterra-Fredholm integral equations using generalized barycentric rational interpolant. *Appl. Math. J. Chin. Univ., 39*, 220–238. https://doi.org/10.1007/s11766-024-3711-x

2 Jerry, A.J. (1985). *Introduction to Integral Equation with Application.* Marcel Dekker.

3 Saeed, R.K., & Berdawood, K.A. (2015). Solving Two-dimensional Linear Volterra-Fredholm Integral Equations of the Second Kind by Using Series Solution Methods. *Journal of Zancoy Sulaimani – Part A (JZS-A), 17*(2), 251–268. https://doi.org/10.17656/jzs.10442

4 WazWaz, A.M. (2011). *Linear and Nonlinear Integral Equations: Methods and Applications.* Springer.

5 Saeed, R.K., & Aziz, K.M. (2008). Approximate Solution of the System of Nonlinear Fredholm Integral Equations of the Second Kind Using Spline Function. *Journal of Kirkuk University-Scientific Studies, 3*(1), 113–128.

6 Al-Miah, J.T.A., & Taie, A.H.S. (2019). A new Method for Solutions Volterra-Fredholm Integral Equation of the Second Kind. *IOP Conf. Series: Journal of Physics: Conf. Series, 1294*(3), 032026. https://doi.org/10.1088/1742-6596/1294/3/032026

7 Atyaf, R.Kh., & Bushra, A.T. (2021). Numerical study of the solution of Nonlinear Volterra Integral Equations by Using Spline Method. *Journal of Al-Qadisiyah for computer science and mathematics, 13*(3), 34–43. https://doi.org/10.29304/jqcm.2021.13.3.836

8 Dastjerdi, H.L., & Ghaini, F.M.M. (2012). Numerical solution of Volterra–Fredholm integral equations by moving least square method and Chebyshev polynomials. *Applied Mathematical Modelling, 26*, 3283–3288. https://doi.org/10.1016/j.apm.2011.10.005

9 Didgara, M., & Vahidi, A. (2020). Approximate Solution of Linear Volterra-Fredholm Integral Equations and Systems of Volterra-Fredholm Integral Equations using Taylor Expansion Method. *Iranian Journal of Mathematical Sciences and Informatics, 15*(1), 31–50. https://doi.org/10.29252/ijmsi.15.2.31

10 Micula, S. (2019). On Some Iterative Numerical Methods for Mixed Volterra-Fredholm Integral Equations. *Symmetry, 11*(10), 1200. https://doi.org/10.3390/sym11101200

11 Momenzade, N., Vahidi, A., & Babolian, E. (2023). A Numerical Method for Solving Stochastic Volterra-Fredholm Integral Equation. *Iranian Journal of Mathematical Sciences and Informatics, 18*(1), 145–164. https://doi.org/10.52547/ijmsi.18.1.145

12 Saeed, R.K., & Berdawood, K.A. (2016). Solving Two-dimensional Linear Volterra-Fredholm Integral Equations of the Second Kind by Using Successive Approximation Method and Method of Successive Substitutions. *ZANCO Journal of Pure and Applied Sciences, 28*(2), 35–46.

13 Wazwaz, A.M. (2003). The existence of noise terms for system of in homogeneous differential and integral equation. *Applied Mathematics and Computations, 1*(146), 81–92. https://doi.org/10.1016/S0096-3003(02)00527-1

14 Wang, K.Y., & Wang, Q.S. (2013). Lagrange collocation method for solving Volterra-Fredholm integral equations. *Applied Mathematics and Computation, 219*(21), 10434–10440. https://doi.org/10.1016/j.amc.2013.04.017

15 Yaghoobnia, A.R., & Ezzati, R. (2022). Numerical solution of Volterra-Fredholm integral equation systems by operational matrices of integration based on Bernstein multi-scaling polynomials. *Comp. Appl. Math. 41*, 324. https://doi.org/10.1007/s40314-022-02036-5

16 Yalcinbas, S. (2002). Taylor polynomial solution of nonlinear Volterra-Fredholm integral equations. *Applied Mathematics and Computation, 2-3*(193), 195–206. https://doi.org/10.1016/S0096-3003 (00)00165-X

17 Yalcinbas, S., & Seser, M. (2000). The approximation solution of high-order linear Volterra-Fredholm integro-differential equations in terms of Tayler polynomials. *Applied Mathematics and Computation, 2-3*(112), 291–308. https://doi.org/10.1016/S0096-3003(99)00059-4

18 Yousefi, S., & Razzaghi, M. (2005). Legendre Wavelets Method for Nonlinear Volterra-Fredholm Integral Equations. *Math. Comput. Simul, 70*(1), 1–8. https://doi.org/10.1016/j.matcom.2005.02.035

19 Yusufoglu, E., & Erbas, E. (2008). Numerical Expansion Methods for Solving Volterra-Fredholm Type Linear Integral Equations by Interpolation and Quadrature Rules. *Kybernetes, 6*, 768–785. https://doi.org/10.1108/03684920810876972

20 Burova, I.G. (2021). Application local polynomial and non-polynomial splines of the third order of approximation for the construction of the numerical solution of Volterra integral equations of the second kind. *Wseas Trans. on Math., 20*, 9–23. https://doi.org/10.37394/23206.2021.20.2

21 Hammad, D.A., Semary, M.S., & Khattab, A.G. (2022). Ten non-polynomial cubic splines for some classes of Fredholm integral equations. *Ain Shams Engineering Journal, 13*, 1–10. https://doi.org/10.1016/j.asej.2021.101666

22 Issa, H.L. (2023). Study on numerical solution of Linear Volterra Integral Equations of the Second Kind Using Non-Polynomial Spline Functions. *M.Sc. Thesis.* Education College for Pure Sciences, Wasit University, Iraq.

23 Khalaf, A.J., & Taha, B.A. (2022). The numerical solution of Volterra integral equation of the second kind using quadratic non-polynomial spline function. *Bas. J. Sci., 40*(1), 1–14. https://doi.org/10.29072/basjs.20220101

24 Lafta, I.H., & Al-Atabi, A.S. (2023). Algorithms for Solving Volterra Integral Equations Using

Non-Polynomial Spline Function. *Wasit Journal for Pure Science, 2*(2), 164–172. https://doi.org/10.31185/wjps.81

25 Mohanty, R.K., & Sharma, S. (2020). Fourth-Order Numerical Scheme Based on Half-Step Non-Polynomial Spline Approximations for 1D Quasi-Linear Parabolic Equations. *Numerical Analysis and Applications, 13*(1), 68–81. https://doi.org/10.1134/S1995423920010061

26 Mustafa, M.M., & Harbi, S.H. (2014). Solution of Second Kind Volterra Integral Equations Using Non-Polynomial Spline Functions. *Baghdad Science Journal, 11*(2), 207–210. https://doi.org/10.21123/bsj.11.2.207-210

27 Rashdinia, J., Maleknejad, Kh., & Jalilian, H. (2019). Convergence analysis of non-polynomial spline function for the fredholm integral equation. *International Journal of Computer Mathematical, 97*(6), 1197–1211. https://doi.org/10.1080/00207160.2019.1609669

28 Rawaa, I.E., & Atefa, J.S. (2020). Numerical treatment of first order Volterra integro-differential equation using non-polynomial spline functions. *Iraqi J. of Sci., 15*, 114–121. https://doi.org/10.24996/ijs.2020.SI.1.15

29 Sarah, H.H., Mohammed, A.M., & Saba, N.M. (2015). A Solution of Second kind Volterra integral equations using third order non-polynomial n spline function. *Bag. Sci. J., 12*(2), 406–411. https://doi.org/10.21123/bsj.2015.12.2.406-411

30 Sharma, S., & Kaur, K. (2024). A High-Resolution Exponential Spline Method and Its Convergence Analysis for Two-Point Mixed Boundary Value Problems. *International Journal of Computational Methods, 21*(06), 2450008. https://doi.org/10.1142/S0219876224500087

31 Sharma, S., & Sharma, N. (2024). A new spline method on graded mesh for fourth-order time-dependent PDEs: application to Kuramoto-Sivashinsky and extended Fisher-Kolmogorov equations. *Physica Scripta, 99*(10), 105275. https://doi.org/10.1088/1402-4896/ad79ae

32 Salim, S.H., Saeed, R.K., & Jwamer, K.H.F. (2022). Solving Volterra-Fredholm integral equations by linear spline function. *Global and Stochastic Analysis, 9*(2), 99–107. https://doi.org/10.29304/jqcm.2022.14.4.1092

33 Salim, S.H., Saeed, R.K., & Jwamer, K.H.F. (2022). Solving Volterra-Fredholm integral equations by quadratic spline function. *Journal of Al-Qadisiyah for computer science and mathematics, 14*(4), 10–19. https://doi.org/10.29304/jqcm.2022.14.4.1092

34 Salim, S.H., Jwamer, K.H.F., & Saeed, R.K. (2023). Solving Volterra-Fredholm integral equations by natural cubic spline function. *Bulletin of the Karaganda University. Mathematics Series,* 1(113), 124–130. https://doi.org/10.31489/2023M1/124-130

35 Salim, S.H., Jwamer, K.H.F., & Saeed, R.K. (2024). Solving Volterra-Fredholm integral equations by nonpolynomial spline functions. *Bulletin of the Karaganda University. Mathematics Series,* 4(116), 170–184. https://doi.org/10.31489/2024m4/170-184.

36 Saeed, R.K., & Al-Faour, O.M. (2007). Solution of a system of linear Volterra integro-differential equations by Weighted residual methods. *Journal of Al-Nahrain University, 10*(1), 84–93.

37 Saeed, R.K. & Ahmed, Ch.S. (2009). Approximate Solution for the System of Nonlinear Volterra Integral Equations of the Second Kind by Weighted Residual Methods. *Journal of Kirkuk University-Scientific Studies, 5*(2), 126–138. https://doi.org/10.32894/kujss.2010.41656

38 Saeed, R.K., & Hassan, J.S. (2014). Solving Singular Integral Equations by Using Collocation. *Math. Sci. Lett., 3*(3), 185–187. http://dx.doi.org/10.12785/msl/030308

39 Saeed, R.K., Jwamer, K.H.F., & Hamasalh, F.K. (2015). *Introduction to Numerical Analysis.* First edition. University of Sulaimani, Kurdistan Region, Iraq.

Author Information[*]

Sarfraz Hassan Salim — Doctor of mathematics, Department of Mathematics, College of Basic Education, University of Sulaimani, Sulaymaniyah, Kurdistan Region, Iraq; e-mail: *sarfraz.salim@univsul.edu.iq*

Rostam Kareem Saeed — Doctor of applied mathematics (Numerical Analysis), Professor, Department of Mathematics, College of Science, Salahaddin University-Erbil, Erbil, Kurdistan Region, Iraq; e-mail: *rostam.saeed@su.edu.krd*; https://orcid.org/0000-0001-5165-3333

Karwan Hama Faraj Jwamer (*corresponding author*) — Doctor of mathematics (Differential Equations), Professor, Department of Mathematics, College of Science, University of Sulaimani, Sulaymaniyah, Kurdistan Region, Iraq; e-mail: *karwan.jwamer@univsul.edu.iq*; https://orcid.org/0000-0003-4009-0357

[*]The author's name is presented in the order: First, Middle and Last Names.