



# A Novel Super-Convergent Numerical Method for Solving Nonlinear Volterra Integral Equations Based on B-Splines

M. Ghasemi, A. Goligerdian and S. Moradi

**Abstract.** We introduce and thoroughly examine a novel approach grounded in B-spline techniques to address the solution of second-kind nonlinear Volterra integral equations. Our method revolves around the application of B-spline interpolation, incorporating innovative end conditions, and delving into the associated existence and error estimation aspects. Notably, we develop this technique separately for even and odd-degree splines, leading to super-convergent approximations, particularly significant when employing even-degree splines. This paper extends its commitment to a comprehensive analysis, delving deeply into the method's convergence characteristics and providing insightful error bounds. To empirically validate our approach, we present a series of numerical experiments. These experiments underscore the method's efficacy and practicality, showcasing numerical approximations that closely align with the anticipated theoretical outcomes. Our proposed method thus emerges as a promising and robust tool for addressing the challenging realm of nonlinear Volterra integral equations, bridging the gap between theoretical expectations and practical applications.

**Mathematics Subject Classification.** 45Dxx, 65R20, 41A15.

**Keywords.** B-spline, Volterra integral equations, Super-convergence, Gauss–Legendre quadrature.

## 1. Introduction

Integral equations is a branch of mathematics that deals with equations in which an unknown function appears under an integral sign. These equations arise in many areas of science and engineering, such as physics, chemistry, biology, economics, signal processing, potential theory electrostatics and fluid dynamics [1–12]. The theory of integral equations is an important tool for understanding and solving complex problems in various fields of science and engineering. Consider the second kind nonlinear Volterra integral equation

$$u(x) = \int_0^x G(x, t, u(t)) dt + g(x), \quad x \in [0, T], \quad (1.1)$$

where  $g$  and  $G$  are given functions. A large variety of numerical methods have been developed to approximate the numerical solution of Volterra integral equations. In [13], the higher degree fuzzy transform technique is used for the numerical solution of the second kind Volterra integral equations with singular and nonsingular kernels. A smoothing transformation technique based on quasi-linearization and product integration methods has been used for the numerical solution of nonlinear weakly singular Volterra integral equations in [14]. A solution has been obtained in [15], using the fixed point technique in the setting of dislocated extended b-metric space for the Volterra integral equations. Xiao-yong [16] developed a high-order algorithm for the solution of nonlinear Volterra integral equations with vanishing variable delays. The algorithm was a nontrivial extension of single step methods. A Runge–Kutta method based on the first kind Chebyshev polynomials has been developed to approximate the solution of nonlinear stiff Volterra integral equations of the second kind in [17]. In [18], a collocation method based on Chebyshev polynomial interpolation is developed for the numerical solution of Volterra integral equations of the second kind. This method used the roots of Chebyshev polynomial as collocation points. A lot of other numerical approaches such as Chebyshev collocation method [19], Romberg extrapolation method [20, 21], iterated Galerkin method [22] and finite difference method [23] have been used to approximate the numerical solution of Volterra integral equations of the second kind.

Using spline functions in the context of numerical solution of integral equations dates back to 1970s. Hung [24], used spline function to approximate the solution of Volterra integral equations of the first and second kind. He obtained a convergent scheme based on linear and quadratic spline. He also proved that using cubic spline leads to divergent approximations. Tom [25], developed an approximation based on spline function to approximate the solution Volterra integral equations of the second kind. It is proved that, splines of degree greater than two with maximum degree of smoothness lead to unstable approximations for the solution of Volterra equations of the second time. Tom [26], determined the conditions for stability of spline based methods for the solution of Volterra integral equations. In [27], a  $C^2$ -cubic spline is used to approximate the solution of the Volterra integral equation of the second kind but the question of stability does not answered. Kauthen [28], developed a polynomial spline collocation method for the solution of an integral-algebraic equations of the Volterra type. In successive papers [29–32], Oja et al. discuss the stability and convergence of spline collocation method for the solution of Volterra integral equations of the second kind. It is proved that solving Volterra integral equations with step by step collocation using  $m$ -th degree spline of class  $C^{m-1}[0, T]$  leads to an unstable and divergent approximations for  $m \geq 3$ . In recent years, many researchers did attempt to approximate the solution of second kind Volterra integral equations using spline [33–37].

In the current work, we endeavor to devise a novel approach anchored in the collocation B-spline method, tailored to solve nonlinear Volterra integral equations of the second kind. Our methodology is meticulously designed, taking into account both odd and even-degree splines, which are treated separately to harness their unique strengths. For odd degree splines, we collocate the problem on the grid points of the uniform partition while the midpoints of the partition are used for the even degrees. For  $m \geq 2$ , some extra relations are needed to uniquely determine the spline approximation. To construct such extra relations, we collocate the problem at near boundary midpoints for odd values of  $m$  and near boundary grid points for even values of  $m$ . At first, we prove that such a spline is exist and we obtain the local error bounds for interpolating sufficiently smooth functions. Then, we use the method to approximate the solution of nonlinear Volterra integral equations of the second kind. It is proved that the method is super-convergent for even  $m$ .

The rest of the paper is organized as follows: In Sect. 2, we will introduce a B-spline collocation method with new end conditions and we will obtain some local error bounds to be used in the convergence analysis. In Sect. 3, we will construct our new approach based on proposed B-spline for the solution of the nonlinear Volterra integral equations. Section 4 is devoted to the convergence analysis and error bound of the method. In Sect. 5, we will solve some problems to show the applicability and efficiency of the proposed method.

## 2. B-Spline Collocation Method

B-splines are the splines with smallest compact support, which means that the basis functions are non-zero only within a small interval or segment of the domain. Such a property makes B-splines computationally efficient, as the calculations can be localized to these smaller segments. B-splines form a partition of unity, which means that their sum over the entire domain is equal to one. This property allows for the construction of a global approximation by combining local B-spline basis functions. Each basis function is associated with a specific control point and only affects a small region of the domain. By properly choosing the control points and their corresponding basis functions, it is possible to create a global approximation that accurately represents the behavior of the solution throughout the domain. The partition of unity property also ensures that the approximated solution satisfies important properties, such as conservation laws or boundary conditions. This is crucial in many applications where physical laws or constraints need to be preserved. The partition of unity and non-negativity of the B-spline basis functions are equivalent to the affine invariance and the convex hull properties of B-spline curves. Meaning that, a B-spline curve is contained in the convex hull of its control polyline. Consequently, as the interpolated function  $u$  moves from  $a$  to  $b$  and crosses a knot, a basis function becomes zero and a new non-zero basis function becomes effective. As a result, one control point whose coefficient becomes zero will leave the definition of the current convex

hull and is replaced with a new control point whose coefficient becomes non-zero. One of the advantages of using B-splines for this purpose is their ability to provide a smooth and continuous representation of the solution. This is particularly important in the context of differential equations, where the solution is often expected to be differentiable and exhibits certain smoothness properties.

In what follows, we will provide a succinct overview of (cardinal) B-spline functions and their properties. Afterward, we will introduce a novel numerical scheme that utilizes the Gauss–Legendre quadrature method. This scheme aims to numerically solve integral equations with high precision and convergence. If you wish to delve deeper into the subject matter, we recommend referring to references [38] and [39] for comprehensive information on B-splines and Gauss–Legendre quadrature method.

Let  $\Delta \equiv \{0 = x_0 < x_1 < \dots < x_n = T\}$  be a uniform partition of the interval  $[0, T]$  with  $h = \frac{T}{n}$  as step size and  $\Omega \equiv \{\tau_i = x_i - \frac{h}{2}\}_{i=1}^n$  be the set of mid-points of  $\Delta$ . Also let denote by  $I_i = [x_{i-1}, x_i], i = 1, \dots, n - 1$  and  $I_n = [x_{n-1}, x_n]$ , the subintervals of the partition  $\Delta$ . Given a positive integer  $m < n$ , the space of piecewise polynomials of degree  $m$  is defined as

$$\mathcal{P}\Pi_m(\Delta) \equiv \{q : \exists p_i \in \Pi_m, i = 1, \dots, n, \text{ s.t. } q(x) = p_i(x), \forall x \in I_i\},$$

where  $\Pi_m$  is the space of polynomials of degree  $m$ . Now for  $\Delta$  and  $m$  defined above the space of polynomial spline of degree  $m$  with the simple knots at the points  $x_0, x_1, \dots, x_n$  can be defined as

$$\mathcal{S}\mathcal{P}_m(\Delta) \equiv \mathcal{P}\Pi_m(\Delta) \cap C^{m-1}[0, T],$$

where  $C^{m-1}[0, T]$  is the space of  $m - 1$  time continuously differentiable functions defined on  $[0, T]$ . It is the smoothest space of piecewise polynomials of degree  $m$ . Polynomial spline spaces are finite dimensional linear spaces with very convenient bases. In the ongoing, following [38], we will introduce a basis for  $\mathcal{S}\mathcal{P}_m(\Delta)$ , which is compact support and defines a partition of unity.

Let us define the extended partition  $\tilde{\Delta} = \{\tilde{x}_i\}_{i=0}^{n+2m}$ , associated with  $\Delta$  with the knots  $\tilde{x}_i$  defined as

$$\tilde{x}_0 \leq \dots \leq \tilde{x}_m = 0, T = \tilde{x}_{n+m} \leq \dots \leq \tilde{x}_{n+2m}$$

and

$$\tilde{x}_{m+i} = x_i, i = 1, \dots, n - 1.$$

It should be noted that the interior knots  $\{\tilde{x}_i\}_{i=m}^{m+n}$  are uniquely determined while the first and last  $m$  points can be chosen arbitrarily. Let  $\tilde{\Delta}$  be the extended partition associated with  $\Delta$  and define

$\mathfrak{B}_i(x) = (-1)^{m+1}(\tilde{x}_{m+i+1} - \tilde{x}_i)[\tilde{x}_i, \dots, \tilde{x}_{m+i+1}](x - y)_+^m, 0 \leq i \leq n + m - 1$  where  $(x - y)_+^m$  is the truncated power function

$$(x - y)_+^m = \begin{cases} (x - y)^m & x \geq y \\ 0 & x < y \end{cases},$$

and the divided difference operator  $[\tilde{x}_i, \dots, \tilde{x}_{m+i+1}](x - y)_+^m$  is defined in [38] Definition 2.49. The set of functions  $\{\mathfrak{B}_i(x), 0 \leq i \leq n + m - 1\}$  forms a

basis for the space of spline functions of  $\mathcal{SP}_m(\Delta)$  which is compact support, that is

$$\mathfrak{B}_i(x) = 0, \quad x \notin [\tilde{x}_i, \tilde{x}_{i+m+1}]$$

and

$$\mathfrak{B}_i(x) > 0, \quad x \in (\tilde{x}_i, \tilde{x}_{i+m+1}).$$

Moreover, the basis functions form a partition of unity that is

$$\sum_{i=0}^{n+m-1} \mathfrak{B}_i(x) = 1, \quad \forall x \in [0, T].$$

In another terminology using the cardinal spline function

$$\mathcal{B}_{m+1}(x) = \frac{1}{m!} \sum_{j=0}^{m+1} (-1)^j \binom{m+1}{j} (x-j)_+^m, \tag{2.1}$$

the B-spline basis functions can be constructed as follows [40]

$$\mathfrak{B}_i(x) = \mathcal{B}_{m+1} \left( \frac{x-x_0}{h} - i + 2 \right), \quad i = 0, \dots, n+m-1,$$

which means that

$$\mathcal{SP}_m(\Delta) = \text{span} \left\{ \mathcal{B}_{m+1} \left( \frac{x-x_0}{h} - i + 2 \right), \quad i = 0, \dots, n+m-1 \right\}.$$

Therefore, any spline  $s \in \mathcal{SP}_m(\Delta)$  can be written in the following form

$$s(x) = \sum_{i=0}^{n+m-1} c_i \mathcal{B}_{m+1} \left( \frac{x-x_0}{h} - i + 2 \right),$$

where  $c_i, i = 0, \dots, n+m-1$  are unknown coefficients to be determined. We will define the interpolation conditions on the grid points and midpoints of the partition for odd and even  $m$ , respectively. Let for  $f \in C[a, b]$ ,  $s(x)$  satisfies the interpolatory conditions

$$s(\tau_i) = f(\tau_i), \quad 1 \leq i \leq n, \tag{2.2}$$

for even  $m$ , and

$$s(x_i) = f(x_i), \quad 0 \leq i \leq n, \tag{2.3}$$

for odd  $m$ .

As we know the dimension of  $\mathcal{SP}_m(\Delta)$  is  $n+m$ , thus in order to uniquely determine  $s(x)$ , we need  $m-1$  end conditions for odd  $m$  and  $m$  end conditions for even  $m$  along with interpolatory conditions. To avoid the utilization of derivatives of the solution at the initial point as additional conditions, we introduce the following alternative end conditions:

$$s(x_i) = f(x_i), \quad i \in \{0, 1, \dots, \delta\} \cup \{n-\delta, \dots, n\}, \tag{2.4}$$

for even  $m$  and

$$s(\tau_i) = f(\tau_i), \quad i \in \{1, \dots, \sigma\} \cup \{n-\sigma+1, \dots, n\}, \tag{2.5}$$

for odd  $m$ , where  $\delta = \lfloor \frac{m-1}{2} \rfloor$  and  $\sigma = \lfloor \frac{m}{2} \rfloor$  ( $\lfloor \nu \rfloor$  means the greatest integer smaller than  $\nu$ ).

To establish the well-defined nature of the spline interpolation introduced above, it is necessary to refer to the following Theorem from [41], which provides a sufficient condition for the unique solvability of the resulting system.

**Theorem 2.1.** *The coefficients matrix  $A = (\mathcal{B}_i(x_j))$  of the B-spline, interpolating a function at the grid points  $x_j$  is nonsingular if and only if all diagonal elements  $\mathcal{B}_i(x_i)$  are nonzero.*

At both the grid points and midpoint of the partition, the subsequent Theorem presents a local error estimate for the aforementioned spline interpolation. Before, we need to recall the following Lemma from [42].

**Lemma 2.1.** *Let  $A = \{a_{ij}\}$  be an  $n \times n$  matrix with  $|a_{ii}| \geq \sum_{j=1, j \neq i}^n |a_{ij}| + \varepsilon$ ,  $1 \leq i \leq n$ , where  $\varepsilon > 0$ , then we have  $\|A^{-1}\|_\infty < \varepsilon^{-1}$ .*

**Theorem 2.2.** *Let  $s \in \mathcal{SP}_m(\Delta)$  be the unique spline interpolation for  $f \in C^{m+2}[0, T]$  satisfying interpolatory conditions (2.2)–(2.4) for even  $m$  or (2.3)–(2.5) for odd  $m$ , then*

$$s(x_i) = f(x_i) + O(h^{m+2}), \quad 0 \leq i \leq n, \tag{2.6}$$

if  $m$  is even, and

$$s(\tau_i) = f(\tau_i) + O(h^{m+1}), \quad 1 \leq i \leq n, \tag{2.7}$$

if  $m$  is odd.

*Proof.* We will prove the first relation, since the second one can be proved in a similar manner. Consider the following consistency relation which connect  $s(x)$  at the midpoints and grid points of the partition [40]

$$\sum_{j=-\delta-1}^{\delta+1} l_i s(x_{i+j}) = \sum_{j=-\delta}^{\delta+1} d_i s(\tau_{i+j}), \quad \delta < i < n - \delta, \tag{2.8}$$

where

$$l_i = \mathcal{B}_{m+1} \left( m + \frac{1}{2} - i \right), \quad d_i = \mathcal{B}_{m+1} (m - i). \tag{2.9}$$

Let us define the error function  $e(x) = s(x) - f(x)$ . Subtracting  $\sum_{j=-\delta-1}^{\delta+1} l_i f(x_{i+j})$  from both sides of (2.8), we have

$$\begin{aligned} & \sum_{j=-\delta-1}^{\delta+1} l_i s(x_{i+j}) - \sum_{j=-\delta-1}^{\delta+1} l_i f(x_{i+j}) \\ &= \sum_{j=-\delta}^{\delta+1} d_i s(\tau_{i+j}) - \sum_{j=-\delta-1}^{\delta+1} l_i f(x_{i+j}), \quad \delta < i < n - \delta. \end{aligned}$$

which using the definition of  $e(x)$  the left hand side can be written as  $\sum_{j=-\delta-1}^{\delta+1} l_i e(x_{i+j})$  and by the interpolatory conditions (2.2),  $s(\tau_k)$  in the right hand side can be replaced by  $f(\tau_k)$  to obtain

$$\sum_{j=-\delta-1}^{\delta+1} l_i e(x_{i+j}) = \sum_{j=-\delta}^{\delta+1} d_i f(\tau_{i+j}) - \sum_{j=-\delta-1}^{\delta+1} l_i f(x_{i+j}), \quad \delta < i < n - \delta.$$

By the assumptions  $f \in C^{m+2}[0, T]$ , so  $M_1 = \max_{0 \leq x \leq T} |f^{(m+2)}(x)| < \infty$ . By expanding  $f$  in both terms in the right hand side of the relation above, using Taylor series expansion about  $x_i$  and simplifying we obtain

$$\sum_{j=-\delta-1}^{\delta+1} l_i e(x_{i+j}) = O(M_1 h^{m+2}) = O(h^{m+2}), \quad \delta < i < n - \delta. \quad (2.10)$$

The system above contains  $n - m + 1$  equations with  $n + 1$  unknown  $e(x_i)$ ,  $0 \leq i \leq n$ . Using (2.4) and (2.10) together, we obtain the following  $(n + 1) \times (n + 1)$  system

$$\begin{cases} e(x_0) = e(x_1) = \dots = e(x_\delta) = 0, \\ \sum_{j=-\delta-1}^{\delta+1} l_i e(x_{i+j}) = O(h^{m+2}), & \delta < i < n - \delta, \\ e(x_{n-\delta}) = \dots = e(x_{n-1}) = e(x_n) = 0. \end{cases} \quad (2.11)$$

Obviously, certain elements within the above system are zero. By eliminating the unknowns corresponding to these zero elements, the size of the system can be reduced. Consequently, the system can be reformulated in its matrix form as shown below:

$$AE = O(h^{m+2}) \quad (2.12)$$

where

$$E = [e(x_{\delta+1}), e(x_{\delta+2}), \dots, e(x_{n-\delta-1})]^T$$

and

$$A = \begin{pmatrix} l_\delta & l_{\delta+1} & \dots & l_m & & & & & \\ l_{\delta-1} & l_\delta & \dots & l_{m-1} & l_m & & & & \\ & & \ddots & & & & & & \\ l_0 & l_1 & \dots & l_\delta & \dots & l_{m-1} & l_m & & \\ & & & & \ddots & & & & \\ & & & l_0 & l_1 & \dots & l_\delta & \dots & l_{m-1} & l_m \\ & & & & & \ddots & & & & \\ & & & & & & l_0 & l_1 & \dots & l_\delta & l_{\delta+1} \\ & & & & & & & l_0 & l_1 & \dots & l_\delta \end{pmatrix}. \quad (2.13)$$

According to Theorem 2.1, the coefficients matrix  $A$  is nonsingular if all the diagonal elements are nonzero. Specifically, all the diagonal elements of  $A$  are equal to  $l_\delta$  which possesses the following value

$$l_\delta = \mathcal{B}_{m+1} \left( m + \frac{1}{2} - \delta \right).$$

Since  $m + \frac{1}{2} - \delta \in \text{support}(\mathcal{B}_{m+1})$  thus  $l_\delta \neq 0$  and the matrix  $A$  is nonsingular. Therefore, from (2.12), we obtain

$$\|E\| = \|A^{-1}\| O(h^{m+2}). \quad (2.14)$$

For  $m \leq 6$ , the system is strictly diagonally dominant and using Lemma 2.1 we have  $\|A^{-1}\|_\infty < 46$  thus, obviously in this case the proof is complete. Let  $m > 6$ , it can be readily verified that for each B-spline basis function,

all  $l_i$  values are positive and  $\sum_{i=0}^m l_i = 1$ , so  $\|\mathcal{A}\|_1 = 1$  and  $\|\mathcal{A}\|_\infty = 1$ . By leveraging the properties of the B-spline basis function  $\mathcal{B}_{m+1}$ , it results

$$l_i = l_{m-i}, \quad i = 0, \dots, \frac{m}{2},$$

thus,  $\mathcal{A}$  is a symmetric matrix. On the other hand, the nonsingularity of  $\mathcal{A}$  implies that all of its eigenvalues are nonzero. Specifically, we have  $\mu(\mathcal{A}) \neq 0$ , where  $\mu$  represents smallest absolute of eigenvalue of  $\mathcal{A}$ . Drawing upon concepts from linear algebra, we have

$$\|\mathcal{A}\|_2 = \sqrt{\rho(\mathcal{A}^T \cdot \mathcal{A})} = \rho(\mathcal{A}) \tag{2.15}$$

where  $\rho$  represents the largest absolute eigenvalue of  $\mathcal{A}$ . On the other hand using the Gershgorin's Theorem, all eigenvalues  $\lambda$  of  $\mathcal{A}$  must lie within the Gershgorin's discs

$$|\lambda - l_\delta| \leq \sum_{i=0, i \neq \delta}^m |l_i|$$

which results

$$-1 < l_\delta - \sum_{i=0, i \neq \delta}^m |l_i| \leq \lambda \leq l_\delta + \sum_{i=0, i \neq \delta}^m |l_i| = 1 \Rightarrow \rho(\mathcal{A}) \leq 1.$$

It is noteworthy that the numerical experiments show that for fixed  $m$ , by varying  $n$ , we have  $\rho(\mathcal{A}) \neq 1$  which means that  $\|\mathcal{A}\|_2 < 1$ . Consequently, based on the principles of linear algebra for symmetric matrices, we have

$$\|\mathcal{A}\|_2 \|\mathcal{A}^{-1}\|_2 = \frac{\rho(\mathcal{A})}{\mu(\mathcal{A})} \tag{2.16}$$

which means that  $\|\mathcal{A}^{-1}\|_2$  cannot be infinite. Let denote by  $M_2$  the upper bound of  $\|\mathcal{A}^{-1}\|_2$ , then substituting into (2.14) we obtain

$$\|E\|_\infty \leq \|E\|_2 = \|\mathcal{A}^{-1}\|_2 O(h^{m+2}) = M_2 O(h^{m+2}) = O(h^{m+2}). \tag{2.17}$$

This together with (2.4) complete the proof of (2.6). □

### 3. Numerical Method

In this section, we present a cutting-edge numerical methodology centered around the innovative utilization of B-spline methods for solving integral equations. The proposed approach offers a robust and computationally efficient framework for tackling integral equations. By employing B-splines to represent the unknown functions in the integral equation, we can construct a concise and highly accurate representation, leading to a significant reduction in computational effort while preserving solution accuracy. Throughout this section, we detail the theoretical foundation, algorithmic implementation, and numerical consideration of our B-spline-based method. We further substantiate the effectiveness and superiority of our approach through comprehensive examples including nonlinear integral equations in Sect. 5.



The operator representation of the integral Eq. (1.1) can be illustrated as follows

$$u(x) = \mathcal{K}u(x) + g(x), \tag{3.1}$$

where  $\mathcal{K} : C^{m+1} \rightarrow C^{m+1}$  is a nonlinear integral operator in the following form:

$$\mathcal{K}u(x) = \int_0^x G(x, t, u(t)) dt, \quad x \in [0, T]. \tag{3.2}$$

Assuming the existence and uniqueness of the solution to Eq. (3.1), our objective is to employ an  $m$ -th degree B-spline of the following form

$$s(x) = \sum_{j=0}^{n+m-1} c_j \mathcal{B}_{m+1} \left( \frac{x - x_0}{h} - j + 2 \right),$$

to approximate the solution.

### 3.1. Even Degree B-Spline

Let  $m$  be even. Upon replacing  $u(x)$  with  $s(x)$  in Eq. (3.1), we take the approach of collocating the obtained equation at the midpoints of the partition, we have

$$s(\tau_i) = \int_0^{\tau_i} G(\tau_i, t, s(t)) dt + g(\tau_i), \quad 1 \leq i \leq n. \tag{3.3}$$

Then employing the B-spline basis functions at points  $\tau_i$ , the Eq. (3.3) can be expressed as

$$\begin{aligned} & \sum_{j=0}^{n+m-1} c_j \mathcal{B}_{m+1} \left( \frac{\tau_i - x_0}{h} - j + 2 \right) \\ &= \int_0^{\tau_i} G \left( \tau_i, t, \sum_{j=0}^{n+m-1} c_j \mathcal{B}_{m+1} \left( \frac{t - x_0}{h} - j + 2 \right) \right) dt + g(\tau_i), \end{aligned} \tag{3.4}$$

for  $1 \leq i \leq n$ . The system described above consists of  $n$  equations and  $m + n$  unknowns. In order to ensure the unique determination of the spline approximation, we require an additional  $m$  equations. To this end, we employ the collocation equation at  $m$  boundary and near-boundary grid points in the following manner

$$s(x_i) = \int_0^{x_i} G(x_i, t, s(t)) dt + g(x_i), \quad i \in \{0, 1, \dots, \delta\} \cup \{n - \delta, \dots, n\}. \tag{3.5}$$

Upon utilizing the expressive B-spline basis functions, we can express Eq. (3.5) as follows

$$\begin{aligned} & \sum_{j=0}^{n+m-1} c_j \mathcal{B}_{m+1} \left( \frac{x_i - x_0}{h} - j + 2 \right) \\ &= \int_0^{x_i} G \left( x_i, t, \sum_{j=0}^{n+m-1} c_j \mathcal{B}_{m+1} \left( \frac{t - x_0}{h} - j + 2 \right) \right) dt + g(x_i), \end{aligned} \tag{3.6}$$

for  $i \in \{0, 1, \dots, \delta\} \cup \{n - \delta, \dots, n\}$ . The relation (3.4) together with (3.6) yields a system of  $n+m$  equations involving  $n+m$  unknowns  $c_j, j = 0, \dots, n+$

$m - 1$ , which must be diligently solved to define  $s(x)$ . The solution to this system will lead us to the desired approximation of the function  $s(x)$  using proposed B-spline basis functions.

To effectively handle the integrals in (3.4) and (3.6), we employ the  $q$ -point Gauss–Legendre quadrature method. This numerical integration technique is well-suited for accurately approximating integrals over specific intervals by distributing  $q$  quadrature’s points in a way that optimally captures the underlying function’s behavior [39]. By introducing the notation  $z(t) = G\left(\tau_i, t, \sum_{j=0}^{n+m-1} c_j \mathcal{B}_{m+1}\left(\frac{t-x_0}{h} - j + 2\right)\right)$  and change of variable, the integrals in (3.4) can be reformulated as follows

$$\begin{aligned} I_i &= \int_0^{\tau_i} z(t)dt = \int_0^{\tau_1} z(t)dt + \int_{\tau_1}^{\tau_i} z(t)dt \\ &= \frac{\tau_1}{2} \int_{-1}^1 z\left(\frac{\tau_1}{2}(\eta + 1)\right) d\eta + \sum_{r=1}^{i-1} \frac{\tau_{r+1} - \tau_r}{2} \int_{-1}^1 z\left(\frac{\tau_{r+1} - \tau_r}{2}\eta + \frac{\tau_{r+1} + \tau_r}{2}\right) d\eta. \end{aligned}$$

Now by incorporating the Gauss–Legendre quadrature into our system of equations, we can efficiently compute the unknown coefficients  $c_j$ , ultimately leading to a highly accurate approximation of  $s(x)$ . We end up with

$$\begin{aligned} I_i &= \frac{\tau_1}{2} \sum_{k=1}^q \vartheta_k z\left(\frac{\tau_1}{2}(\eta_k + 1)\right) \\ &\quad + \sum_{r=1}^{i-1} \frac{\tau_{r+1} - \tau_r}{2} \sum_{k=1}^q \vartheta_k z\left(\frac{\tau_{r+1} - \tau_r}{2}\eta_k + \frac{\tau_{r+1} + \tau_r}{2}\right), \end{aligned} \tag{3.7}$$

where  $\vartheta_k$  and  $\eta_k$  represent the weights and abscissas associated with  $q$ -point Gauss–Legendre integrating formula. In a same manner by introducing the notation  $\tilde{z}(t) = G\left(x_i, t, \sum_{j=0}^{n+m-1} c_j \mathcal{B}_{m+1}\left(\frac{t-x_0}{h} - j + 2\right)\right)$  the integrals in (3.6) can be approximated as follows

$$\tilde{I}_i = \int_0^{x_i} \tilde{z}(t)dt = \sum_{r=0}^{i-1} \frac{x_{r+1} - x_r}{2} \sum_{k=1}^q \vartheta_k \tilde{z}\left(\frac{x_{r+1} - x_r}{2}\eta_k + \frac{x_{r+1} + x_r}{2}\right). \tag{3.8}$$

### 3.2. Odd Degree B-Spline

For odd  $m$ , we collocate the problem at grid points of the partition as follows

$$\begin{aligned} &\sum_{j=0}^{n+m-1} c_j \mathcal{B}_{m+1}\left(\frac{x_i - x_0}{h} - j + 2\right) \\ &= \int_0^{x_i} G\left(x_i, t, \sum_{j=0}^{n+m-1} c_j \mathcal{B}_{m+1}\left(\frac{t - x_0}{h} - j + 2\right)\right) dt + g(x_i), \end{aligned} \tag{3.9}$$

for  $0 \leq i \leq n$ . Obviously, the above system of equations requires  $m - 1$  extra relations to be uniquely solvable. In this case, the collocated problem

$$\begin{aligned} & \sum_{j=0}^{n+m-1} c_j \mathcal{B}_{m+1} \left( \frac{\tau_i - x_0}{h} - j + 2 \right) \\ &= \int_0^{\tau_i} G \left( \tau_i, t, \sum_{j=0}^{n+m-1} c_j \mathcal{B}_{m+1} \left( \frac{t - x_0}{h} - j + 2 \right) \right) dt + g(\tau_i), \end{aligned} \tag{3.10}$$

can be utilized at the near boundary mid points of the partition for  $i \in \{1, \dots, \sigma\} \cup \{n - \sigma + 1, \dots, n\}$ . It is worth noting that the integrals in (3.9) and (3.10) can be approximated in the same manner as (3.7) and (3.8).

### 4. Convergence Analysis and Error Estimation

#### 4.1. Spline Approximation Error

In this section, we embark on an in-depth exploration of the error estimation and convergence rates pertaining to the scheme proposed within this paper. Our objective is to rigorously assess the accuracy and convergence properties of the introduced methodology. We constructed a collocation method based on B-splines for the solution of

$$u = \mathcal{K}u + g \tag{4.1}$$

where  $\mathcal{K}$  is the Volterra integral operator (3.2). We will prove the convergence for even degree splines and the proof is similar for odd degrees. Therefore, let  $m$  be even, in order to achieve an error bound and analyze the convergence of the proposed method, we define the operator  $\mathcal{P}_n u : C^{m+1} \rightarrow \mathcal{SP}_m(\Delta)$  such that, for any function  $\alpha \in C^{m+1}$  we have

$$\begin{cases} (\mathcal{P}_n \alpha)(x_i) = \alpha(x_i), & i = 0, 1, \dots, \delta, \\ (\mathcal{P}_n \alpha)(\tau_i) = \alpha(\tau_i), & 1 \leq i \leq n, \\ (\mathcal{P}_n \alpha)(x_i) = \alpha(x_i), & i = n - \delta, \dots, n. \end{cases} \tag{4.2}$$

Rewriting (3.3) and (3.5), we have

$$\begin{aligned} s(x_i) &= \int_0^{x_i} G(x_i, t, s(t)) dt + g(x_i), & i \in \{0, 1, \dots, \delta\} \\ s(\tau_i) &= \int_0^{\tau_i} G(\tau_i, t, s(t)) dt + g(\tau_i), & 1 \leq i \leq n, \\ s(x_i) &= \int_0^{x_i} G(x_i, t, s(t)) dt + g(x_i), & i \in \{n - \delta, \dots, n\} \end{aligned}$$

which by means of operator  $\mathcal{P}_n$  and (4.1) can be written in the operator form as

$$\mathcal{P}_n s = \mathcal{P}_n \mathcal{K} s + \mathcal{P}_n g.$$

Also  $s \in \mathcal{SP}_m(\Delta)$ , thus  $\mathcal{P}_n s = s$  and we have

$$s = \mathcal{P}_n \mathcal{K} s + \mathcal{P}_n g. \tag{4.3}$$

**Lemma 4.1.** *The operator  $\mathcal{P}_n$  is uniformly bounded in  $C^k[0, T]$ .*

*Proof.* Using [38], Theorem 6.22, for every  $\alpha \in C[a, b]$ , we have

$$\|\mathcal{P}_n\alpha\|_{C[a,b]} \leq (2m)^m \|\alpha\|_{C[a,b]},$$

so  $\mathcal{P}_n$  is uniformly bounded. □

**Lemma 4.2.** *The sequence of operators  $\mathcal{P}_n$  uniformly converges to identity operator.*

*Proof.* Let  $\alpha \in C[0, T]$ , then according to Lebesgue Lemma, we have

$$\|\mathcal{P}_n\alpha - \alpha\| \leq (1 + \|\mathcal{P}_n\|) \inf_{u \in \Pi_m} \|\alpha - u\|.$$

But the right hand side of the above inequality represents the error of best approximation for  $\alpha$  in  $\Pi_m$ . Suppose that  $u^*$  be the best approximation for  $f$  then using Jackson’s Theorem in each interval  $[\tau_i, \tau_{i+1}]$  we have

$$\sup_{\tau_i \leq x \leq \tau_{i+1}} |\alpha(x) - u^*(x)| \leq 6\omega\left(\alpha, \frac{h}{2m}\right).$$

Thus we have

$$\|\mathcal{P}_n\alpha - \alpha\| \leq 6(1 + \|\mathcal{P}_n\|)\omega\left(\alpha, \frac{h}{2m}\right) \rightarrow 0,$$

and the proof is complete. □

In fact from Theorem 2.2 for  $\alpha \in C^{m+1}[0, T]$  we have

$$\|\mathcal{P}_n\alpha - \alpha\| = O(h^{m+1})$$

and if  $\alpha \in C^{m+2}[0, T]$ , for even  $m$  we have the following local bound

$$|\mathcal{P}_n\alpha - \alpha|_{x_i} = O(h^{m+2}).$$

We will show that (4.3) has a unique solution that converges to the solution of (4.1) and finally, we will obtain the rate of convergence. Let us restate the following Theorem from [43].

**Theorem 4.1.** *Suppose that operators  $T$  and  $T_n$  on the Banach space  $B$  can be represented as*

$$T = PK, \quad T_n = P_nK$$

where  $K$  is a nonlinear, completely continuous operator mapping  $B$  into another Banach space  $B'$  and  $P$  and  $P_n$  are continues linear operators taking  $B'$  into  $B$ . Suppose that the operator equation

$$w = PKw \tag{4.4}$$

has a solution  $v$ . A sufficient condition that  $v$  be an isolated solution of (4.4) in some sphere  $\|\omega - v\| \leq \mu$  ( $\mu > 0$ ) is that  $K$  be differentiable at the point  $v$  the homogeneous equation

$$w - PK'(v)w = 0, \tag{4.5}$$

has only the trivial solution  $w = 0$ . Suppose further that the sequence of operators  $P_n$  converges strongly to the operator  $P$ , then the equation

$$w = T_n w \tag{4.6}$$

has a solution  $v_n$  satisfying  $\|v_n - v\| \leq \mu$  for all sufficiently large  $n$ ,  $v_n \rightarrow v$  as  $n \rightarrow \infty$ , and the rate of convergence is bounded by

$$\|v_n - v\| \leq M' \|(P_n - P)Kv\| \quad (M' = \text{constant}). \tag{4.7}$$

*Proof.* See [43]. □

**Theorem 4.2.** Let for even  $m$ ,  $u \in C^{m+1}[a, b]$  and  $s \in \mathcal{SP}_m(\Delta)$  be the exact and spline solutions of problem (1.1), respectively. If  $s$  is obtained by (3.4)–(3.6), then it converges to  $u$  as  $n$  tends to infinity and for some constant  $C$ , we have

$$\|s - u\| \leq C h^{m+1}, \quad h \rightarrow 0.$$

Also, if  $u \in C^{m+2}[a, b]$ , the following local error bound holds

$$|s - u|_{x_i} = O(h^{m+2}), \quad x_i \in \Delta.$$

*Proof.* Since  $\mathcal{P}_n \rightarrow I$  thus we have  $\mathcal{P}_n \mathcal{K} \rightarrow \mathcal{K}$  and the assumptions of Theorem 4.1 are satisfied. Hence, the collocation equation

$$s = \mathcal{P}_n \mathcal{K} s + \mathcal{P}_n g$$

has a solution  $s$  for which  $\|s - u\| \rightarrow 0$ . Also, using Theorem 4.1 we have the following bound

$$\|s_n - u\| \leq M' \|(P_n - I)Ku\|. \tag{4.8}$$

where  $M'$  is some finite constant. On the other hand, we have

$$\|\mathcal{P}_n g - g\| = O(h^{m+1})$$

which combining with complete continuity of  $\mathcal{K}$  gives

$$\|s - u\| = O(h^{m+1}).$$

Also since  $|\mathcal{P}_n g - g|_{x_i} = O(h^{m+2})$ , it results that

$$|s - u|_{x_i} = O(h^{m+2}), \quad x_i \in \Delta.$$

□

### 4.2. Effect of Quadrature Rule on Error

We used a  $q$ -point Gauss–Legendre quadrature method to approximate integrals arose in the collocation system. It is well known that the error associated with  $q$ -point Gauss–Legendre quadrature for  $f \in C^{2q}[a, b]$  is bounded by [44]

$$E(f) = \frac{(b - a)^{2n+1} (n!)^4}{(2n + 1)(2n!)^3} f^{(2q)}(\theta), \quad a < \theta < b.$$

Since we used quadrature in each subinterval thus  $b - a \leq h$ . Having that for  $q$  in range, the term  $\frac{(n!)^4}{(2n+1)(2n!)^3}$  remains bounded, thus

$$E(f) = O(h^{2q+1}),$$

for small  $q$ . On the other hand, we proved that the error of the collocation system is  $O(h^{m+1})$  for odd  $m$  and  $O(h^{m+2})$  for even  $m$ . Thus,  $q$  should be chosen such that it satisfies

$$\begin{cases} 2q + 1 \geq m + 1, & \text{odd } m, \\ 2q + 1 \geq m + 2, & \text{even } m. \end{cases}$$

While the Newton-Cotes quadrature can be employed to approximate integrals in the collocation system, it is worth noting that its accuracy is inferior to Gauss quadrature method. Consequently, a greater number of quadrature nodes will be required for satisfactory results.

## 5. Some Test Problems

In this section, we solve some test problems of second kind Volterra integral equations to show the applicability of the proposed method. Two examples with smooth kernel and one with non-smooth kernel are solved. The acquired results are compared with various existing methods to demonstrate the efficacy of our approach. The maximum absolute errors and the experimental orders of convergence are presented in Tables 1, 2, 3, 4, 5, 6, 7. The experimental orders are calculated by

$$Order = \frac{\log\left(\frac{E_{n_1}}{E_{n_2}}\right)}{\log\left(\frac{n_2}{n_1}\right)}$$

where  $E_n$  denotes the maximum absolute error obtained with  $n+1$  collocation points. All the programs are written in Mathematica 12, and run on a system with Intel Core i7-2670 2.20 GHz and 8 GB of RAM.

*Example 5.1.* As first test problem consider the linear second kind Volterra integral equation

$$u(x) = \bar{\lambda} \int_0^x u(t)dt + f(x), \quad x \in [0, 1],$$

with the exact solution  $u(x) = \frac{1}{2}(\sin(x) + \cos(x) + e^x)$ , where  $\bar{\lambda}$  is a constant parameter. In [30] and [31], it is show that step-by-step spline collocation procedure based on quadratic and cubic spline may diverge for some choices of collocation parameters. It is shown that the nonlocal approach based on spline can be better comparing to step-by-step algorithm. Here, we solve the test problem for various values of  $\bar{\lambda}$  using quadratic and cubic spline approach and compared the results with splines in [30] and [31]. The results are tabulated in Tables 1 and 2. It is obvious that using our approach more accurate results can be achieved. Also, the method is stable and the solution remains bounded. We also run the problem in larger domain  $[0, 5]$  and tabulated the results in Table 3. The results show that the method is still accurate and remains stable in larger domains.

Table 1. Maximum absolute errors using quadratic spline,  $m = 2$  for example 5.1 in  $[0,1]$

Step-by-step ([31])	$n = 4$	$n = 8$	$n = 16$	$n = 32$	$n = 64$
$\bar{\lambda} = -2$	3.28(-3)	—	98.02	—	1.1(+27)
$\bar{\lambda} = -1$	2.59(-3)	—	76.56	—	8.5(+26)
$\bar{\lambda} = 1$	1.46(-3)	—	45.99	—	5.1(+26)
$\bar{\lambda} = 2$	1.02(-3)	—	35.36	—	4.0(+26)
Nonlocal ([31])					
$\bar{\lambda} = -2$	3.27(-3)	—	6.27(-5)	—	1.02(-6)
$\bar{\lambda} = -1$	3.11(-3)	—	6.18(-5)	—	1.02(-6)
$\bar{\lambda} = 1$	2.77(-3)	—	5.99(-5)	—	1.01(-6)
$\bar{\lambda} = 2$	2.60(-3)	—	5.91(-5)	—	1.01(-6)
Our Method					
$\bar{\lambda} = -2$	3.72(-5)	2.40(-6)	1.54(-7)	9.76(-9)	6.14(-10)
$\bar{\lambda} = -1$	4.74(-5)	3.07(-6)	1.96(-7)	1.24(-8)	7.81(-10)
$\bar{\lambda} = 1$	9.46(-5)	6.72(-6)	4.48(-7)	2.89(-8)	1.83(-09)
$\bar{\lambda} = 2$	1.47(-4)	1.18(-5)	8.35(-7)	5.55(-8)	3.58(-09)

Table 2. Maximum absolute errors using cubic spline,  $m = 3$  for example 5.1 in  $[0,1]$

Cubic spline [30]	$n = 4$	$n = 8$	$n = 16$	$n = 32$	$n = 64$
$\lambda = 2$	1.65(-4)	—	6.81(-7)	—	2.70(-9)
$\lambda = -1$	1.94(-4)	—	8.83(-7)	—	3.48(-9)
Our Method					
$\lambda = 2$	2.09(-5)	2.12(-6)	1.57(-7)	1.06(-8)	6.88(-10)
$\lambda = -1$	3.90(-6)	3.00(-7)	2.01(-8)	1.29(-9)	8.23(-11)

Table 3. Maximum absolute errors using cubic spline,  $m = 3$  for example 5.1 in  $[0,5]$

Quadratic spline	$n = 16$	$n = 32$	$n = 64$	$n = 128$	$n = 256$
$\lambda = -2$	1.44(00)	9.66(-2)	6.20(-3)	3.92(-4)	2.46(-5)
$\lambda = -1$	1.91(-2)	1.43(-3)	9.80(-5)	6.40(-6)	4.09(-7)
$\lambda = 1$	3.30(-3)	2.39(-4)	1.61(-5)	1.04(-6)	6.64(-8)
$\lambda = 2$	2.59(-3)	1.89(-4)	1.28(-5)	8.33(-7)	5.31(-8)
Cubic spline					
$\lambda = -2$	1.95(00)	2.32(-2)	1.63(-3)	1.07(-4)	6.96(-6)
$\lambda = -1$	4.84(-3)	3.37(-4)	2.20(-5)	1.41(-6)	8.90(-8)
$\lambda = 1$	3.97(-4)	2.77(-5)	1.81(-6)	1.16(-7)	7.32(-9)
$\lambda = 2$	5.52(-4)	3.80(-5)	2.46(-6)	1.56(-7)	9.83(-9)

Table 4. Number of collocation points to obtain the tolerance with  $m = 3$  for example 5.2

Error	[45], Meth. 1	[45], Meth. 2	[45], Meth. 3	[46]	Our Meth.
$10^{-9}$	110	74	45	110	28
$10^{-11}$	350	234	130	350	91
$10^{-13}$	850	739	400	–	301

*Example 5.2.* Consider the following nonlinear second kind Volterra integral equation

$$u(x) = \phi(x, u(x)) \int_0^x \Phi(x, t, u(t))dt + f(x), \quad x \in [0, 1],$$

where  $\Phi(x, t, u(t)) = xt + u(t)$  and  $\phi(x, t, u(x)) = \frac{x}{1+x} \ln(1 + u(x))$ . The exact solution is  $u(x) = \frac{1}{10}x^{10}$ . The problem is already solved using cubic spline [45] and a method based on mean-value Theorem [46]. We solved the problem using cubic spline approach. The number of collocation points for reaching the tolerance are tabulated in Table 4. Obviously, our method may require fewer collocation points comparing to the existing methods.

*Example 5.3.* Consider the following second kind singular cordial Volterra integral equation

$$u(x) = \int_0^x x^{-1} \varphi\left(\frac{t}{x}\right)u(t)dt + f(x), \quad x \in [0, 1],$$

with  $\varphi(t/x) = (t/x)^{\xi-1}$  and  $\xi > 0$ . In [43], for  $\varphi \in L^1[0, 1]$ , it is proved that  $V_\varphi \in \mathcal{L}(C^m)$  where  $V_\varphi$  is the cordial integral operator

$$V_\varphi = \int_0^x x^{-1} \varphi\left(\frac{t}{x}\right)u(t)dt$$

and  $\mathcal{L}(C^m)$  is the space of linear bounded operators from  $C^m$  to  $C^m$ . So the new approach can be used successfully without any modifications.

We solved this problem for various kinds of exact solutions. The problem is solved for  $\xi = \frac{3}{2}$  using various degrees of splines and maximum absolute errors and orders of convergence are tabulated in tables 4-7. Based on Theorem 4.2, the order of convergence depends on the regularity of the exact solution. The results in tables 4-7 verify the theoretical results as well. Also, it is obvious that using even degree splines we obtain super-convergent approximations.

*Example 5.4.* Consider the following nonlinear second kind Volterra integral equation

$$u(x) = \int_0^x 3 \sin(t - x)u^2(x)dt + 1 + \sin^2(x), \quad x \in [0, T],$$

with the exact solution  $u(x) = \cos(x)$ . The problem is already solved by a cubic spline method with standard end conditions in [47]. In Table 9, we



Table 5. Maximum absolute errors for example 5.3 with  $u(x) = x^3 + x^{3.5}$

$n$	$m = 1$	Order	$m = 2$	Order	$m = 3$	Order	$m = 4$	Order
8	1.00(-2)		1.69(-4)		4.98(-6)		6.94(-07)	
16	2.65(-3)	1.92	1.78(-5)	3.24	4.40(-7)	3.50	6.13(-08)	3.50
32	6.75(-4)	1.97	1.94(-6)	3.20	3.89(-8)	3.50	5.42(-09)	3.50
64	1.69(-4)	1.99	2.17(-7)	3.16	3.44(-9)	3.50	4.79(-10)	3.50

Table 6. Maximum absolute errors for example 5.3 with  $u(x) = x^4 + x^{4.5}$

$n$	$m = 1$	Order	$m = 2$	Order	$m = 3$	Order	$m = 4$	Order
8	1.29(-2)		3.03(-4)		2.60(-5)		8.23(-08)	
16	3.26(-3)	1.98	1.97(-5)	3.94	1.87(-6)	3.80	3.63(-09)	4.50
32	8.17(-4)	1.99	1.25(-6)	3.97	1.25(-7)	3.89	1.60(-10)	4.50
64	2.04(-4)	1.99	7.96(-8)	3.98	8.19(-9)	3.93	7.09(-12)	4.49

Table 7. Maximum absolute errors for example 5.3 with  $u(x) = x^5 + x^{5.5}$

$n$	$m = 1$	Order	$m = 2$	Order	$m = 3$	Order	$m = 4$	Order
8	1.54(-2)		8.64(-4)		5.47(-5)		1.91(-06)	
16	3.87(-3)	1.99	5.83(-5)	3.88	3.73(-6)	3.87	4.72(-09)	5.34
32	9.68(-4)	1.99	3.78(-6)	3.94	2.41(-7)	3.94	1.19(-09)	5.30
64	2.42(-4)	1.99	2.40(-7)	3.97	1.53(-8)	3.97	3.12(-11)	5.25

Table 8. Maximum absolute errors for example 5.3 with  $u(x) = x^6 + x^{6.5}$

$n$	$m = 1$	Order	$m = 2$	Order	$m = 3$	Order	$m = 4$	Order
8	1.79(-2)		1.93(-3)		8.72(-5)		7.02(-06)	
16	4.49(-3)	1.99	1.37(-4)	3.80	6.28(-6)	3.79	1.15(-08)	5.92
32	1.12(-3)	1.99	9.19(-6)	3.90	4.17(-7)	3.91	1.85(-09)	5.96
64	2.81(-4)	1.99	5.92(-7)	3.95	2.67(-8)	3.96	2.93(-11)	5.98

compared the results obtained by our cubic spline method ( $m = 3$ ) with those reported in [47]. It is obvious that our results are more accurate than the standard cubic spline. Also, we solved the problems in the larger domain  $[0, T], T = 2, 3, 4, 5$ , and tabulated the maximum absolute errors in Table 10. The results show that the method is applicable for larger domains as well.

Table 9. Absolute errors using cubic spline,  $m = 3$  for example 5.4 in  $[0,1]$

Error	Our Method [47]		Our Method [47]		Our Method [47]	
	n=10	n=10	n=20	n=20	n=40	n=40
0.1	1.63(- 9)	1.01(- 5)	3.12(- 11)	1.59(- 6)	8.66(- 12)	2.10(- 7)
0.2	2.01(- 9)	2.48(- 5)	5.46(- 10)	3.26(- 6)	4.82(- 11)	4.19(- 7)
0.3	1.42(- 8)	3.65(- 5)	1.52(- 09)	4.72(- 6)	1.15(- 10)	5.99(- 7)
0.4	3.21(- 8)	4.61(- 5)	2.86(- 09)	5.87(- 6)	2.04(- 10)	7.39(- 7)
0.5	5.66(- 8)	5.26(- 5)	4.46(- 09)	6.63(- 6)	3.08(- 10)	8.32(- 7)
0.6	8.38(- 8)	5.59(- 5)	6.18(- 09)	6.98(- 6)	4.17(- 10)	8.72(- 7)
0.7	1.14(- 7)	5.58(- 5)	7.88(- 09)	6.92(- 6)	5.23(- 10)	8.61(- 7)
0.8	1.39(- 7)	5.28(- 5)	9.46(- 09)	6.47(- 6)	6.20(- 10)	8.02(- 7)
0.9	1.51(- 7)	4.65(- 5)	1.08(- 08)	5.70(- 6)	7.00(- 10)	7.02(- 7)

Table 10. Maximum absolute errors for example 5.4 in different intervals  $[0, T]$

$n$	$[0, 2]$	Order	$[0, 3]$	Order	$[0, 4]$	Order	$[0, 5]$	Order
8	8.07(-06)		2.43(-4)		8.38(-3)		7.51(-2)	
16	4.93(-07)	4.03	1.02(-5)	4.56	3.63(-4)	4.52	3.61(-3)	4.37
32	3.13(-08)	3.97	4.72(-7)	4.44	1.65(-5)	4.45	1.60(-4)	4.49
64	1.97(-09)	3.98	2.40(-8)	4.29	8.41(-7)	4.30	7.91(-6)	4.34
128	1.22(-10)	3.99	1.32(-9)	4.17	4.63(-8)	4.18	4.27(-7)	4.21

### 6. Conclusion

The paper has been concerned with the application of B-splines for the solution of second kind Volterra integral equations. It is already known from literature that using splines with maximum order of regularity for the solution of second kind Volterra integral equations may lead to unstable approximations depend on the choice of collocation parameters [30,31]. We developed a method based on B-splines for the second kind nonlinear Volterra integral equations which uses the grid points and mid points of the uniform partition as collocation points. The method is stable and convergent. Also, we found a super-convergence using even degree splines. We used the method for some test problems for which the step-by-step and nonlocal spline approaches are unstable. It is obvious from the obtained results that the method efficient and the theory is supported by numerical experiments as well. It is noteworthy that the method presented in this paper can be readily applied for approximating solutions to Fredholm integral equations, as well as integro-differential equations.

**Author contributions** M. Ghasemi and A. Goligerdian wrote the main manuscript text and edit the text. M. Ghasemi and S. Moradi run the programs and did the numerical tests. All authors wrote the theoretical part. All authors reviewed the manuscript.

**Funding** This research is not supported by any specific funding.

**Availability of data and materials** Not applicable.

## Declarations

**Conflict of interest** The authors declare no conflict of interest.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

## References

- [1] Linz, P.: *Analytical and Numerical Methods for Volterra Equations*. SIAM, Philadelphia (1985)
- [2] Wazwaz, A.M.: *Linear and Nonlinear Integral Equations: Methods and Applications*. Springer, Heidelberg (2011)
- [3] Atkinson, K., Han, W.: *Theoretical Numerical Analysis: A Functional Analysis Framework*. Texts in Applied Mathematics, vol. 39. Springer, Dordrecht (2009)
- [4] Baker, C.T.H.: *The Numerical Treatment of Integral Equations*. Monographs on Numerical Analysis, Clarendon Press, Oxford (1977)
- [5] Laurita, C., De Bonis, M.C., Sagaria, V.: A numerical method for linear volterra integral equations on infinite intervals and its application to the resolution of metastatic tumor growth models. *Appl. Numer. Math.* **172**, 475–496 (2022)
- [6] Ghaderi Aram, M., Beilina, L., Karchevskii, E.M.: An adaptive finite element method for solving 3d electromagnetic volume integral equation with applications in microwave thermometry. *J. Comput. Phys.* **459**, 111122 (2022)
- [7] Klein, M., Ghai, S.K., Ahmed, U., Chakraborty, N.: Energy integral equation for premixed flame-wall interaction in turbulent boundary layers and its application to turbulent burning velocity and wall flux evaluations. *Int. J. Heat Mass Tran.* **196**, 123230 (2022)
- [8] Yao, W.-W., Zhou, X.-P.: Meshless numerical solution for nonlocal integral differentiation equation with application in peridynamics. *Eng. Anal. Bound. Elements* **144**(5), 569–582 (2022)

- [9] Chen, Y., Guo, Z.: Uniform sparse domination and quantitative weighted boundedness for singular integrals and application to the dissipative quasi-geostrophic equation. *J. Differ. Equ.* **378**(5), 871–917 (2024)
- [10] Wu, J., Yang, B., Guo, T.: Well-posedness and regularity of mean-field backward doubly stochastic volterra integral equations and applications to dynamic risk measures. *J. Math. Anal. Appl.* **535**(1), 128089 (2024)
- [11] Assari, P., Dehghan, M.: A meshless local discrete galerkin (mldg) scheme for numerically solving two-dimensional nonlinear Volterra integral equations. *Appl. Math. Comput.* **350**(1), 249–265 (2019)
- [12] Goligerdian, A., Khaksar, M., Oshagh, e: The numerical solution of a time-delay model of population growth with immigration using legendre wavelets. *Appl. Numer. Math* **197**, 243–257 (2024)
- [13] Bahrami, F., Zeinali, M., Shahmorad, S.: Recursive higher order fuzzy transform method for numerical solution of volterra integral equation with singular and nonsingular kernel. *J. Comput. Appl. Math.* **403**(15), 113854 (2022)
- [14] Najafi, E.: Smoothing transformation for numerical solution of nonlinear weakly singular volterra integral equations using quasilinearization and product integration methods. *Appl. Numer. Math.* **153**(15), 540–557 (2020)
- [15] Karapinar, E., Panda, S.K., Atangana, A.: A numerical schemes and comparisons for fixed point results with applications to the solutions of volterra integral equations in dislocated extended b-metric space. *Alexandria Eng. J.* **59**(2), 815–827 (2020)
- [16] Xiao-yong, Z.: A new strategy for the numerical solution of nonlinear volterra integral equations with vanishing delays. *Appl. Math. Comput.* **365**, 124608 (2020)
- [17] Zhang, L., Ma, F.: Pouzet-runge-kutta-chebyshev method for volterra integral equations of the second kind. *J. Comput. Appl. Math.* **288**, 323–331 (2015)
- [18] Hu, X., Wang, Z., Hu, B.: A collocation method based on roots of chebyshev polynomial for solving volterra integral equations of the second kind. *App. Math. Lett.* **29**, 108804 (2023)
- [19] Afonso, S.M., Azevedo, J.S., DA Silva, M.P.G.: Numerical analysis of the chebyshev collocation method for functional volterra integral equations. *App. Math. Lett.* **3**, 521–536 (2020)
- [20] Tao, L., Yong, H.: Extrapolation method for solving weakly singular nonlinear volterra integral equations of the second kind. *J. Math. Anal. Appl.* **324**(1), 225–237 (2006)
- [21] Meštrović, M., Ocvirk, E.: An application of romberg extrapolation on quadrature method for solving linear volterra integral equations of the second kind. *Appl. Math. Comput.* **194**(2), 389–393 (2007)
- [22] Lin, Y., Brunner, H., Zhang, S.: Higher accuracy methods for second-kind volterra integral equations based on asymptotic expansions of iterated galerkin methods. *J. Int. Eq. Appl.* **10**(4), 375–237 (1998)
- [23] Jazbi, B., Jalalvand, M., Mokhtarzadeh, M.R.: A finite difference method for the smooth solution of linear volterra integral equations. *Int. J. Nonlinear Anal. Appl.* **4**(2), 1–10 (2013)
- [24] Hung, H.S.: The numerical solution of differential and integral equations by spline functions. In: Tech. Summary Report 1053, Math. Res. Center, U.S. Army, University of Wisconsin, Madison, MR 40 6130 (1970)

- [25] El Tom, M.E.A.: Application of spline functions to volterra integral equations. *Inst. Math. Appl.* **8**, 354–357 (1971)
- [26] El Tom, M.E.A.: On the numerical stability of spline function approximations to solutions of volterra integral equations of the second kind. *BIT* **14**, 136–143 (1974)
- [27] Netravali, A.N.: Spline approximation to the solution of the volterra integral equation of the second kind. *Math. Comput.* **27**(121), 99–106 (1973)
- [28] Kauthen, J.P.: The numerical solution of integral-algebraic equations of index 1 by polynomial spline collocation methods. *Math. Comput.* **70**(236), 1503–1514 (2001)
- [29] Oja, P.: Stability of the spline collocation method for volterra integral equations. *J. Int. Equ. Appl.* **13**(2), 141–155 (2001)
- [30] Oja, P., Saveljeva, D.: Cubic spline collocation for volterra integral equations. *Computing* **69**(2), 319–337 (2002)
- [31] Oja, P., Deputat, V., Saveljeva, D.: Quadratic spline subdomain method for volterra integral equations. *Math. Model. Anal.* **10**(4), 335–344 (2005)
- [32] Kangro, R., Oja, P.: Convergence of spline collocation for volterra integral equations. *Appl. Numer. Math.* **58**, 1434–1447 (2005)
- [33] Murad, M.A., Harbi, S.H., Majee, S.N.: A solution of second kind volterra integral equation using third order non-polynomial spline function. *Baghdad Sci. J.* **12**(2), 406–4011 (2015)
- [34] Gabbasov, N.S.: Special version of the spline method for integral equations of the third kind. *Diff. Equ.* **55**(9), 1218–1225 (2019)
- [35] Maleknejad, KH., Rostami, Y.: B-spline method for solving fredholm integral equations of the first kind. *Int. J. Indust. Math.* **11**(1):ID IJIM–1057 (2019)
- [36] Liu, Z., Liu, X., Xie, J., Huang, J.: The cardinal spline methods for the numerical solution of nonlinear integral equations. *J. Chem.* 3236813 (2020)
- [37] Ramazanov, A.K., Ramazanov, A.R.K., Magomedova, V.G.: On the dynamic solution of the volterra integral equation in the form of rational spline functions. *Math. Notes* **111**(4), 595–603 (2022)
- [38] Schumaker, L.L.: *Spline Functions: Basic Theory*. Cambridge University Press, Cambridge (2007)
- [39] Stoer, J., Bulirsch, R.: *Introduction to Numerical Analysis*. Texts in Applied Mathematics, 3rd edn. Springer, New York (2002)
- [40] Sakai, M.: Some new consistency relations connecting spline values at mesh and mid points. *BIT Numer. Math.* **23**, 543–546 (1983)
- [41] Schoenberg, I.J., Whitney, A.: On polya frequency functions, iii: the positivity of translation determinants with an application to the interpolation problem by spline curves. *Trans. Am. Math. Soc.* **74**, 246–259 (1953)
- [42] Lucas, T.R.: Error bounds for interpolating cubic splines under various end conditions. *SIAM J. Numer. Anal.* **11**(3), 569–584 (1974)
- [43] Vainikko, G.: The convergence of the collocation method for non-linear differential equations. *USSR Comput. Math. Math. Phys.* **6**(1), 47–58 (1966)
- [44] Gautschi, W.: On the construction of gaussian quadrature rules from modified moments. *Math. Comp.* **24**, 245–260 (1970)
- [45] Sauter, S., Maleknejad, Kh., Torabi, P.: Numerical solution of a non-linear volterra integral equation. *Vietnam J. Math.* **44**, 5–28 (2016)

- [46] Martire, A.L., Angelis, P.D., Marchis, R.D.: A new numerical method for a class of volterra and fredholm integral equations. *J. Comput. Appl. Math.* **379**, 112944 (2020)
- [47] Rashidinia, J., Ebrahimi, N.: Collocation method for linear and nonlinear fredholm and volterra integral equations. *Appl. Math. Comput.* **270**, 156–164 (2015)

M. Ghasemi and S. Moradi  
Department of Mathematics  
University of Kurdistan  
P. O. Box 416 Sanandaj  
Iran  
e-mail: [m.ghasemi@uok.ac.ir](mailto:m.ghasemi@uok.ac.ir)

A. Goligerdian  
Department of Mathematics  
University of Houston  
3551 Cullen Blvd  
Houston TX 77204  
USA

Received: October 21, 2023.

Revised: April 29, 2024.

Accepted: May 7, 2024.