Journal of Hyperstructures 8 *(2) (2019), 156-170.*
ISSN: 2322-1666 print/2251-8436 online

NEW APPROACH FOR SOLUTION OF VOLTERRA INTEGRAL EQUATIONS USING SPLINE QUASI-INTERPOLANT

MARYAM DERAKHSHAN KHANGHAH AND MOHAMMAD ZAREBNIA

Abstract. In this paper, we present quadratic rule for approximate solution of integrals using spline quasi-interpolant. The method is applied for solving the linear Volterra integral equations. Also the convergence analysis of the method is given. The method is applied to a few examples to illustrate the accuracy and implementation of the method.

Key Words: Spline; Quasi-interpolant; Volterra; Convergence.
2010 Mathematics Subject Classification: 65R20; 41A15; 41A25.

1. Introduction

Integral and integro-differential equations are mathematical tools in many branches of science and engineering. The numerical methods for linear integral equations of the second kind studied in Delves [6]. Among these equations, Volterra integral equations arise from multiple applications, for example physics and engineering such as potential theory, Dirichlet problems, electrostatics, the particle transport problems and heat transfer problems [3, 9]. Several numerical methods have been considered to approximate the solution of Volterra integral equations such as the papers [1, 4, 5, 7, 10] that are concerned respectively with rational basis functions with product integration methods, collocation

Received:22 December 2019, Accepted: 1 August 2020. Communicated by Nasrin Eghbali
∗Address correspondence to Maryam Derakhshan; E-mail: M.Derakhshan @uma.ac.ir

methods in certain polynomial and polynomial spline spaces with uniform and graded meshes, the generalized Newton-Cotes formulae combined with product integration rules, mathematical programming methods and fractional linear multistep methods. In [17], Riley approximated the Volterra integral equations by sinc approximation methods. Reihani have solved Fredholm and Volterra integral equations by rationalized Haar functions method [16]. Simpson's quadrature method [13], Galerkin method with the Chebyshev polynomials [15] and repeated Simpson's and Trapezoidal quadrature rule [2] are other works on developing and analyzing numerical methods for solving Volterra integral equations.

Moreover, one can refer to other methods such as [14, 19, 20]. The linear Volterra integral equation is considered as

$$u - Ku = g, \tag{1.1}$$

where linear integral operator K is defined as

$$(Ku)(x) = \int_a^x k(x,t)u(t)dt, \tag{1.2}$$

$g(x)$ and $k(x,t)$ are known continuous functions and $u(x)$ is the unknown function to be determined. Integration of a function is an important operation for many physical problems.
The organization of the paper is as follows. In Section 2, we describe the construction of quadrature rule based on spline quasi-interpolant. In Section 3, we give an application of the quadrature rule of Section 2 to the numerical solution of Volterra integral equations. In Section 4, the convergence and error analysis of the numerical solution are provided. At the end we give some numerical examples which confirm our theoretical results.

2. Quadrature rule based on a quadratic spline quasi-interpolant

Let $X_n := \{x_k,\ 0 \le k \le n\}$ be the uniform partition of the interval $I = [a,b]$ into n equal subintervals, i.e. $x_k := a + kh$, with $h = \frac{b-a}{n}$. We consider the space $S_2 = S_2(I, X_n)$ of quadratic splines of class C^1 on this partition. Canonical basis is formed by the $n+2$ normalized B-splines, $\{B_k,\ k \in J\}$, $J := \{1, 2, \cdots, n+2\}$. Consider the quadratic spline quasi-interpolant (dQI) of a function f defined on I and given in

[18], that is

$$\phi_2 f = \sum_{k\in J} \upsilon_k(f) B_k, \tag{2.1}$$

where

$$\upsilon_1(f) = f_1, \quad \upsilon_{n+2}(f) = f_{n+2},$$
$$\upsilon_2(f) = -1/3f_1 + 3/2f_2 - 1/6f_3 = \beta_1 f_1 + \beta_2 f_2 + \beta_3 f_3,$$

$$\upsilon_{n+1}(f) = -1/6f_n + 3/2f_{n+1} - 1/3f_{n+2} = \beta_3 f_n + \beta_2 f_{n+1} + \beta_1 f_{n+2}, \tag{2.2}$$

and for $3 \le j \le n$,

$$\upsilon_j(f) = -1/8f_{j-1} + 5/4f_j - 1/8f_{j+1} = \gamma_1 f_{j-1} + \gamma_2 f_j + \gamma_3 f_{j+1}, \tag{2.3}$$

with $f_i = f(t_i)$, $t_1 = a$, $t_{n+2} = b$, $t_i = a + (i-3/2)h,\ 2 \le i \le n+1$. The quadratic B-spline functions at knots are defined as

$$B_i(x) = \begin{cases} \frac{(x-x_{i-3})^2}{(x_{i-1}-x_{i-3})(x_{i-2}-x_{i-3})}, & x_{i-3} \le x < x_{i-2}, \\ \frac{(x_i-x)(x-x_{i-2})}{(x_i-x_{i-2})(x_{i-1}-x_{i-2})} + \frac{(x-x_{i-3})(x_{i-1}-x)}{(x_{i-1}-x_{i-3})(x_{i-1}-x_{i-2})}, & x_{i-2} \le x < x_{i-1}, \\ \frac{(x_i-x)^2}{(x_i-x_{i-2})(x_i-x_{i-1})}, & x_{i-1} \le x < x_i. \end{cases}$$

We consider the quadrature rule defined by

$$\Im_{\phi_2 f}(x) := \int_a^x Q_2 f(\hat{a}) d\hat{a}. \tag{2.4}$$

By considering $\int_a^x B_j$ for $h = 1, n = 10$ we can get

$$\int_a^x B_1(\xi)d\xi = \begin{cases} 0, & x \le 0, \\ x - x^2 + \frac{1}{3}x^3, & 0 < x \le 1, \\ \frac{1}{3}, & else, \end{cases}$$

$$\int_a^x B_2(\xi)d\xi = \begin{cases} 0, & x \le 0, \\ \frac{1}{2}(2x^2 - x^3), & 0 < x \le 1, \\ -\frac{2}{3} + 2x - x^2 + \frac{1}{6}x^3, & 1 < x \le 2, \\ \frac{2}{3}, & else, \end{cases}$$

$$\int_a^x B_{11}(\xi)d\xi = \begin{cases} 0, & x \le 8, \\ -\frac{256}{3} + 32x - 4x^2 + \frac{1}{6}x^3, & 8 < x \le 9, \\ \frac{1202}{3} - 130x + 14x^2 - \frac{1}{2}x^3, & 9 < x \le 10, \\ \frac{2}{3}, & else, \end{cases}$$

$$\int_a^x B_{12}(\xi)d\xi = \begin{cases} 0, & x \le 9, \\ -243 + 81x - 9x^2 + \frac{1}{3}x^3, & 9 < x \le 10, \\ \frac{1}{3}, & else, \end{cases}$$

and for $3 \le j \le n$,

$$\int_a^x B_j(\xi)d\xi =$$

$$\begin{cases} 0 \; for \;\; 3 - j + x \le 2, \\ -\frac{4}{3} + 2(x+3-j) - (x+3-j)^2 + \frac{1}{6}(x+3-j)^3 \\ for \;\; 2 < x+3-j \le 3, \\ \frac{73}{6} - \frac{23}{2}(x+3-j) + \frac{7}{2}(x+3-j)^2 + \frac{1}{3}(x+3-j)^3 \\ for \;\; 3 < x+3-j \le 4, \\ -\frac{119}{6} + \frac{25}{2}(x+3-j) - \frac{5}{2}(x+3-j)^2 + \frac{1}{6}(x+3-j)^3 \\ for \;\; 4 < x+3-j \le 5, \\ 1, \;\; else. \end{cases}$$

Quadrature formula $\Im_{\phi_2 f}(x)$ can be obtained as

$$\Im_{\phi_2 f}(x) = \tilde{\zeta}_1(x)f_1 + \sum_{j=1}^{3} \beta_j(\tilde{\zeta}_2(x)f_j + \tilde{\zeta}_{n+2}(x)f_{n+3-j})$$

$$+ \sum_{j=3}^{n} \tilde{\zeta}_j(x)(\gamma_1 f_{j-1} + \gamma_2 f_j + \gamma_3 f_{j+1}) + \tilde{\zeta}_{n+2}(x)f_{n+2}, \tag{2.5}$$

where

$$\tilde{\zeta}_1(x) = \int_a^x B_1(\xi)d\xi = \begin{cases} x - \frac{1}{h}x^2 + \frac{1}{3h^2}x^3, & 0 \le x \;\&\&h - x > 0, \\ 0, & else, \end{cases}$$

$$\tilde{\zeta}_2(x) = \int_a^x B_2(\xi)d\xi =$$

$$\begin{cases} 0, & x \le 0, \\ \frac{1}{2h^2}(2hx^2 - x^3), & 0 \le x \;\&\&h - x > 0, \\ 2x - \frac{1}{h}x^2 + \frac{1}{6h^2}x^3, & h - x \le 0\&\&2h - x > 0, \\ 0, & else, \end{cases}$$

$$\tilde{\zeta}_{n+1}(x) = \int_a^x B_{n+1}(\xi)d\xi =$$
$$\begin{cases} -\frac{(n-2)^2}{2}x - \frac{(n-2)}{2h}x^2 + \frac{1}{6h^2}x^3, & (n-2)h - x \le 0\&\&(n-1)h - x > 0, \\ -\frac{3n^2-4n}{2}x + \frac{(3n-2)}{2h}x^2 - \frac{1}{2h^2}x^3, & (n-1)h - x \le 0\&\&nh - x > 0, \\ 0, & else, \end{cases}$$

$$\tilde{\zeta}_{n+2}(x) = \int_a^x B_{n+2}(\xi)d\xi =$$
$$\begin{cases} (n-1)^2 x - \frac{(n-1)}{h}x^2 + \frac{1}{3h^2}x^3, & (n-1)h - x \le 0\&\&nh - x > 0, \\ 0, & else, \end{cases}$$

and for $3 \le j \le n$,

$$\tilde{\zeta}_j(x) = \int_a^x B_j(\xi)d\xi$$
$$= \begin{cases} \frac{1}{6h^2}(x+3h-jh)^3 \\ for\ x+3h-jh \ge 0\&\& - x - 2h + jh > 0, \\ -\frac{3}{2}(x+3h-jh) + \frac{3}{2h}(x+3h-jh)^2 - \frac{1}{3h^2}(x+3h-jh)^3 \\ for\ -x-2h+jh \le 0\&\& - x - h + jh > 0, \\ \frac{9}{2}(x+3h-jh) - \frac{3}{2h}(x+3h-jh)^2 + \frac{1}{6h^2}(x+3h-jh)^3 \\ for\ -x-h+jh \le 0\&\& - x + jh > 0, \\ 0,\ \ else. \end{cases}$$

This quadrature formula can be written as

$$\mathfrak{I}_{\phi_2 f}(x) = \sum_{j=1}^{3} \psi_j(x) f_j + \sum_{j=4}^{n-1} \eta_j(x) f_j + \sum_{j=n}^{n+2} \theta_j(x) f_j, \tag{2.6}$$

where

$$\psi_1(x) = \tilde{\zeta}_1(x) - \frac{1}{3}\tilde{\zeta}_2(x), \qquad \theta_n(x) = \tilde{\zeta}_{n+2}(x) - \frac{1}{3}\tilde{\zeta}_{n+1}(x),$$
$$\psi_2(x) = \frac{3}{2}\tilde{\zeta}_2(x) - \frac{1}{8}\tilde{\zeta}_3(x), \qquad \theta_{n+1}(x) = \frac{3}{2}\tilde{\zeta}_{n+1}(x) - \frac{1}{8}\tilde{\zeta}_n(x),$$
$$\psi_3(x) = -\frac{1}{6}\tilde{\zeta}_2(x) + \frac{5}{4}\tilde{\zeta}_3(x) - \frac{1}{8}\tilde{\zeta}_4(x), \quad \theta_{n+2}(x) = -\frac{1}{6}\tilde{\zeta}_{n+1}(x) + \frac{5}{4}\tilde{\zeta}_n(x) - \frac{1}{8}\tilde{\zeta}_{n-1}(x),$$

(2.7)

$$\eta_j(x) = -\frac{1}{8}\tilde{\zeta}_{j+1}(x) + \frac{5}{4}\tilde{\zeta}_j(x) - \frac{1}{8}\tilde{\zeta}_{j-1}(x), \qquad 4 \le j \le n-1.$$

Theorem 2.1. *For any partition X of I, the infinity norm of Q is uniformly bounded by 3. If the partition is uniform, one has* $\| \phi \|_\infty = \frac{305}{207} \approx$ 1.4734.

Proof. For proof, refer to [18]. □

According [18], there exists a constant C such that

$$\| f - \phi_2 f \|_\infty \leq Ch^3 \| D^3 f \|_\infty .$$

3. Application to Volterra integral equations

In this section, we illustrate an application of the quadrature rule to numerical solution of Volterra integral equation

$$u(x) - \int_a^x k(x,t)u(t)dt = g(x),$$

where $k(.,.) \in C([a,b] \times [a,b])$ and $g(x) \in C([a,b])$ are known functions and $u(x)$ is the unknown function to be determined. We use spline quasi-interpolant method for Volterra integral equation. The method associated with the quadrature formula

$$(K_n u)(x) = \int_a^x \phi_2(k(x,.)u(.))(\hat{a})d\hat{a} = \Im_{\phi_2 ku}(x), \tag{3.1}$$

we obtain

$$u(x) - (K_n u)(x) = g(x),$$

consists in looking for a solution u satisfying

$$u(x) - (\sum_{j=1}^{3} \psi_j(x)k(x,t_j)u_j + \sum_{j=4}^{n-1} \eta_j(x)k(x,t_j)u_j + \sum_{j=n}^{n+2} \theta_i(x)k(x,t_j)u_j) = g(x). \tag{3.2}$$

In summary, we can write

$$u(x) - \sum_{j=1}^{n+2} F_j(x)k(x,t_j)u_j = g(x),$$

where

$$F_j = \begin{cases} \psi_j(x), & 1 \leq j \leq 3 \\ \eta_j(x), & 4 \leq j \leq n-1, \\ \theta_j(x), & n \leq j \leq n+2. \end{cases}$$

By replacing x_i, we have

$$u(x_i) - \sum_{j=1}^{n+2} F_j(x_i) k(x_i, t_j) u_j = g(x_i). \tag{3.3}$$

Equation (3.3) can be simplified in the matrix form

$$(I - K^*)U = G, \tag{3.4}$$

where

$$\begin{aligned} &U = [u_1, u_2, \ldots, u_{n+2}]^T, \\ &K^* = [F_j(x_i) k(x_i, t_j)]_{i,j}, i, j = 1, \ldots, n+2, \\ &G = [g(x_1), g(x_2), \ldots, g(x_{n+2})]^T. \end{aligned}$$

Having used the solution $u_j,\ j = 1, \ldots, n+2$, in the system (3.4), we employ a method similar to the Nystrom's idea for the Volterra integral equation, i.e. we used

$$u_n(x) = \sum_{j=1}^{n+2} F_j(x) k(x, t_j) u(t_j) + g(x). \tag{3.5}$$

4. Convergence Analysis

In this section, we shall provide the convergence analysis of the proposed method. For this purpose, we consider the following theorem.

Theorem 4.1. *Let $\tilde{r}_n$ error term for the spline quasi interpolant method. Furthermore, let $M_0 = \max |F_j(x_i)||k(x_i, t_j)|$ and $\chi_j = \max |k(x_i, t_j)|$. Then*

$$|\epsilon_{n,i}| \leq \frac{O(h^4)}{1 - M_0} exp(\sum_{j=1}^{n+1} \frac{\chi_j}{1 - M_0}).$$

Proof. In fact

$$u_n(x_i) = \sum_{j=1}^{n+2} F_j(x_i) k(x_i, t_j) u(t_j) + g(x_i).$$

Thus

$$u_n(x_i) - u(x_i) = \sum_{j=1}^{n+2} F_j(x_i)k(x_i,t_j)(u_n(t_j) - u_j) + \sum_{j=1}^{n+2} F_j(x_i)k(x_i,t_j)u(x_i) - \int_a^{x_i} k(x_i,s)u(s)ds.$$

Let

$$\epsilon_{n,i} = u_n(x_i) - u(x_i),$$

and

$$\tilde{r}_n(x_i) = \sum_{j=1}^{n+2} F_j(x_i)k(x_i,t_j)u(x_i) - \int_a^{x_i} k(x_i,s)u(s)ds.$$

Hence

$$\epsilon_{n,i} = \sum_{j=1}^{n+2} F_j(x_i)k(x_i,t_j)\epsilon_{n,j} + \tilde{r}_n(x_i).$$

Thus

$$|\epsilon_{n,i}| \leq \sum_{j=1}^{n+2} |F_j(x_i)||k(x_i,t_j)||\epsilon_{n,j}| + |\tilde{r}_n(x_i)|,$$

so that

$$|\epsilon_{n,i}| \leq \sum_{j=1}^{n+1} |F_j(x_i)||k(x_i,t_j)||\epsilon_{n,j}| + |F_n(x_i)k(x_i,t_n)||\epsilon_{n,n}| + |\tilde{r}_n(x_i)|.$$

Now using the Gronwall Lemma [8], we obtain

$$|\epsilon_{n,i}| \leq \frac{O(h^4)}{1-M_0} exp(\sum_{j=1}^{n+1} \frac{\chi_j}{1-M_0}).$$

□

Theorem 4.2. *Let $k \in C([a,b]\times[a,b])$ and $u \in C[a,b]$. Then we have*

$$\| u_n - u \|_\infty \leq \| (I-K_n)^{-1} \| \| (K_n - K)u \|_\infty .$$

Proof. For each $x \in [a,b]$, Let $\kappa(.,.)$ set by

$$\kappa(x,t) = \begin{cases} k(x,t), & a \leq t \leq x, \\ 0, & x < t \leq b. \end{cases}$$

The method associated with the quadrature formula

$$(K_n u)(x) = \int_a^b \phi_2(k(x,.)u(.))(t)dt = \sum_{j=1}^{n+2} \digamma_j(x)k(x,t_j)u(t_j),\ a \leq x \leq b.$$

There exists a constant M such that

$$\sum_{j=1}^{n+2} |\digamma_j(x)| \leq M,\ a \leq x \leq b.$$

We get

$$\parallel K_n u \parallel_\infty \leq M max|\kappa(x,t)| \parallel u \parallel_\infty .$$

Further, for all $x_1, x_2 \in [a,b]$, we have

$$\parallel K_n u(x_1) - K_n u(x_2) \parallel_\infty \leq M max|\kappa(x_1,t) - \kappa(x_2,t)| \parallel u \parallel_\infty .$$

On the other

$$I - K_n = I - K + K - K_n.$$

Then

$$I - K_n = (I-K)[I + (I-K)^{-1}(K-K_n)].$$

From equation (3.4) we obtain

$$u_n - u = (I-K_n)^{-1}g - (I-K)^{-1}g.$$

Thus

$$\begin{aligned} u_n - u &= (I-K_n)^{-1}(K_n - K)(I-K)^{-1}g \\ &= (I-K)^{-1}(I + (I-K)^{-1}(K-K_n))^{-1}g - (I-K)^{-1}g \\ &= ((I + (I-K)^{-1}(K-K_n))^{-1} - I)(I-K)^{-1}g \\ &= (((I-K_n)(I-K)^{-1})^{-1} - I)(I-K)^{-1}g \\ &= (I-K_n)^{-1}(K_n - K)(I-K)^{-1}g. \end{aligned}$$

Hence

$$u_n - u = (I-K_n)^{-1}(K_n - K)u,$$

and we deduce

$$\parallel u_n - u \parallel_\infty \leq \parallel (I-K_n)^{-1} \parallel \parallel (K_n - K)u \parallel_\infty .$$

□

5. Numerical example

In this section, in order to illustrate the performance of the presented method in solving Volterra integral equations and justify the accuracy and efficiency of the method, we consider the following examples.

Example 5.1. Consider the following Volterra integral equation

$$u(x) - \int_0^x (x+t)u(t)dt = \frac{7}{12}x^4 + \frac{5}{3}x^3 + \frac{5}{2}x^2 + 2x + 1,$$

where the exact solution is $u(x) = (1+x)^2$. In Table 1, numerical results are presented for rules $\Im_{\phi_2 f}$. We obtain an approximation to the solution of (3.4). Numerical results illustrate accuracy of the proposed quadrature rule. By increasing the value of n, the errors have been decreased. In Table 2, we present the absolute errors of algorithm combines Trapezoidal and Simpson rules [12] for different values of n. A numerical comparison between Tables 1, 2 shows that spline quasi-interpolant method is more accurate than of algorithm combines Trapezoidal and Simpson rules [12].

Table 1. Max. Abs. Err. for Example 5.1 ($\Im_{Q_2 f}$)

n	$\parallel u - u_n \parallel_\infty$	*order*
19	7.43928×10^{-7}	$-$
23	3.48711×10^{-7}	3.96587
35	6.55478×10^{-8}	3.98106
45	2.40445×10^{-8}	3.99051
55	1.0788×10^{-8}	3.99397
65	5.534×10^{-9}	3.99585
75	3.12346×10^{-9}	3.99697
85	1.89379×10^{-9}	3.99767
95	1.21395×10^{-9}	3.99818

Example 5.2. Consider the following Volterra integral equation

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

where the exact solution is $u(x) = x$. In Table 3, numerical results are presented for rule $\Im_{\phi_2 f}$. In table 4, we compare the absolute errors of the

TABLE 2. Max. Abs. Err. [12]

n	$\| u - u_n \|_\infty$
19	8.36×10^{-5}
23	4.77×10^{-5}
35	1.33×10^{-5}
45	6.29×10^{-6}
55	3.44×10^{-6}
65	2.08×10^{-6}
75	1.36×10^{-6}
85	9.35×10^{-7}
95	6.70×10^{-7}

spline quasi-interpolant method for $n = 32$ with numerical expansion-iterative method [11]. The results show the efficiency and rate of convergence of the method. Figure 1 shows the maximum absolute errors for the proposed method.

TABLE 3. Max. Abs. Err. for Example 5.2 ($\Im_{Q_2 f}$)

n	$\| u - u_n \|_\infty$
20	3.88179×10^{-7}
40	2.42884×10^{-8}
60	4.79842×10^{-9}
80	1.5183×10^{-9}
100	6.21901×10^{-10}

Example 5.3. Consider the following Volterra integral equation

$$u(x) + \int_0^x xtu(t)dt = e^{-x^2} + \frac{x(1-e^{-x^2})}{2},$$

where the exact solution is $u(x) = e^{-x^2}$. In Table 5, numerical results are presented for rule $\Im_{\phi_2 f}$. In table 6, we compare the absolute errors of the spline quasi-interpolant method for $n = 32$ with numerical expansion-iterative method [11]. Figure 2 shows the maximum absolute errors for the proposed method.

TABLE 4. Numerical results for Example 5.2

x	*Absolute error*	*Absolute error*[11]
0	0	1.5625×10^{-2}
0.1	8.55709×10^{-9}	9.375×10^{-3}
0.2	1.86236×10^{-8}	3.125×10^{-3}
0.3	2.78283×10^{-8}	3.125×10^{-3}
0.4	3.36708×10^{-8}	9.375×10^{-3}
0.5	3.80447×10^{-8}	1.5625×10^{-2}
0.6	4.72105×10^{-8}	9.375×10^{-3}
0.7	5.65627×10^{-8}	3.68045×10^{-3}
0.8	5.83707×10^{-8}	3.125×10^{-3}
0.9	4.9622×10^{-8}	9.375×10^{-3}

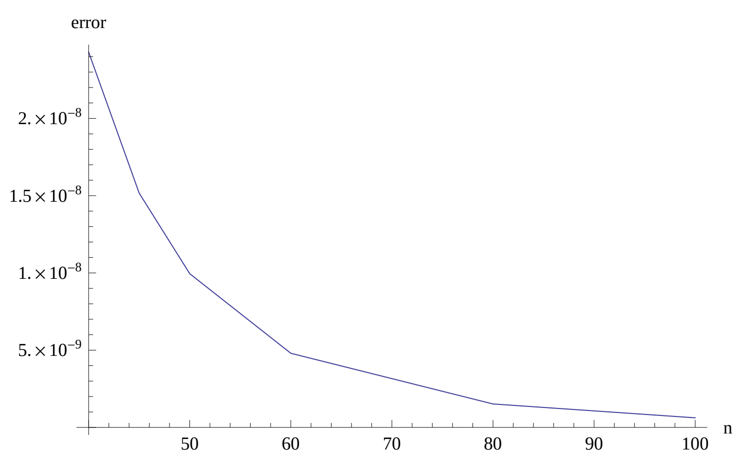

FIGURE 1. The absolute error $\| u - u_n \|_\infty$ for different values of n for Example 5.2.

Example 5.4. Consider the following Volterra integral equation

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

where the exact solution is $u(x) = \frac{1}{3}(2\cos\sqrt{3}t + 1)$. In Table 7, numerical results are presented for rule $\mathfrak{I}_{\phi_2 f}$. In table 8, we compare the absolute

TABLE 5. Max. Abs. Err. for Example 5.3 ($\Im_{Q_2 f}$)

n	$\parallel u - u_n \parallel_\infty$
20	4.78517×10^{-7}
40	3.07465×10^{-8}
60	6.10838×10^{-9}
80	1.93431×10^{-9}
100	7.92292×10^{-10}

TABLE 6. Numerical results for Example 5.3

x	*Absolute error*	*Absolute error*[11]
0	0	3.25×10^{-4}
0.1	7.99755×10^{-9}	2.02×10^{-3}
0.2	1.41309×10^{-8}	1.28×10^{-3}
0.3	1.37384×10^{-8}	1.647×10^{-3}
0.4	5.21359×10^{-9}	6.295×10^{-3}
0.5	8.52356×10^{-9}	1.2284×10^{-2}
0.6	2.59421×10^{-8}	7.87×10^{-3}
0.7	4.61481×10^{-8}	2.669×10^{-3}
0.8	6.24702×10^{-8}	2.661×10^{-3}
0.9	6.84528×10^{-8}	7.562×10^{-3}

errors of the spline quasi-interpolant method for $n = 64$ with Rationalized Haar functions method [16]. Figure 3 shows the maximum absolute errors for the proposed method.

TABLE 7. Max. Abs. Err. for Example 5.4 ($\Im_{Q_2 f}$)

n	$\parallel u - u_n \parallel_\infty$
20	8.97969×10^{-7}
40	5.75157×10^{-8}
60	1.14124×10^{-8}
80	3.61664×10^{-9}
100	1.48246×10^{-9}

TABLE 8. Numerical results for Example 5.4

x	*Absolute error*	*Absolute error*[16]
0	0	0.6×10^{-5}
0.1	8.57026×10^{-9}	4.23×10^{-4}
0.2	7.42193×10^{-9}	3.45×10^{-4}
0.3	6.67605×10^{-9}	4.55×10^{-4}
0.4	6.28758×10^{-9}	2.7×10^{-5}
0.5	4.13835×10^{-9}	0.7×10^{-4}
0.6	3.59153×10^{-9}	3.7×10^{-4}
0.7	1.79297×10^{-9}	1.52×10^{-4}
0.8	2.97626×10^{-10}	6.7×10^{-5}
0.9	1.00404×10^{-9}	2.4×10^{-4}

6. Conclusion

In this article, we illustrated a rule based on spline quasi-interpolant. In following, we employed this rule to the solution of a Volterra integral equation. The numerical examples were presented to illustrate the accuracy and the implementation of the method.

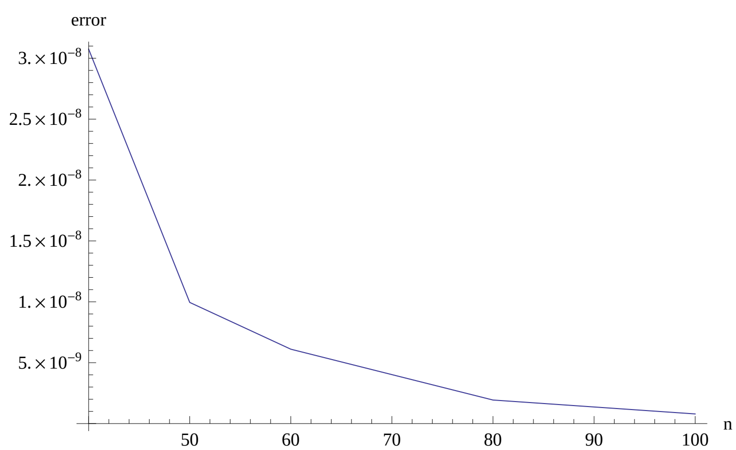

FIGURE 2. The absolute error $\| u - u_n \|_\infty$ for different values of n for Example 5.3.

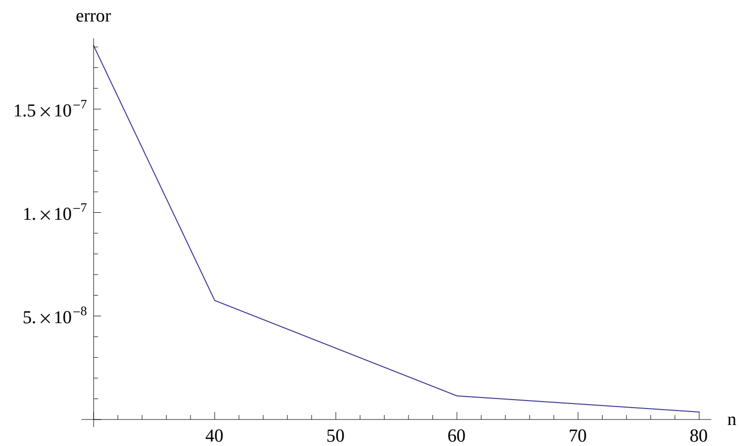

FIGURE 3. The absolute error $\| u - u_n \|_\infty$ for different values of n for Example 5.4.

REFERENCES

[1] S. Abelman, D. Eyre, *A rational basis for second kind Abel integral equation, J. Comput. Appl. Math,* **34** *(1991), 281–290.*

[2] M. Aigo, On the numerical approximation of Volterra integral equations of second kind using quadrature rules, International Journal of Advanced Scientific and Technical Research, **1** *(2013).*

[3] K. Atkinson, The numerical solution of integral equations of the second kind, The press Syndicate of the University of Cambridge, United Kingdom, (1997).

[4] H. Brunner, The numerical solution of weakly singular Volterra integral equations by collocation on graded meshes, Math. Comput, **45** *(1985), 417–437.*

*[5] R.F. Cameron, S. Mckee, Product integration methods for second kind Abel integral equations, J. Comput. Appl. Math,***11** *(1984), 1–10.*

[6] L.M. Delves, J.L. Mohamed, Computational methods for integral equations, Cambridge University Press, (1985).

[7] E.A. Galperin, E.J. Kansa, A. makroglou, S. A. Nelson, Mathematical programming methods in the numerical solution of Volterra integral and integro-differential equations with weakly singular kernel, Proceedings of the Second World Congress of Nonlinear Analysts. Athens, Greece, Nonlinear Anal. Theory, Mehods Appl, **30** *(1997), 1505–1513.*

[8] W. Hackbusch, Integral equations theory and numerical treatment, Birkhauser, (1995).

[9] A.J. Jerri, Introduction to integral equations with applications, John Wiley and Sons, INC, (1999).

[10] Ch. Lubich, Fractional linear multistep methods for Abel-Volterra integral equations of the second kind, Math. Comput, **45** *(1985), 463–469.*

[11] Z. Masouri, *Numerical expansion-iterative method for solving second kind Volterra and Fredholm integral equations using block-pulse functions, Advanced Computational Techniques in Electromagnetics, (2012).*

[12] A. Mennouni, *A new numerical approximation for Volterra integral equations combining two quadrature rules, Applied Mathematics and Computation.* **218** *(2011), 1962–1969.*

[13] F. Mirzaee, *A computational method for solving linear Volterra integral equations, Appl. Math. Sci,* **6** *(2012), 807–814.*

[14] M. Mustafa, *Numerical solution of volterra integral equations with Delay using Block methods, AL-Fatih journal,* **36** *(2008), 42–51.*

[15] M.M. Rahman, M. A. Hakim, M. Kamrul, *Numerical solutions of Volterra integral equations of second kind with the help of Chebyshev polynomials, Annals of Pure and Applied Mathematics,* **1** *(2012), 158–167.*

[16] M.H. Reihani, Z. Abadi, *Rationalized Haar functions method for solving Fredholm and Volterra integral equations, J. Comput. Appl. Math,* **200** *(2007), 12–20.*

[17] B.V. Riley, *the numerical solution of Volterra integral equations with nonsmooth solutions based on sinc approximation, Appl. Numer. Math,* **9** *(1992), 249–247.*

[18] P. Sablonniere, *A quadrature formula associated with a univariate quadratic spline quasi-interpolant, BIT Numerical Mathematics,* **47** *(2007), 825–837.*

[19] P.W. Sharp, J.H. Verner, *Extended explicit Bel'Tyukov pairs of orders 4 and 5 for Volterra integral equations of the second kind, Appl. Numer. Math,* **34** *(2000), 261–274.*

[20] T. Tang, X. Xu, J. Cheng, *On spectral methods for Volterra type integral equations and the convergence analysis, J. Comp. Math,* **26** *(2008), 825–837.*

Maryam Derakhshan Khanghah
Department of Mathematics, University of Mohaghegh Ardabili, P.O.Box 56199-11367, Ardabil, Iran
Email: `M.Derakhshan@uma.ac.ir`

Mohammad Zarebnia
Department of Mathematics, University of Mohaghegh Ardabili, P.O.Box 56199-11367, Ardabil, Iran
Email: `Zarebnia@uma.ac.ir`