

Application of Splines of the second order approximation to Volterra integral equations of the second kind. Applications in Systems Theory and Dynamical Systems.

I. G. Burova, G. O. Alcybeev
St. Petersburg State University
7/9 Universitetskaya nab., St.Petersburg, 199034
Russia
i.g.burova@spbu.ru, burovaig@mail.ru

Received: June 30, 2020. Revised: January 8, 2021. Accepted: February 1, 2021. Published: February 9, 2021.

Abstract—This paper discusses the application of local interpolation splines of the second order of approximation for the numerical solution of Volterra integral equations of the second kind. Computational schemes based on the use of polynomial and non-polynomial splines are constructed. The advantages of the proposed method include the ability to calculate the integrals which are present in the computational methods. The application of splines to the solution of nonlinear Volterra integral equations is also discussed. The results of numerical experiments are presented

Keywords—Systems Theory, Dynamical Systems, non-polynomial splines, polynomial splines, Volterra integral equations of the second kind

I. INTRODUCTION

VARIOUS numerical methods are known for solving Volterra integral equations of the second kind. The most common numerical methods are based on the use of quadrature formulas [1]-[3]. New methods for calculating integrals appear periodically. For example, in paper [4] the application of the finite-difference methods are investigated to compute the definite integrals. When solving a variety of tasks, splines are successfully used. Paper [5] deals with the use of the first two vanishing moments for constructing cubic spline-wavelets orthogonal to polynomials of the first degree. The method proposed in [6] can be used to calculate the real eigenvalues of an arbitrary matrix with real elements. This method uses splines of the Lagrangian type of the fifth order and/or polynomial integro-differential splines of the fifth order. In paper [7] periodic cubic splines were used to obtain smooth walking trajectories of every joint in the biped model. Recently, a number of studies have appeared on the use of splines for solving integral equations. Among them, special mention should be made of works using nonpolynomial splines. The most commonly used are polynomial splines.

Non-polynomial splines are used much less often but they can provide a solution with smaller error. Among the studies using splines to solve the problem of solving Volterra integral equation, we note the following. In paper [8] spline functions were used to propose a new scheme for solving the linear Volterra–Fredholm integral equations of the second kind. In paper [10] two types of non-polynomial spline functions (linear, and quadratic) were used to find the approximate solution of Volterra integro–differential equations. In [11], a computational method for solving nonlinear Volterra–Fredholm Hammerstein integral equations is proposed by using compactly supported semiorthogonal cubic B-spline wavelets as basis functions. In [12], the non-polynomial spline basis and Quasi-linearization method to solve the nonlinear Volterra integral equation were used. In [13], a new collocation technique for the numerical solution of the Fredholm, Volterra and mixed Volterra–Fredholm integral equations of the second kind is introduced, and a numerical integration formula on the basis of the linear Legendre multi-wavelets is also developed. In paper [9] the advanced multistep hybrid methods were used for the construction of the numerical methods for solving the Volterra integral equations.

This paper discusses the application of the local interpolation of both polynomial splines and non-polynomial splines (see [14]-[17]) of the second and the third order of approximation for the numerical solution of the Volterra integral equations of the second kind. These splines have the following properties:

- these splines are continuous, piecewise defined functions;
- the support of the basic spline occupies several neighboring grid intervals (the support of the piecewise-linear basic spline occupies two adjacent grid intervals, the support of the piecewise-quadratic basic spline occupies three adjacent grid intervals);
- the approximation is constructed separately on each grid interval in the form of the sum of the products of the values of the function at the grid nodes and the basis splines.

For the first time, approximations by such quadratic splines were used by Professor Ryabenkiy V.S. Note, that the polynomial Courant functions are well known and have been used for a long time. The method for obtaining an estimate of the error of approximation of a function by non-polynomial piecewise-given functions is discussed in detail in paper [17]. Let n be integer, and a, b be real. Suppose that a grid of nodes $\{x_j\}$ with step $h = \frac{b-a}{n}$, is constructed on the interval $[a, b]$. Thus, $x_j = a + jh, j = 0, \dots, n$. Let us consider the numerical solution of the Volterra equation of the second kind. The linear Volterra equation of the second kind has the form:

$$u(x) + \int_a^x K(x, s)u(s)ds = f(x), \quad x \in [a, b], \quad (1)$$

where f is a given function, K, f are continuous functions and $u(x)$ is an unknown function.

First, we recall the method for solving integral equations based on the composite trapezoidal rule.

As it is known, the composite trapezoidal rule for calculating an integral has the following form (see [3]):

$$\int_a^b u(s)ds \approx h \sum_{j=1}^{n-1} u(x_j) + (u(x_0) + u(x_n))/2.$$

If the function u is as such that $u \in C^2[a, b]$, then the remainder can be written in the form:

$$\int_a^b u(s)ds - h \sum_{j=1}^{n-1} u(x_j) - \frac{u(x_0) + u(x_n)}{2} = -\frac{(b-a)h^2}{12} u''(\xi), \quad a \leq \xi \leq b.$$

Consistently applying the composite trapezoidal rule for calculating the integral in equation (1), taking into account the error in calculating the integral and putting $x = x_j$, we obtain the system of equations:

$$u(x_0) = f(x_0),$$

$$\tilde{u}(x_1) = hK_{10}u(x_0)/2 + hK_{11}\tilde{u}(x_1)/2 + f(x_1),$$

$$\tilde{u}(x_k) = \frac{hK_{k0}u(x_0)}{2} + h \sum_{j=1}^{k-1} K_{kj}\tilde{u}(x_j) + \frac{hK_{kk}\tilde{u}(x_k)}{2} + f(x_k), \quad k = 2, \dots, n.$$

Here $K_{kj} = K(x_k, s_j), \tilde{u}(x_k) \approx u(x_k)$.

The calculations can be carried out according to the following scheme:

$$\tilde{u}(x_k) = \frac{1}{1 - \frac{hK_{kk}}{2}} \left(\frac{hK_{k0}u(x_0)}{2} + f(x_k) + h \sum_{j=1}^{k-1} K_{kj}\tilde{u}(x_j) \right).$$

As a result, we obtain approximate values $\tilde{u}(x_k)$ of the solution u of integral equation (1) at the grid points.

II. APPLICATION OF SPLINES OF THE SECOND ORDER OF APPROXIMATION

Let us consider the numerical solution of the Volterra equation of the second kind

$$u(x) + \int_a^x K(x, s)u(s)ds = f(x), \quad x, s \in [a, b]$$

using splines of the second order of approximation. Suppose $K(x, \cdot)u(\cdot) \in C^2[a, b]$. Suppose that $\{x_j\}$ is a grid of nodes with step h on the interval $[a, b]$.

The function $u(s)$ can be approximated by the polynomial spline (see [17])

$$U(s) = u(x_j)\omega_j(s) + u(x_{j+1})\omega_{j+1}(s), \quad (2)$$

$$s \in [x_j, x_{j+1}],$$

where the basis splines have the form:

$$\omega_j(s) = \frac{s - x_{j+1}}{x_j - x_{j+1}}, \quad s \in [x_j, x_{j+1}],$$

$$\omega_{j+1}(s) = \frac{s - x_j}{x_{j+1} - x_j}, \quad s \in [x_j, x_{j+1}].$$

Let us denote

$$\|u''\|_{[x_j, x_{j+1}]} = \max_{[x_j, x_{j+1}]} |u''(x)|,$$

Theorem 1. Let $u \in C^2[a, b]$. To approximate the function $u(x), x \in [x_j, x_{j+1}], u \in C^2[a, b]$, by spline (2), the following inequality is valid:

$$|u(x) - U(x)| \leq K_1 h^2 \|u''\|_{[x_j, x_{j+1}]}, K_1 = 1/8.$$

Proof. It is easy to notice that U is an interpolation polynomial of the first degree, and x_j, x_{j+1} are the interpolation nodes, $u(x_j) = U(x_j), u(x_{j+1}) = U(x_{j+1})$. Using the remainder term we get $u(x) - U(x) = \frac{u''(\tau)}{2!}(x - x_j)(x - x_{j+1}), \tau \in [x_j, x_{j+1}]$.

When $x \in [x_j, x_{j+1}]$ we can use the relation $x = x_j + th, t \in [0, 1]$. Therefore $(x - x_j)(x - x_{j+1}) = h^2 t(t - 1)$.

We can see that $\max_{t \in [0, 1]} |t(t - 1)| = 1/4$.

Thus, $\max_{x \in [x_j, x_{j+1}]} |u(x) - U(x)| = \max_{[x_j, x_{j+1}]} |u''| \frac{h^2}{8}$.

The proof is complete.

The function $u(x), u \in C^2[a, b]$ can also be approximated with non-polynomial splines. As it is shown in paper [17], if the functions φ_1, φ_2 form a Chebyshev system, then the basis functions $\omega_k, k = j, j + 1$, can be determined by solving the system of equations

$$\varphi_1(x_j)\omega_j(x) + \varphi_1(x_{j+1})\omega_{j+1}(x) = \varphi_1(x),$$

$$\varphi_2(x_j)\omega_j(x) + \varphi_2(x_{j+1})\omega_{j+1}(x) = \varphi_2(x),$$

$$x \in [x_j, x_{j+1}].$$

Suppose that the determinant of the system does not equal zero. Let us investigate the case when $\varphi_1(x) = 1, \varphi_2(x) = \varphi(x)$.

We construct a non-polynomial approximation of function $u(x)$, on each grid interval $[x_j, x_{j+1}]$ in the form:

$$U(x) = u(x_j)\omega_j(x) + u(x_{j+1})\omega_{j+1}(x),$$

$$x \in [x_j, x_{j+1}],$$

where

$$\omega_j(x) = \frac{\varphi(x) - \varphi(x_{j+1})}{\varphi(x_j) - \varphi(x_{j+1})},$$

$$\omega_{j+1}(x) = \frac{\varphi(x) - \varphi(x_j)}{\varphi(x_{j+1}) - \varphi(x_j)}$$

Note that this formula for function interpolation can also be applied on a uniform grid of nodes.

Depending on the choice of the function $\varphi(x)$, we obtain slightly different estimates of the error (see paper [17]), but they all have an approximation error of the order of $O(h^2)$. This approximation error can be obtained using Taylor's theorem. But we will apply the method described in paper [17].

Table 1 shows the actual errors of approximation of some functions $u(x)$ obtained with the use of the polynomial and non-polynomial splines when $h = 0.1$, $[a, b] = [-1, 1]$.

Table 1. The actual errors of approximation of some functions obtained with the use of the polynomial and non-polynomial splines

$u(x)$	$\varphi(x) = x$	$\varphi(x) = \exp(x)$	$\varphi(x) = \exp(-x)$
$\exp(x)$	0.00323	0	0.00646
$\sin(x)$	0.00102	0.00177	0.00177
x^2	0.00250	0.00487	0.00487
$\exp(-x)$	0.00323	0.00646	0

Table 2 shows the theoretical errors of approximation of some functions obtained with the use of the polynomial splines when $h = 0.1$, $[a, b] = [-1, 1]$.

Table 2. The theoretical errors of approximation of some functions with the polynomial splines

$u(x)$	$\max_{[-1,1]} u - U $
$\exp(x)$	0.00340
$\sin(x)$	0.00105
x^2	0.00250

Let ordered distinct nodes $\{x_j\}$ be such that $x_{j+1} - x_j = h$.

Theorem 2. Let function $u(x)$ be such that $u \in C^2([a, b])$. Suppose the basis splines $\omega_j(x), \omega_{j+1}(x)$ are constructed when $U(x) = u(x), u(x) = 1, \varphi(x)$, for $x \in [x_i, x_{i+1}]$.

Then, for $\varphi(x) = \exp(x), \exp(-x), x \in [x_i, x_{i+1}]$ we have $|u(x) - U(x)| \leq K_2 h^2 \|Lu\|, x \in [x_j, x_{j+1}], K_2 > 0$.

Here $Lu = \exp(-x) (u''(x) + u'(x))$ in the case of $\varphi_1(x) = 1, \varphi_2(x) = \exp(-x)$,
 $Lu = \exp(x) (u''(x) - u'(x))$ in the case of $\varphi_1(x) = 1, \varphi_2(x) = \exp(x)$.

Proof. In the case of the non-polynomial splines as it was shown in paper [17] we construct a homogeneous equation, which has a fundamental system of solutions $1, \varphi(x)$

$$Lu = \begin{vmatrix} 1 & \varphi(x) & u(x) \\ 0 & \varphi'(x) & u'(x) \\ 0 & \varphi''(x) & u''(x) \end{vmatrix} = 0.$$

It is easy to see, that the Wronskian $W(x) = \begin{vmatrix} 1 & \varphi(x) \\ 0 & \varphi'(x) \end{vmatrix}$ does not equal zero when $\varphi(x) = \exp(x)$, or $\exp(-x)$. Now we can construct a general solution of the nonhomogeneous equation $Lu = F$ by the method of variation of the constants.

Expanding the determinant according to the elements of the last column and dividing all terms of the equation by $W(x)$ we obtain the equation $Lu = 0$ in the form. $u'' + qu' + pu = 0$. Here q and p are some coefficients.

After we have constructed a general solution of nonhomogeneous equation $Lu = F$ by the method of variation of the constants we obtain the function $u(x), x \in [x_j, x_{j+1}]$.

Let us write down the results for several special cases.

In the case of $\varphi_1(x) = 1, \varphi_2(x) = \exp(x)$ we obtain $Lu = \exp(x) (u''(x) - u'(x))$ and

$$u(x) = \int_{x_j}^x \exp(t) (u''(t) - u'(t)) dt + c_1 + c_2 \exp(x).$$

In the case of $\varphi_1(x) = 1, \varphi_2(x) = \exp(-x)$ we obtain $Lu = -\exp(-x) (u''(x) + u'(x))$ and

$$u(x) = - \int_{x_j}^x (u''(t) + u'(t)) \exp(-t) dt + c_1 + c_2 \exp(-x).$$

Here $c_i, i = 1, 2$, are some arbitrary constants, $x \in [x_i, x_{i+1}]$.

We construct the approximation of $u(x)$ in the form:

$$U(x) = u(x_j)\omega_j(x) + u(x_{j+1})\omega_{j+1}(x), x \in [x_j, x_{j+1}].$$

Thus, using the results from paper [17], we get

$$|u(x) - U(x)| \leq K_2 h^2 \|Lu\|, x \in [x_j, x_{j+1}],$$

in the case of $\varphi_1(x) = 1, \varphi_2(x) = \varphi(x)$.

The proof is complete.

Now we will consider, in detail, the construction of computational schemes for solving the Volterra equation of the second kind. In the construction of the computational schemes, we will use the approximation formulas with polynomial and non-polynomial splines of the second order of approximation.

Transforming the integral $\int_{x_j}^{x_{j+1}} K(x, s)u(s)ds$ using formula

(2) or (1), we obtain

$$\int_{x_j}^{x_{j+1}} K(x, s)u(s)ds = u(x_j) \int_{x_j}^{x_{j+1}} K(x, s)\omega_j(s)ds + u(x_{j+1}) \int_{x_j}^{x_{j+1}} K(x, s)\omega_{j+1}(s)ds + O(h^3).$$

To construct a numerical method, we discard the error and denote $\tilde{u}(x_j) \approx u(x_j)$. Let us introduce the notation

$$W_j(x) = \tilde{u}(x_j) \int_{x_j}^{x_{j+1}} K(x, s)\omega_j(s)ds + \tilde{u}(x_{j+1}) \int_{x_j}^{x_{j+1}} K(x, s)\omega_{j+1}(s)ds.$$

Setting $x = x_k$, we obtain the numerical method

$$\tilde{u}(x_k) + \sum_{s=0}^{k-1} W_s(x_k) = f(x_k), k = 0, \dots, n.$$

In a more detailed notation, the system of equations has the form

$$\tilde{u}(x_0) = f(x_0), \tilde{u}(x_1) + \tilde{u}(x_0) \int_{x_0}^{x_1} K(x_1, s)\omega_0(s)ds +$$

$$\begin{aligned}
 & +\tilde{u}(x_1) \int_{x_0}^{x_1} K(x_1, s)\omega_1(s)ds = f(x_1), \\
 & \dots\dots\dots \\
 & \tilde{u}(x_n) + \sum_{s=0}^{n-2} W_s(x_n) \\
 & +\tilde{u}(x_{n-1}) \int_{x_{n-1}}^{x_n} K(x_n, s)\omega_{n-1}(s)ds \\
 & +\tilde{u}(x_n) \int_{x_{n-1}}^{x_n} K(x_n, s)\omega_n(s)ds = f(x_n).
 \end{aligned}$$

We find the approximate values of the solution of the Volterra integral equation at the grid nodes by solving each equation sequentially. It is assumed that the integrals can be calculated exactly, or a quadrature formula with an error not less than $O(h^3)$ can be applied.

III. NUMERICAL RESULTS

Now we apply the polynomial, the non-polynomial splines of the second order of approximation, and the composite trapezoidal rule for solving some Volterra integral equations. We will carry out the calculations in the package MAPLE with *Digits* = 15 and the number of nodes $n = 32, 64, 128$.

Problem 1. We take the equation

$$u(x) = g(x) - \int_0^x (x-t)\cos(x-t)u(t)dt, \quad x \in [0, 1].$$

The exact solution of the equation is

$$u(x) = 2 \cos(\sqrt{3}x + 1) / 3.$$

The function $g(x)$ is as follows:

$$\begin{aligned}
 g(x) = & \left(\frac{2}{3}\right) \cos(\sqrt{3}x + 1) - \left(\frac{1}{3}\right) \sqrt{3} x \cos(x) \sin(1) \\
 & + \left(\frac{2}{3}\right) \cos(x)\cos(1) - \left(\frac{1}{3}\right) x \sin(x) \cos(1) \\
 & - \left(\frac{1}{3}\right) \sqrt{3} \sin(x) \sin(1) - \left(\frac{2}{3}\right) \cos(\sqrt{3} x) \cos(1) \\
 & + \left(\frac{2}{3}\right) \sin(\sqrt{3} x) \sin(1).
 \end{aligned}$$

An approximate solution of this integral equation will first be calculated using the composite trapezoidal rule, then using polynomial and finally using non-polynomial piecewise linear splines.

When we use the linear piecewise polynomial splines with the 128 nodes on the interval $[0,1]$ we receive the plot of the error between the exact solution of the equation and the numerical solution obtained with the application of the linear polynomial splines. This plot is shown in Fig.1. When we use the composite trapezoidal rule with the 128 nodes on the interval $[0,1]$ we receive the plot of the error between the exact solution of the equation and the numerical solution obtained with the application of the composite trapezoidal rule. This plot is shown in Fig.2. Note that the solution is obtained in the form of a grid function, but the points on the graph are connected by a solid line for visualization clarity.

Table 3 show the maximum of the error in absolute value between the exact solution of the equation and the numerical

solution obtained with the application of the linear polynomial splines (column 2). The maximum of the error in absolute value obtained with the application of the non-polynomial splines when $\varphi(x) = \exp(x)$ is given in the column 3. The maximum of the error in absolute value obtained with the application of the composite trapezoidal rule is presented in the fourth column of Table 3. The number of nodes (n) is given in the first column of Table 3.

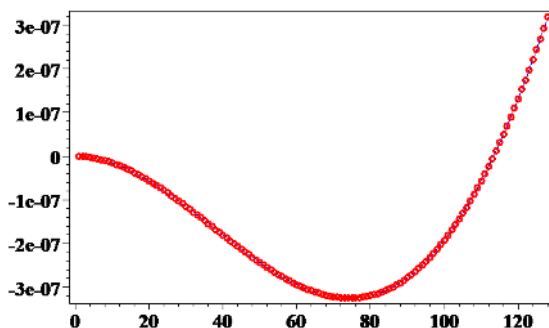


Fig.1. The plot of error between the exact solution of the equation (problem 1) and the numerical solution obtained with the application of the linear polynomial splines when we use 128 nodes

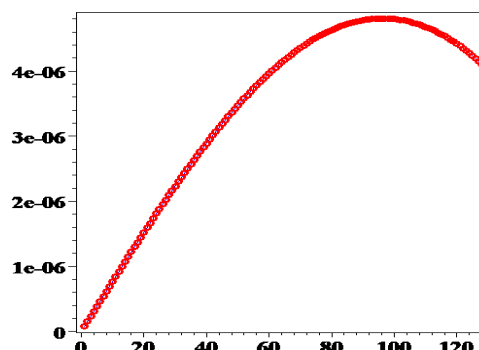


Fig.2. The plot of error between the exact solution of the equation (problem 1) and the numerical solution obtained with the application of the composite trapezoidal rule when we use 128 nodes

Table 3. The maximum of the error in absolute value between the exact solution of the equation from problem 1 and the numerical solution obtained with the application of the linear polynomial splines, non-polynomial splines and the composite trapezoidal rule

n	The error obtained with the use of the linear polynomial splines	The error obtained with the use of the non-polynomial splines $\varphi(x) = \exp(x)$	The error obtained with the composite trapezoidal rule application
32	$0.518 \cdot 10^{-5}$	$0.350 \cdot 10^{-4}$	$0.770 \cdot 10^{-4}$
64	$0.130 \cdot 10^{-5}$	$0.875 \cdot 10^{-5}$	$0.192 \cdot 10^{-4}$
128	$0.324 \cdot 10^{-6}$	$0.219 \cdot 10^{-5}$	$0.481 \cdot 10^{-5}$

Now let us solve some equations from paper [13] and compare the results with the result obtained with the polynomial and the non-polynomial splines. Let us compare the results with the result obtained with the composite trapezoidal rule.

Problem 2. We take the equation 7 from paper [13].

$$u(x) = \exp(-x) + x \exp(x) - \int_0^x \exp(x+t)u(t)dt, \quad x \in [0, 1].$$

The exact solution of the equation is $u(x) = \exp(-x)$

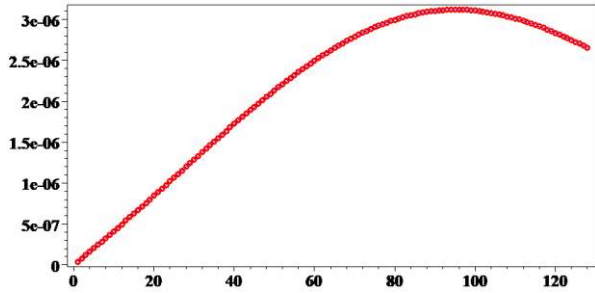


Fig.3. The plot of error between the exact solution of the equation (problem 2) and the numerical solution obtained with the application of the linear polynomial splines when we use 128 nodes

When we use the linear piecewise polynomial splines with the 64 nodes on the interval $[0, 1]$ we receive the plot of the error between the exact solution of the equation and the numerical solution obtained with the application of the linear polynomial splines. This plot is shown in Fig.3. When we use the non-polynomial splines ($\varphi(x) = \exp(-x)$) with the 64 nodes on the interval $[0,1]$ we receive the plot of the error between the exact solution of the equation and the numerical solution. This plot is shown in Fig.4.

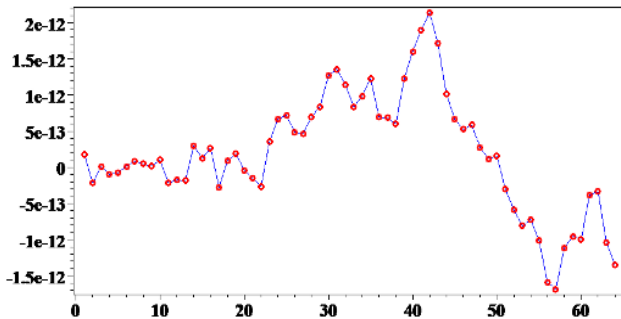


Fig.4. The plot of error between the exact solution of the equation (problem 2) and the numerical solution obtained with the application of the non-polynomial splines ($\varphi(x) = \exp(-x)$) when we use 64 nodes

When we use the non-polynomial splines ($\varphi(x) = \exp(x)$) with the 128 nodes on the interval $[0,1]$ we receive the plot of the error between the exact solution of the equation and the numerical solution. The plot is shown in Fig.5.

Table 4 shows the maximum of the error in absolute value between the exact solution of the equation and the numerical solution obtained with the application of the linear polynomial splines (column 2). The maximum of the error in absolute value obtained with the application of the non-polynomial splines ($\varphi(x) = \exp(x)$) is presented in the third column of Table 4. The number of nodes (n) is given in the first column of Table 4. The application of the composite trapezoidal rule for this equation gives us the exact solution.

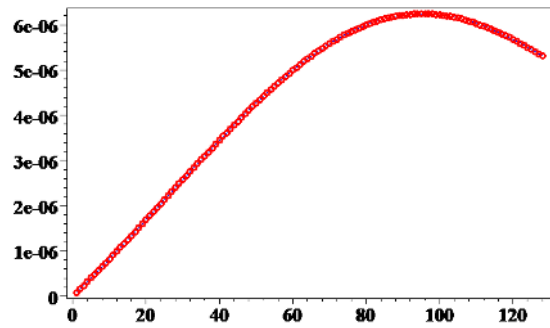


Fig.5. The plot of error between the exact solution of the equation (problem 2) and the numerical solution obtained with the application of the non-polynomial splines ($\varphi(x) = \exp(x)$) when we use 128 nodes

Table 4. The maximum of the error in absolute value between the exact solution of the equation from problem 2 and the numerical solution obtained with the application of the linear polynomial splines, non-polynomial splines

n	The error obtained with the use of the linear polynomial splines	The error obtained with the use of the non-polynomial splines $\varphi(x) = \exp(x)$
32	$0.499 \cdot 10^{-4}$	$0.999 \cdot 10^{-4}$
64	$0.125 \cdot 10^{-4}$	$0.250 \cdot 10^{-4}$
128	$0.312 \cdot 10^{-5}$	$0.624 \cdot 10^{-5}$

When we use the polynomial splines ($\varphi(x) = x$) with the 128 nodes on the interval $[0,1]$ we receive the plot of the error between the exact solution of the equation and the numerical solution. The plot is shown in Fig.6.

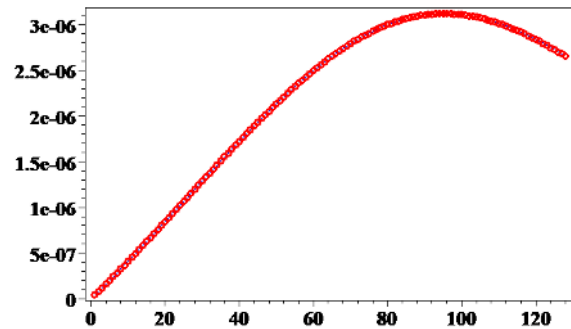


Fig.6. The plot of error between the exact solution of the equation (problem 2) and the numerical solution obtained with the application of the polynomial splines ($\varphi(x) = x$) when we use 128 nodes

Problem 3. We take the equation

$$u(x) = g(x) + \int_0^x (1 - (x-s)\exp(x))u(t)dt, \quad x \in [0, 1].$$

Here $g(x) = 1 - \exp(x) \sin(x) - x \exp(x) - \exp(x)/2 + \exp(2x)(\sin(x) + \cos(x))/2$.

The exact solution of the equation is

$$u(x) = \exp(x)(\cos(x) - \sin(x)).$$

Table 5 shows the maximum of the error in absolute value between the exact solution of the equation (Problem 3) and the numerical solution obtained with the application of the linear polynomial splines (column 2). The maximum of the error in

absolute value obtained with the application of the non-polynomial splines ($\varphi(x) = \exp(x)$) is presented in column 3, the error obtained with the composite trapezoidal rule application is given in the fourth column of Table 5. The number of nodes (n) is given in the first column of Table 5.

Table 5. The maximum of the error in absolute value between the exact solution of the equation from problem 3 and the numerical solution obtained with the application of the linear polynomial splines, non-polynomial splines and the composite trapezoidal rule

n	The error obtained with the use of the linear polynomial splines	The error obtained with the use of the non-polynomial splines $\varphi(x) = \exp(x)$	The error obtained with the composite trapezoidal rule application
32	$0.114 \cdot 10^{-3}$	$0.798 \cdot 10^{-4}$	$0.860 \cdot 10^{-3}$
64	$0.285 \cdot 10^{-4}$	$0.200 \cdot 10^{-4}$	$0.215 \cdot 10^{-3}$
128	$0.712 \cdot 10^{-5}$	$0.498 \cdot 10^{-5}$	$0.537 \cdot 10^{-4}$

The plot of error between the exact solution of the equation (problem 3) and the numerical solution obtained with the application of the exponential splines when we use 128 nodes is given in Fig.7. The plot of error between the exact solution of the equation and the numerical solution obtained with the application of the composite trapezoidal rule when we use 128 nodes is given in Fig.8.

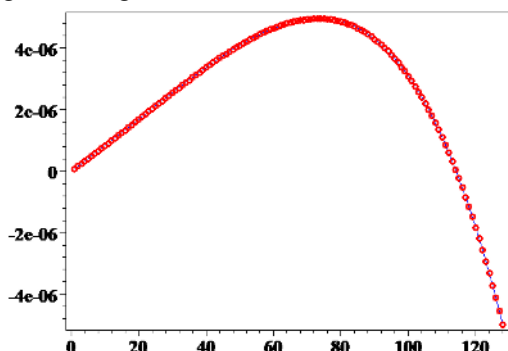


Fig.7. The plot of error between the exact solution of the equation (problem 3) and the numerical solution obtained with the application of the exponential splines when we use 128 nodes

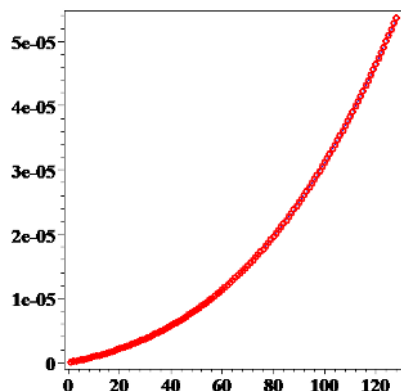


Fig.8. The plot of error between the exact solution of the equation (problem 3) and the numerical solution obtained with the application of the composite trapezoidal rule when we use 128 nodes

IV. APPLICATION TO SOLVING A NONLINEAR EQUATION

Let us now consider the application of splines of the second order of approximation to the solution of the nonlinear Volterra integral equation of the second kind.

$$u(x) + \int_a^x K(x, s, u(s))ds = f(x), \quad x \in [a, b].$$

We approximate $u(s)$ with the expression:

$$U(x) = u(x_j)\omega_j(x) + u(x_{j+1})\omega_{j+1}(x), \quad (3)$$

$$x \in [x_j, x_{j+1}].$$

Now transforming the integral $\int_{x_j}^{x_{j+1}} K(x, s, u(s))ds$ using formula (3), we obtain

$$\int_{x_j}^{x_{j+1}} K(x, s, u(s))ds \approx \int_{x_j}^{x_{j+1}} K(x, s, U(s))ds.$$

Here we take the nonlinear Volterra integral equation of the second kind of the form:

$$u(x) + \int_a^x K(x, s, u^2(s))ds = f(x), \quad x \in [a, b].$$

Now let us solve the equation from paper [13] and compare the results with the result obtained with splines. In problem 4 we consider problem 6 from paper [13].

Problem 4. We take the equation

$$u(x) = g(x) + \int_0^x xs^2u^2(s)ds, \quad x \in [0, 1].$$

Here

$$g(x) = (1 + 2x^4/9 - x^3/3 + 2x^2/3 + 11x/9)\log(x + 1)$$

$$+ \left(-\frac{x^4}{3} - \frac{x}{3}\right)(\ln(x + 1))^2 + 5x^3/18 - 11x^2/9 - 2x^4/27.$$

The exact solution is taken by $u(x) = \log(x + 1)$.

When we use the linear piecewise polynomial splines with the 128 nodes on the interval [0,1] we receive the plot of the error between the exact solution of the equation and the numerical solution obtained with the application of the linear polynomial splines which is shown in Fig.9.

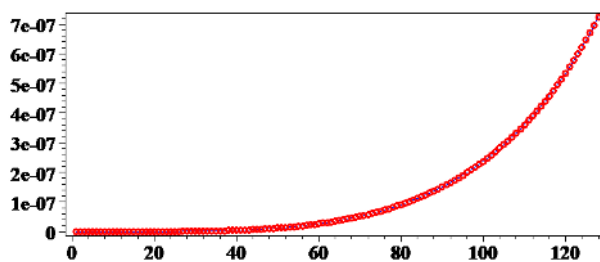


Fig.9. The plot of error between the exact solution of the equation (problem 4) and the numerical solution obtained with the application of the linear polynomial splines when we use 128 nodes

Table 6 shows the maximum of the error in absolute value between the exact solution of the equation (Problem 4) and the numerical solution obtained with the application of the linear polynomial splines (column 2). The maximum of the error in absolute value obtained with the application of the non-polynomial splines ($\varphi(x) = \exp(x)$) is presented in column 3. The maximum of the error in absolute value obtained with the application of the non-polynomial splines ($\varphi(x) = \exp(-x)$) is given in column 4. The number of nodes (n) is given in the first column of Table 6.

When we use the piecewise non-polynomial splines with the 128 nodes on the interval $[0,1]$ we receive the plots of the error between the exact solution of the equation and the numerical solution obtained with the application of the non-polynomial splines which are shown in Figs. 10-11.

Table 6. The maximum of the error in absolute value between the exact solution of the equation from problem 4 and the numerical solution obtained with the application of the linear polynomial splines, non-polynomial splines

n	The error obtained with the use of the linear polynomial splines	The error obtained with the use of the non-polynomial splines $\varphi(x) = \exp(x)$	The error obtained with the use of the non-polynomial splines $\varphi(x) = \exp(-x)$
32	$0.116 \cdot 10^{-4}$	$0.317 \cdot 10^{-4}$	$0.856 \cdot 10^{-5}$
64	$0.290 \cdot 10^{-5}$	$0.793 \cdot 10^{-5}$	$0.214 \cdot 10^{-5}$
128	$0.724 \cdot 10^{-6}$	$0.198 \cdot 10^{-5}$	$0.535 \cdot 10^{-6}$

Comment. This paper considers the case when it is possible to calculate the exact integral of the product of the kernel and the basis function. If this is not possible, then we can use a quadrature formula, for example, Simpson's rule.

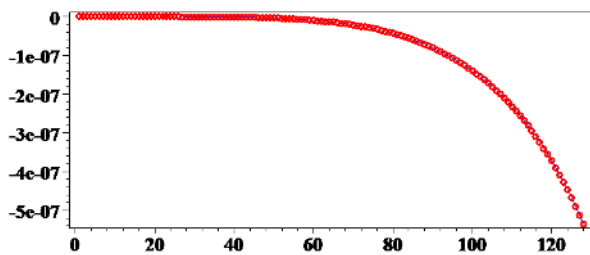


Fig.10. The plot of error between the exact solution of the equation (problem 4) and the numerical solution obtained with the application of the non-polynomial splines ($\varphi(x) = \exp(-x)$) when we use 128 nodes

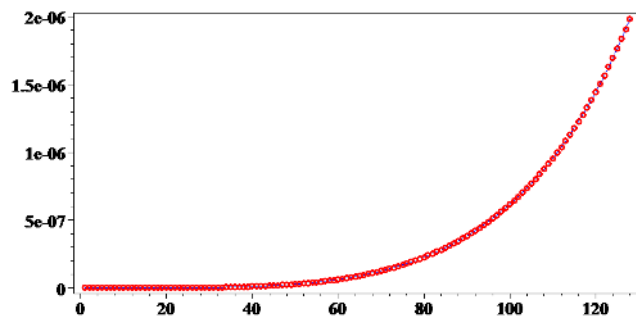


Fig.11. The plot of error between the exact solution of the equation (problem 4) and the numerical solution obtained with the application of the non-polynomial splines ($\varphi(x) = \exp(x)$) when we use 128 nodes

Problem 5. The applied problem of mathematical description of the gas-turbine engine state modification processes in the form of Volterra integral equation of second kind with separable kernel is solved in [18]. We construct an equation based on the data in paper [18]. First, let us take the model problem

$$u(x) = g(x) + \int_0^x (t + xt^2)u(t)dt, \quad x \in [0, 1].$$

We construct the function $g(x)$ by specifying the piecewise function $u(x)$ (see Fig.12). The plot of the solution of Volterra integral equation of second kind obtained with the application of the linear polynomial splines when 64 nodes were taken is given in Fig.12.

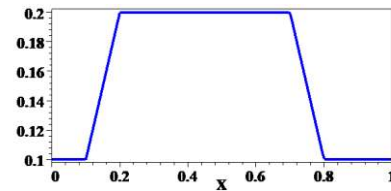


Fig.12. The plot of the piecewise function $u(x)$

The plot of error between the exact solution of the equation solution obtained with the application of the linear polynomial splines when 64 nodes were taken is given in Fig.14.

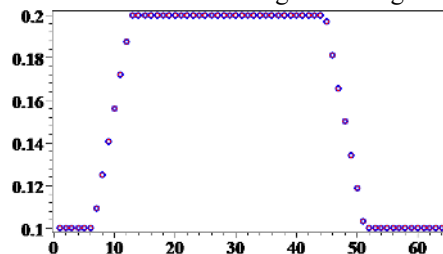


Fig.13. The plot of the solution obtained with the application of the linear polynomial splines when 64 nodes were taken/

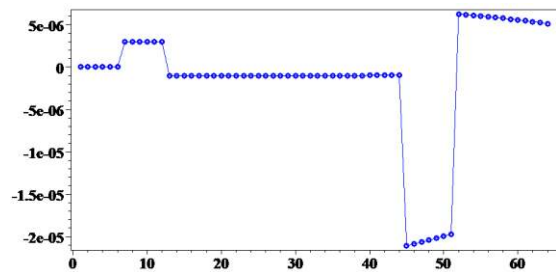


Fig.14. The plot of error between the exact solution of the equation solution obtained with the application of the linear polynomial splines when 64 nodes were taken.

Now let us take the next model problem

$$u(x) = g(x) + \int_0^x (t + xt^2)u(t)dt, \quad x \in [0, 1]$$

We construct the function $g(x)$ by specifying the piecewise function $u(x)$ (see Fig.15). The plot of the function $u(x)$ is similar to the plot from paper [18]. The plot of the solution of Volterra integral equation of second kind obtained with the application of the linear polynomial splines when 128 nodes were taken is given in Fig.16. The plot of error between the exact solution of the equation solution obtained with the application of the linear polynomial splines when 128 nodes were taken is given in Fig.17. Our method can find applications in [19] and [20].

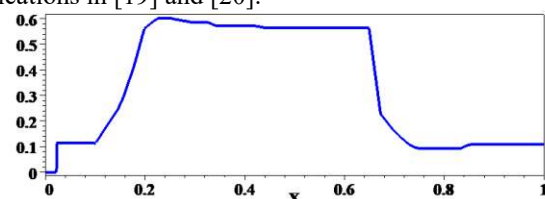


Fig.15. The plot of the piecewise function $u(x)$.

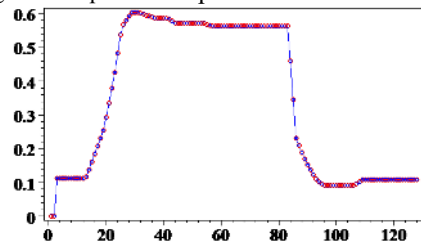


Fig.16. The plot of the solution obtained with the application of the linear polynomial splines when 128 nodes were taken.

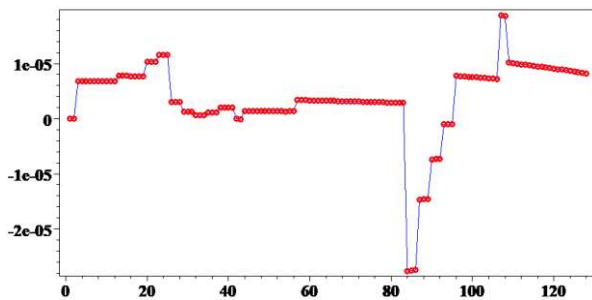


Fig.17. The plot of error between the exact solution of the equation solution obtained with the application of the linear polynomial splines when 128 nodes were taken.

V. CONCLUSION

In this paper, polynomial and non-polynomial splines of the second order of approximation are used to solve the linear and nonlinear models of the Volterra integral equations of the second kind. It is assumed, that the integral of the product of the kernel and the basis spline can be calculated exactly (in the form of a formula). The basis spline is the polynomial spline or non-polynomial spline of the second order of approximation. The numerical examples are done. The graphs of the error between the exact and approximate solutions at the different number of grid points are also drawn. As it is shown, the application of the local interpolation splines of the second order of approximation for the Volterra integral equations of the second kind can give an appropriate result, better than the composite trapezoidal rule. The non-polynomial splines sometimes can provide the result better than the polynomial splines. The application of the presented method to solve the model of an engineering problem presented in paper [18] gave a completely reliable result.

References

- [1] Ronald A. Devore, Karl Scherer, Quantitative Approximation, Academic Press, New York, London, Sydney, Toronto, San Francisco 1980, 324 p.
- [2] Zdeněk Kopal, Numerical Analysis with emphasis on the applications of numerical techniques to problems of infinitesimal calculus in single variable, Wiley, New York, 1955, 556 p.
- [3] Walter Gautschi, Numerical Analysis. An Introduction, Boston, Basel, Berlin, 1997, 506 p.
- [4] Galina Mehdiyeva, Vagif Ibrahimov, Mehrib Imanova, "Application of the Finite Differences Methods to Computation of Definite Integrals, WSEAS Transactions on System," vol. 19, pp 1-6, 2020.
- [5] Boris Shumilov, Shifted Cubic Spline Wavelets with Two Vanishing Moments on the Interval and a Splitting Algorithm, WSEAS Transactions On Systems, vol. 19, pp.149-158, 2020.
- [6] I.G. Burova, V.M. Ryabov, M.A.Kalnitskaia, A.V. Malevich, "The interpolation method for calculating eigenvalues of matrices," WSEAS Transactions on Systems and Control, vol.14, pp. 104-111, 2019.
- [7] N.Onn, M.Hussein, C.H.H.Tang, M.Z.M.Zain, M.Mohamad, W.Y.Lai, "Motion control of human bipedal model in sagittal planem," WSEAS Transactions on Systems and Control, vol.10, pp. 160-171, 2015.
- [8] H. Du, Z.Chen, "A new reproducing kernel method with higher convergence order for solving a Volterra–Fredholm integral equation," Applied Mathematics Letters, vol.102, paper 106117, 2020.
- [9] Galina Mehdiyeva, Vagif Ibrahimov, Mehriban Imanova, "On the Construction of the Advanced Hybrid Methods and Application to Solving Volterra Integral Equation," WSEAS Transactions on Systems and Control, vol.14, pp 183-189, 2019.
- [10] R.I. Esa, A.J. Saleh, "Numerical treatment of first order Volterra integro-differential equation using non-polynomial spline functions," Iraqi Journal of Science, Special Issue, pp. 114-121, 2020.
- [11] M.N.Sahlan, "Convergence of approximate solution of mixed Hammerstein type integral equations," Boletim da Sociedade Paranaense de Matematica, vol.38 (2), pp. 61-74, 2020.
- [12] K.Maleknejad, J.Rashidinia, H. Jalilian, "Non-polynomial spline functions and quasi-linearization to approximate nonlinear volterra integral equation," \ Filomat, vol.32 (11), pp. 3947-3956, 2018.
- [13] M. Asif, I. Khan, N. Haider, Q. Al-Mdallal, "Legendre multi-wavelets collocation method for numerical solution of linear and nonlinear integral equations." Alexandria Engineering Journal, vol. 59, pp.5099-5109, 2020.
- [14] I.G.Burova, N.S. Domnin, "On the solution of the Fredholm equation with the use of quadratic integro-differential splines," Lecture Notes in Electrical Engineering, vol.574, pp. 35-41, 2019.
- [15] I.G.Burova, N.S. Domnin, A.E.Vezhlev, A.V.Lebedeva, A.N.Pakulina, "On the solution of the Fredholm equation of the second kind," WSEAS Transactions on Mathematics, vol.17, pp. 319-328, 2018.
- [16] I.G.Burova, E.F. Muzafarova, I.I. Narbutovskikh, "Local splines of the second and third order, complex-valued splines and image processing," International Journal of Circuits, Systems and Signal Processing, vol.13, pp. 419-429, 2019.
- [17] I.G.Burova, "On left integro-differential splines and Cauchy problem," International Journal of Mathematical Models and Methods in Applied Sciences, vol.9, pp. 683-690, 2015.
- [18] V. Myrhorod, and I. Hvozdeva, "On one solution of Volterra integral equations of second kind," AIP Conference Proceedings 1773, 040006, 2016, doi: 10.1063/1.4964969, Published by the American Institute of Physics, View online: <http://dx.doi.org/10.1063/1.4964969>

- [19] Nicholas D. Assimakis, "Kalman Filter Gain Elimination in Linear Estimation", Engineering World, Published by International Academic Publishers, pp. 183-188, Volume 2, 2020.
- [20] Nicholas Assimakis, Maria Adam, Grigorios Tziallas, Lainiotis Information Filter, Engineering World, Published by International Academic Publishers, pp. 270-273, Volume 2, 2020.

**Creative Commons Attribution License 4.0
(Attribution 4.0 International, CC BY 4.0)**

This article is published under the terms of the Creative Commons Attribution License 4.0

https://creativecommons.org/licenses/by/4.0/deed.en_US