Application Local Polynomial and Non-polynomial Splines of the third order of approximation for the construction of the Numerical Solution of the Volterra Integral Equation of the Second Kind

I.G.BUROVA
Department of Computational Mathematics
St. Petersburg State University
7/9 Universitetskaya nab., St.Petersburg, 199034
RUSSIA

Abstract: - The present paper is devoted to the application of local polynomial and non-polynomial interpolation splines of the third order of approximation for the numerical solution of the Volterra integral equation of the second kind. Computational schemes based on the use of the splines include the ability to calculate the integrals over the kernel multiplied by the basis function which are present in the computational methods. The application of polynomial and nonpolynomial splines to the solution of nonlinear Volterra integral equations is also discussed. The results of the numerical experiments are presented.

Key-Words: - Volterra integral equation, non-polynomial spline, polynomial spline

Received: December 21, 2019. Revised: January 25, 2021. Accepted: February 19, 2021.
Published: March 2, 2021.

1 Introduction

Here we study the application of the local polynomial and non-polynomial interpolation splines of the third order of approximation for the construction the numerical scheme for the solution of the Volterra integral equation of the second kind. There are many numerical methods for solving Volterra integral equations of the second kind. The most common numerical methods are based on the use of quadrature formulas.

In connection with the emerging needs for constructing methods of high accuracy, many researchers, again, resort to modernizing the known methods for solving integral equations and construction the new ones. The authors of papers [1]-[10] devoted a lot of attention to the modification of the known numerical methods and the construction of new numerical methods for solving integral equations. In study [1] a numerical scheme for approximating the solutions of the nonlinear system of fractional-order Volterra-Fredholm integral differential equations was proposed. The main characteristic of this approach is that it reduces such problems to a linear system of algebraic equations.

In paper [2], a new and efficient method for solving the three-dimensional Volterra-Fredholm integral equations of the second kind, first kind and even singular type of these equations is presented. Here, the authors discuss three variable Bernstein polynomials and their properties. This method has several advantages in reducing the computational burden with a good degree of accuracy. Furthermore, the authors obtain an error bound for this method. A computational technique based on a special family of the Mëuntz-Legendre polynomials to solve a class of Volterra-Fredholm integral equations is presented in paper [3]. The proposed method reduces the integral equation into algebraic equations via the Chebyshev-Gauss-Lobatto points, so that the system matrix coefficients are obtained by the least squares approximation method. The useful properties of the Jacobi polynomials are exploited to analysis the approximation error.

Spline functions were used to propose a new scheme for solving the linear Volterra–Fredholm integral equations of the second kind in paper [4].

Two types of non-polynomial spline functions (linear, and quadratic) were used in paper [5] to find

the approximate solution of Volterra integro–differential equations.

A computational method for solving nonlinear Volterra-Fredholm Hammerstein integral equations is proposed in [6], by using compactly supported semiorthogonal cubic B-spline wavelets as basis functions. The non-polynomial spline basis and Quasi-linearization method to solve the nonlinear Volterra integral equation were used in [7]. A new collocation technique for the numerical solution of the Fredholm, Volterra and mixed Volterra-Fredholm integral equations of the second kind is introduced in [8], and a numerical integration formula on the basis of the linear Legendre multi-wavelets is also developed in [8].

Note that in papers [4]-[10] splines are used to construct new numerical methods. The construction of various splines and wavelet splines is considered in papers [11]-[16]. The approximations with splines on the irregular set of nodes are of particular interest [13].

The application of the generalized Haar spaces is sometimes very useful [14]. Paper [15] 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 [16] 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 [17] the application of the finite-difference methods are investigated to compute the definite integrals.

At present, the theory of approximation by local interpolation splines continues to evolve. Approximation with local polynomial and local non-polynomial splines of the Lagrange types can be used in many applications. Approximation with the use of these splines is constructed on each mesh interval separately as a linear combination of the products of the values of the function at the grid nodes and basic functions. The basis functions are defined as a solution of a system of linear algebraic equations (approximation relations). The approximation relations are formed from the conditions of accuracy of approximation on the functions forming the Chebyshev system. The constructed basic splines provide an approximation of the prescribed order which is equal to the number of equations in the system, or, in other words, it is equal to the number of grid intervals in the support of the basic splines. Using the basic splines, we can construct continuous types of approximation [10]-[12], [16]. This paper continues the construction of numerical methods based on the use of spline approximations [10]. The proposed numerical methods extend the set of known numerical methods for solving integral equations [18].

The paper is organized as follows. Section 2 discusses the theoretical aspects of the application of polynomial and non-polynomial splines of the second order of approximation. Section 3 considers the properties of polynomial and non-polynomial splines of the third order of approximation. A numerical method for solving the Volterra equation of the second kind is also proposed here. Section 4 presents the results of the numerical solution of the Volterra equations of the second kind using the trapezoidal method, using polynomial and non-polynomial splines of the second order of approximation, as well as using splines of the third order of approximation.

2 Application of Splines of the Second Order of Approximation

In paper [10] the numerical solution of Volterra-Fredholm integral equations of the second kind was constructed with the use of local splines of the second order of approximation.

As it is shown in paper [11], if the functions φ_1, φ_2 form a Chebyshev system, then the basis functions ω_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 take $\varphi_1(x) = 1, \varphi_2(x) = \varphi(x)$. We constructed a non-polynomial approximation of function $u(x)$, on each grid interval $[x_j, x_{j+1}]$ in the form:

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

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 papers [10]-[11]), but they all have an approximation error of the order of $O(h^2)$. This approximation error can be obtained using Taylor's theorem. Let ordered distinct nodes $\{x_j\}$ be such that $x_{j+1} - x_j = h$. But we can apply the method described in paper [11]. Let us denote

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

$$h = h_j = x_{j+1} - x_j.$$

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

Then, for $\varphi(x) = \exp(x), \exp(-x)$, $x \in [x_j, x_{j+1}]$ we have

$$|u(x) - U(x)| \le K_2 h^2 \|Lu\|,$$
$$x \in [x_j, x_{j+1}],\ K_2 > 0.$$

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 can be found in paper [10].

Theorem 2. Let $u \in \mathrm{C}^2[a,b]$. To approximate the function $u(x)$, $x \in [x_j, x_{j+1}]$, by spline (4), the following inequality is valid:

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

Proof can be found in paper [10].

Theorem 3. 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

$$U(x) = u(x), u(x) = \varphi_1(x), \varphi_2(x),$$
$$\text{for } x \in [x_j, x_{j+1}],$$

$$\varphi_1(x) = \cos(x), \varphi_2(x) = \sin(x).$$

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

$$|u(x) - U(x)| \le K_2 h^2 \|Lu\|,$$
$$x \in [x_j, x_{j+1}],\ K_2 > 0.$$

Here $Lu = u''(x) + u(x)$.

Proof can be found in paper [19].

The linear Volterra equation of the second kind has the form:

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

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

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

$$\widetilde{u}(x_0) = f(x_0),$$

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

$$\widetilde{u}(x_1)\int_{x_0}^{x_1} K(x_1,s)\omega_1(s)ds = f(x_1),$$

$$\cdots\cdots\cdots\cdots$$

$$\widetilde{u}(x_n) + \sum_{s=0}^{n-2} W_s(x_n)$$

$$+\widetilde{u}(x_{n-1})\int_{x_{n-1}}^{x_n} K(x_n,s)\omega_{n-1}(s)ds$$

$$+\widetilde{u}(x_n)\int_{x_{n-1}}^{x_n} K(x_n,s)\omega_n(s)ds = f(x_n).$$

We determine the unknowns $\widetilde{u}(x_k), k = 2, \dots, n$, sequentially, starting from the first equation.

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 can be applied to a quadrature formula with an error not less than $O(h^3)$.

3 Application of Splines of the Third Order of Approximation

Suppose $K(x,\cdot), u \in \mathrm{C}^3[a,b]$. Let us consider the numerical solution of the Volterra equation of the second kind with the same assumptions

$$u(x) + \int_a^x K(x,s)u(s)ds = f(x) \qquad (2)$$

using splines of the third order of approximation. On a finite interval $[a,b]$, it is necessary to apply left and right spline approximations. First, we recall the features of the approximation of functions by splines near the right end of the interval $[a,b]$. Continuous polynomial approximation near the right end of the interval $[a,b]$ uses the basic spline $\omega_j^L(x)$ of the form:

$$\omega_j^L(x) = \frac{(x-x_{j+1})(x-x_{j+2})}{(x_j-x_{j+1})(x_j-x_{j+2})}, x \in [x_{j+1}, x_{j+2}],$$

$$\omega_j^L(x) = \frac{(x-x_{j+1})(x-x_{j-1})}{(x_j-x_{j+1})(x_j-x_{j-1})}, x \in [x_j, x_{j+1}],$$

$$\omega_j^L(x) = \frac{(x-x_{j-1})(x-x_{j-2})}{(x_j-x_{j-1})(x_j-x_{j-2})}, x \in [x_{j-1}, x_j],$$

$$\omega_j^L(x) = 0, x \notin [x_{j-1}, x_{j+2}].$$

The support of the basic spline occupies three adjacent grid intervals, supp $\omega_j^L = [x_{j-1}, x_{j+2}]$.The function $u(x), x \in [x_j, x_{j+1}]$, can be approximated by the polynomial spline (see [11]) using the form:

$$U_j^L(x) = u(x_{j-1})\omega_{j-1}^L(x) + u(x_j)\omega_j^L(x) + u(x_{j+1})\omega_{j+1}^L(x), \qquad (3)$$

where

$$\omega_{j-1}^L(x) = \frac{(x-x_j)(x-x_{j+1})}{(x_{j-1}-x_j)(x_{j-1}-x_{j+1})}, x \in [x_j, x_{j+1}],$$

$$\omega_j^L(x) = \frac{(x-x_{j+1})(x-x_{j-1})}{(x_j-x_{j+1})(x_j-x_{j-1})}, x \in [x_j, x_{j+1}],$$

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

Now let us recall how the approximation is constructed near the left end of the interval $[a,b]$.

The continuous polynomial approximation near the left end of the interval $[a,b]$ uses the basic spline $\omega_j^R(x)$ of the form:

$$\omega_j^R(x) = \frac{(x-x_{j+1})(x-x_{j+2})}{(x_j-x_{j+1})(x_j-x_{j+2})}, x \in [x_j, x_{j+1}],$$

$$\omega_j^R(x) = \frac{(x-x_{j+1})(x-x_{j-1})}{(x_j-x_{j+1})(x_j-x_{j-1})}, x \in [x_{j-1}, x_j],$$

$$\omega_j^R(x) = \frac{(x-x_{j-1})(x-x_{j-2})}{(x_j-x_{j-1})(x_j-x_{j-2})}, x \in [x_{j-2}, x_{j-1}],$$

$$\omega_j^R(x) = 0, x \notin [x_{j-2}, x_{j+1}].$$

The support of the basic spline occupies three adjacent grid intervals, supp $\omega_j^R = [x_{j-2}, x_{j+1}]$.

The function $u(x), x \in [x_j, x_{j+1}]$, can be approximated by the polynomial spline (see [11]) using the form:

$$U_j^R(x) = u(x_j)\omega_j^R(x) + u(x_{j+1})\omega_{j+1}^R(x) + u(x_{j+2})\omega_{j+2}^R(x), \qquad (4)$$

where

$$\omega_j^R(x)=\frac{(x-x_{j+1})(x-x_{j+2})}{(x_j-x_{j+1})(x_j-x_{j+2})}, x\in[x_j,x_{j+1}],$$

$$\omega_{j+1}^R(x)=\frac{(x-x_{j+2})(x-x_j)}{(x_{j+1}-x_{j+2})(x_{j+1}-x_j)}, \quad x\in[x_j,x_{j+1}],$$

$$\omega_{j+2}^R(x)=\frac{(x-x_{j+1})(x-x_j)}{(x_{j+2}-x_{j+1})(x_{j+2}-x_j)}, \quad x\in[x_j,x_{j+1}].$$

The approximation properties of these basic splines are well studied. Let us denote $\| u''' \|_{[x_j,x_{j+1}]}=\max\limits_{[a,b]}|u'''(x)|$. The following theorem was proved in [11].

Theorem 4. Let $u\in C^3[a,b]$.To approximate the function $u(x)$, $x\in[x_j,x_{j+1}]$, by spline (3), (4), the following inequality is valid:

$$\left|u(x)-U_j^L(x)\right|\le Kh^3\ \| u''' \|_{[x_{j-1},x_{j+1}]}.$$

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

$$\left|u(x)-U_j^R(x)\right|\le Kh^3\ \| u''' \|_{[x_j,x_{j+2}]}.$$

Proof. It is easy to notice that U_j^R is an interpolation polynomial of the first degree, and x_j, x_{j+1} are the interpolation nodes, $U_j^R(x_j)=u(x_j), U_j^R(x_{j+1})=u(x_{j+1}), U_j^R(x_{j+2})=u(x_{j+2})$. Using the remainder term we get

$$u(x)-U_j^R(x)=\frac{u'''(\tau)}{3!}(x-x_j)(x-x_{j+1})(x-x_{j+2}).$$

It follows that

$$\max_{x\in[x_j,x_{j+2}]}|u(x)-U_j^R(x)|\le 0.0625h^3\max_{[x_j,x_{j+2}]}|u'''|.$$

Thus, $K=0.0625$.

We can also apply a non-polynomial approximation. As is known, in a number of cases the use of non-polynomial splines provides a better approximation, although the order of approximation remains the same.

Let us recall the features of constructing an approximation using non-polynomial splines.

First, consider the construction of basic splines near the left end of the interval $[a,b]$.As shown in paper [10], if the functions φ_1, φ_2, φ_3 form a Chebyshev system, then the basis functions $\omega_k^R(x)$, $k=j,j+1$,j+2 can be determined by solving the system of equations

$$\varphi_1(x_j)\omega_j^R(x)+\varphi_1(x_{j+1})\,\omega_{j+1}^R(x)+\varphi_1(x_{j+2})\,\omega_{j+2}^R(x)=\varphi_1(x),$$

$$\varphi_2(x_j)\omega_j^R(x)+\varphi_2(x_{j+1})\,\omega_{j+1}^R(x)+\varphi_2(x_{j+2})\,\omega_{j+2}^R(x)=\varphi_2(x),$$

$$\varphi_3(x_j)\omega_j^R(x)+\varphi_3(x_{j+1})\,\omega_{j+1}^R(x)+\varphi_3(x_{j+2})\,\omega_{j+2}^R(x)=\varphi_3(x),$$

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

Suppose that the determinant of the system does not equal zero. We find the basic splines near the right end of the interval $[a,b]$ in a similar way. Basic splines are determined by solving a system of equations:

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

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

$$\varphi_3(x_{j-1})\omega_{j-1}^L(x)+\varphi_3(x_j)\,\omega_j^L(x)+\varphi_3(x_{j+1})\,\omega_{j+1}^L(x)=\varphi_3(x),\quad x\in[x_j,x_{j+1}].$$

Suppose that the determinant of the system does not equal zero.

Theorem 5. Let function $u(x)$ be such that $u\in C^3[a,b]$. Suppose the basis splines $\omega_j(x),\omega_{j+1}(x),\omega_{j-1}(x)$, are constructed when

$$U_j^L(x)=u(x_{j-1})\omega_{j-1}^L(x)+u(x_j)\omega_j^L(x)+u(x_{j+1})\omega_{j+1}^L(x),$$

$$U_j^L(x)=u(x), u(x)=\varphi_1(x),\varphi_2(x),\varphi_3(x), \text{ for } x\in[x_j,x_{j+1}],$$

$$\varphi_1(x)=1,\varphi_2(x)=\exp(x),\varphi_3(x)=\exp(-x).$$

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

$$|u(x) - U_j^L(x)| \le K_3 h^3 \|Lu\|_{[x_{j-1}, x_{j+1}]},$$
$$x \in [x_j, x_{j+1}],\ K_3 > 0.$$

Here $Lu = u'(x) - u'''(x)$.

Proof follows from the method described in [11]. In the case of the non-polynomial splines as it was shown in paper [11] we construct a homogeneous equation, which has a fundamental system of solutions $\varphi_1(x) = 1, \varphi_1(x) = \exp(x), \varphi_2(x) = \exp(-x)$.

We have

$$Lu = \begin{vmatrix} 1 & e^{-x} & e^{x} & u(x) \\ 0 & -e^{-x} & e^{x} & u'(x) \\ 0 & e^{-x} & e^{x} & u''(x) \\ 0 & -e^{-x} & e^{x} & u'''(x) \end{vmatrix} =$$
$$2u'(x) - 2u'''(x) = 0.$$

It is easy to see, that the Wronskian $W(x) = \begin{vmatrix} 1 & e^{-x} & e^{x} \\ 0 & -e^{-x} & e^{x} \\ 0 & e^{-x} & e^{x} \end{vmatrix} = -2$ does not equal zero. 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 $u(x), x \in [x_j, x_{j+1}]$.

In particular, we obtain using the method from [11] the formula for $u(x)$ when $x \in [x_j, x_{j+1}]$:

$$u(x) = \int_{x_j}^{x} (u'(t) - u'''(t))(\exp(x-t) + 2\exp(t-x) - 2)\,dt + c_1 + c_2 \exp(x) + c_3 \exp(-x).$$

Here c_1, c_2, c_3 are some arbitrary constants, $x \in [x_j, x_{j+1}]$. We construct the approximation of $u(x)$ in the form:

$$U_j^L(x) = u(x_{j-1})\omega_{j-1}^L(x) + u(x_j)\omega_j^L(x) + u(x_{j+1})\omega_{j+1}^L(x),$$

Thus, using the results from paper [11], we get

$$|u(x) - U(x)| \le K_3 h^3 \|Lu\|_{[x_{j-1}, x_{j+1}]}, \quad x \in [x_j, x_{j+1}],$$

The proof is complete.

Theorem 6. Let function $u(x)$ be such that $u \in C^3[a,b]$. Suppose the basis splines $\omega_j(x), \omega_{j+1}(x), \omega_{j+2}(x),$ are constructed when

$$U_j^R(x) = u(x_j)\omega_j^R(x) + u(x_{j+1})\omega_{j+1}^R(x) + u(x_{j+2})\omega_{j+2}^R(x),$$
$$U_j^R(x) = u(x), u(x) = \varphi_1(x), \varphi_2(x), \varphi_3(x), \text{ for } x \in [x_j, x_{j+1}],$$
$$\varphi_1(x) = 1, \varphi_2(x) = \cos(x), \varphi_3(x) = \sin(x).$$

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

$$|u(x) - U_j^R(x)| \le K_2 h^3 \parallel Lu \parallel_{[x_j, x_{j+2}]},$$
$$x \in [x_j, x_{j+1}],\ K_2 > 0.$$

Proof follows from the method described in [11] and is similar to the proof of Theorem 5. In particular, we obtain using the method from [11] the formula for $u(x)$ when $x \in [x_j, x_{j+1}]$:

$$u(x) = 2\int_{x_j}^{x} (u'''(t) + u'(t))\sin^2\frac{x-t}{2}\,dt + c_1 + c_2 \sin(x) + c_3 \cos(x).$$

Here c_1, c_2, c_3 are arbitrary constants

Here $Lu = u'''(x) + u'(x)$. The proof is complete.

Theorem 7 is similar to Theorem 6.

Theorem 7. Let function $u(x)$ be such that $u \in C^3[a,b]$. Suppose the basis splines $\omega_j(x), \omega_{j+1}(x), \omega_{j-1}(x),$ are constructed when

$$U_j^L(x) = u(x_{j-1})\omega_{j-1}^L(x) + u(x_j)\omega_j^L(x) + u(x_{j+1})\omega_{j+1}^L(x),$$
$$U_j^L(x) = u(x), u(x) = \varphi_1(x), \varphi_2(x), \varphi_3(x), \text{ for } x \in [x_j, x_{j+1}],$$
$$\varphi_1(x) = 1, \varphi_2(x) = \cos(x), \varphi_3(x) = \sin(x).$$

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

$$|u(x) - U_j^L(x)| \le K_2 h^3 \parallel Lu \parallel_{[x_{j-1}, x_{j+1}]},$$
$$x \in [x_j, x_{j+1}],\ K_2 > 0.$$

Here $Lu = u'''(x) + u'(x)$.

Proof follows from the method described in [11] and is similar to the proof of Theorem 5. In particular, we obtain using the method from [11] the formula for $u(x)$ when $x \in [x_j, x_{j+1}]$:

$$u(x) = 2\int_{x_j}^{x}(u'''(t) + u'(t))\sin^2\frac{x-t}{2}dt + c_1 + c_2\sin(x) + c_3\cos(x).$$

Here c_1, c_2, c_3 are arbitrary constants. The proof is complete.

The cases when $U_j^R(x) = u(x)$, $u(x) = \varphi_1(x), \varphi_2(x), \varphi_3(x)$, for $x \in [x_j, x_{j+1}]$, or $U_j^L(x) = u(x), u(x) = \varphi_1(x), \varphi_2(x), \varphi_3(x)$, $\varphi_1(x) = 1, \varphi_2(x) = \exp(x), \varphi_3(x) = \exp(2x)$ and $\varphi_1(x) = 1, \varphi_2(x) = \exp(-x), \varphi_3(x) = \exp(-2x)$ are investigated in the similar way. In all these cases we get a third-order approximation, that is, an error of the order $O(h^3)$. Here we note that in the first case $(1, \exp(x), \exp(2x))$ the solution on the interval $x \in [x_j, x_{j+1}]$, can be represented in the form

$$u(x) = 2\int_{x_j}^{x}\big(u'''(t) - 3u''(t) + 2u'(t)\big)(\exp(3t)\,(\frac{\exp(2x-2t)}{2} - \exp(\mathrm{x}-\mathrm{t}) + 1/2)\,dt + c_1 + c_2\exp(x) + c_3\exp(2x).$$

In the second case $(1, \exp(-x), \exp(-2x))$ the solution on the interval $x \in [x_j, x_{j+1}]$, can be represented in the form:

$$u(x) = 2\int_{x_j}^{x}\big(u'''(t) + 3u''(t) + 2u'(t)\big) \times \exp(-3t)\left(\frac{\exp(2t-2x)}{2} - \exp(\mathrm{t}-\mathrm{x}) + \frac{1}{2}\right)dt + c_1 + c_2\exp(-x) + c_3\exp(-2x).$$

We can do some experiments using Maple when $h = 0.1$ on $[-1,1]$. Tables 1, 2 show the actual errors of approximation of some functions obtained with the use of the polynomial and non-polynomial splines of the third order of approximation. Table 3 shows the actual errors of approximation of some functions obtained with the use of the polynomial and non-polynomial splines of the second order of approximation.

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

$u(x)$	$\varphi_1(x) = 1,$ $\varphi_2(x) = x,$ $\varphi_2(x) = x^2$	$\varphi_1(x) = 1$ $\varphi_2(x)$ $= \cos(x),$ $\varphi_3(x)$ $= \sin(x)$	$\varphi_1(x) = 1,$ $\varphi_2(x)$ $= \exp(-x),$ $\varphi_3(x)$ $= \exp(x)$
$\exp(x)$	0.000160	0.000320	0.0
$sin(x)$	0.0000641	0.0	0.000128
x^2	0.0	0.000127	0.000126
$\exp(-x)$	0.000172	0.000344	0.0
$sin(2x)$	0.000512	0.000384	0.000639
$\frac{1}{1+25x^2}$	0.0296	0.0294	0.0297

Table 2. The actual errors of approximation of some functions obtained with the use of the polynomial and non-polynomial splines of the third order of approximation

$u(x)$	$\varphi_1(x) = 1,$ $\varphi_2(x)$ $= \exp(x),$ $\varphi_3(x)$ $= \exp(2x)$	$\varphi_1(x) = 1,$ $\varphi_2(x)$ $= \exp(-x),$ $\varphi_3(x)$ $= \exp(-2x)$
$\exp(x)$	0.0	0.0
$sin(x)$	0.000200	0.000187
x^2	0.000666	0.000593
$\exp(-x)$	0.00108	0.0
$sin(2x)$	0.000843	0.000775
$\frac{1}{1+25x^2}$	0.0272	0.0311

Table 3. The actual errors of approximation of some functions obtained with the use of the polynomial and non-polynomial splines of the second order of approximation

$u(x)$	$\varphi_1(x) = 1,$ $\varphi_2(x) = x.$	$\varphi_1(x)$ $= \cos(x),$ $\varphi_2(x)$ $= \sin(x)$	$\varphi_1(x) = 1,$ $\varphi_2(x)$ $= \exp(-x)$
$\exp(x)$	0.00323	0.00647	0.00646
$sin(x)$	0.00102	0.0	0.00177
x^2	0.00250	0.00363	0.00487
$\exp(-x)$	0.00323	0.00647	0.0
$sin(2x)$	0.00498	0.00374	0.00558
$\frac{1}{1+25x^2}$	0.0418	0.0407	0.0443

Consider the approximation by polynomial splines. Let the second and third derivatives of the function

$u(x)$ be such that $\max_{[a,b]}|u'''| \le C$, $\max_{[a,b]}|u''| \le C, C = const > 0$. The example is function $u(x) = \exp(x)$The calculation results presented in Tables 1-3 confirm the fact, that the splines of the third order of approximation in this case give a smaller error than the splines of the second order of approximation if in both cases we apply the same step h. A similar statement can be formulated for non-polynomial splines. The theorems formulated above give asymptotic estimates. This allows us to hope that for sufficiently small h, the use of splines of the third order of approximation will give a smaller error than the use of splines of the second order of approximation.

Consider the following example. We will approximate the Runge function $u(x) = \frac{1}{1+25x^2}$ on a uniform grid of nodes built on the interval $[a,b] = [-1,1]$ with the step h . Fig. 1 shows a graph of the absolute value of the second derivative of the Runge function. Fig. 2 shows a graph of the absolute value of the third derivative of the Runge function. We use the results of Theorem 2 and Theorem 4.
Let us introduce the notations:

$$A(h) = h^2/8 \max_{[0,1]}|u''(x)|,$$
$$B(h) = 0.0625h^3 \max_{[0,1]}|u'''(x)|.$$

The plots of $A(h)$ (blue), and $B(h)$ (red) are given in Fig.3.
Solving the equation $A(h) = B(h)$, we find $h_0 \approx 0.171$. At this point, the graph lines intersect. When h is greater than this value ($h > h_0$), the theoretical error when using polynomial splines of the second order of approximation will be lesser then when using polynomial splines of the third order of approximation. It is easy to calculate that when $h = 0.3$ we get $A(h) \approx 0.985, B(h) \approx 0.562$.
Let $h = 1/3$. Fig. 4 shows plot of the actual error of approximation of the Runge function by splines of the second order of approximation. The maximum of the absolute value of the actual error is 0.0623. Fig. 5 shows plot of the actual error of approximation of the Runge function by splines of the third order of approximation. The maximum of the absolute value of the actual error is 0.236.

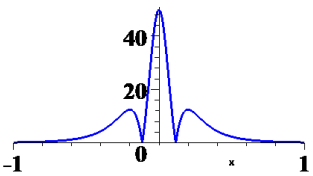

Fig.1. The plot of graph of the absolute value of the second derivative of the Runge function.

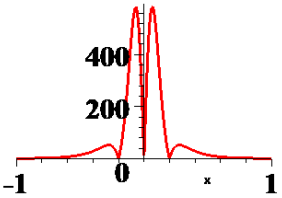

Fig.2. The plot of the graph of the absolute value of the third derivative of the Runge function

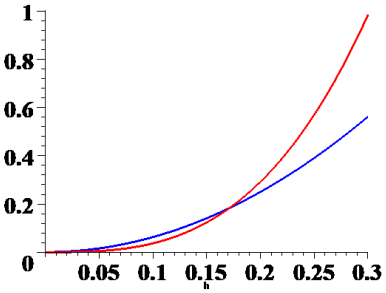

Fig.3. The plots of $A(h)$ (blue), and $B(h)$ (red).

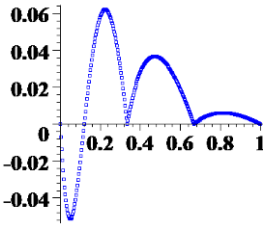

Fig.4. The plot of the actual error of approximation of the Runge function by splines of the second order of approximation

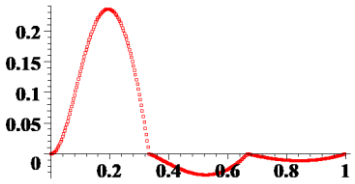

Fig.5. The plot of the actual error of approximation of the Runge function by splines of the third order of approximation

Thus, there are cases when linear polynomial splines will give a smaller approximation error than quadratic ones. therefore, to verify the result, both types of approximations should be applied.

Now let's apply splines to the calculation of the integral $\int_a^x K(x,s)u(s)ds$.

Transforming the integral $\int_{x_j}^{x_{j+1}} K(x,s)u(s)ds$, $j = 1, \dots n-1$, using formula (3), we obtain

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

$$u(x_{j-1})\int_{x_j}^{x_{j+1}} K(x,s)\omega_{j-1}(s)ds +$$

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

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

Let us introduce the notations:

$$V_j(x) = \tilde{u}(x_{j-1})\int_{x_j}^{x_{j+1}} K(x,s)\ \omega_{j-1}^L(s)ds +$$

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

$\tilde{u}(x_{j+1})\int_{x_j}^{x_{j+1}} K(x,s)\ \omega_{j+1}^L(s)ds,\ j = 1,\dots n-1.$

Using formula (4), we obtain

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

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

$+\tilde{u}(x_{j+2})\int_{x_j}^{x_{j+1}} K(x,s)\ \omega_{j+2}^R(s)ds, j = 0,\dots,n-2.$

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

$$\tilde{u}(x_k) + \sum_{s=0}^{k-2} Q_s(x_k) + \sum_{s=k-1}^{k-1} V_s(x_k) = f(x_k),$$

$$k = 0,\dots,n.$$

As a result of solving the system of equations, we obtain approximate values $\tilde{u}$ of the solution u to equation (2) at the grid nodes x_k. First we have $u(x_0) = f(x_0)$. Next, we solve the system of two equations and find $\tilde{u}(x_i)$, $i = 1,2$. Solving each next equation, we find the approximate values of the solution at the next grid points.

In the case of a grid consisting of three nodes, the system of equations has the form:

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

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

$$+\tilde{u}(x_1)\int_{x_0}^{x_1} K(x_1,s)\ \omega_1^R(s)ds$$

$$+\tilde{u}(x_2)\int_{x_0}^{x_1} K(x_1,s)\ \omega_2^R(s)ds = f(x_1),$$

$$\tilde{u}(x_2) + \tilde{u}(x_0)\int_{x_0}^{x_1} K(x_2,s)\ \omega_0^R(s)ds +$$

$$+\tilde{u}(x_1)\int_{x_0}^{x_1} K(x_2,s)\ \omega_1^R(s)ds$$

$$+\tilde{u}(x_2)\int_{x_0}^{x_1} K(x_2,s)\ \omega_2^R(s)ds$$

$$+\tilde{u}(x_0)\int_{x_1}^{x_2} K(x_2,s)\ \omega_0^L(s)ds$$

$$+\tilde{u}(x_1)\int_{x_1}^{x_1} K(x_2,s)\ \omega_1^L(s)ds$$

$$+\tilde{u}(x_2)\int_{x_1}^{x_2} K(x_2,s)\ \omega_2^L(s)ds = f(x_2).$$

The advantages of the proposed method include the ability to calculate the exact integral $\int_{x_i}^{x_{i+1}} K(x,s)\ \omega_0^R(s)ds$ (without error). However, in case of difficulties with calculating the integral, we can apply a quadrature formula that provides the order of approximation $m, m \geq 3$.

3 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 7 from paper [8]:

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

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

Build a grid of equally spaced nodes with a step h. We will solve by different methods: using polynomial and non-polynomial splines, as well as the trapezoidal method.

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 the linear polynomial splines (column 2). The numerical solution obtained with the application of the non-polynomial splines $(\varphi_1(x) = 1 \quad \varphi_2(x) = \sin(x), \quad \varphi_3(x) = \cos(x))$ is given in the third column. The numerical solution obtained with the application of the non-polynomial splines $(\varphi_1(x) = 1, \quad \varphi_2(x) = \exp(x), \quad \varphi_3(x) = \exp(2x))$ is presented in the fourth column of Table 1. The number of nodes (n) is given in the first column of Table 4. The application of the quadrature formula of trapeziums for this equation gives us the exact solution (see Fig.6).

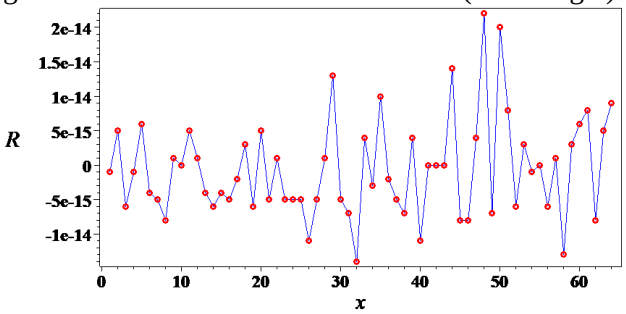

Fig.6. The plot of error between the exact solution of the equation (problem 1) and the numerical solution obtained with the application of the quadrature formula of trapeziums when we use 64 nodes

As it is known, the composite trapezoidal rule for calculating an integral has the following form:

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

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

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

We have the equation

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

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

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

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

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

$$+f(x_k), \qquad k = 2,\dots,n.$$

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:

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

Calculating sequentially according to this scheme, we get $\tilde{u}(x_0) = \exp(-x_0) + x_0\exp(x_0)$,

For $x_0 = 0$, we have $\tilde{u}(x_0) = 1$. Further, for $x_1 = h$, we obtain

$$\tilde{u}(x_1) = \frac{\exp(-h)+h\exp(h)/2}{1+h\exp(2h)/2} = \exp(-h),$$

Continuing to calculate, we obtain $\tilde{u}(x_k) = exp(-x_k)$, which coincides with the exact solution. Applying the quadrature trapezoidal formula to other integral equations, such a good result can hardly be expected.

When we use the linear piecewise polynomial splines with the 64 nodes on the interval [0,1] we receive the plot of the error $R = u - \tilde{u}$ between the exact solution of the equation and the numerical solution obtained with the application the

polynomial splines of the third order approximation which is shown in Fig.7. When we use the non-polynomial splines of the third order approximation ($\varphi_1(x) = 1,\ \varphi_2(x) = \exp(x),\ \varphi_3(x) = \exp(2x)$) 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 which is shown in Fig.8.

When we use the non-polynomial splines of the third order approximation ($\varphi_1(x) = 1,\ \varphi_2(x) = \exp(-x), \varphi_3(x) = \exp(-2x)$) 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 which is shown in Fig.9. Here we connected the points with straight line segments for clarity of the drawing. The graph shows the nodes of the grid at the interval [0,1] and the values of the errors R at these nodes.

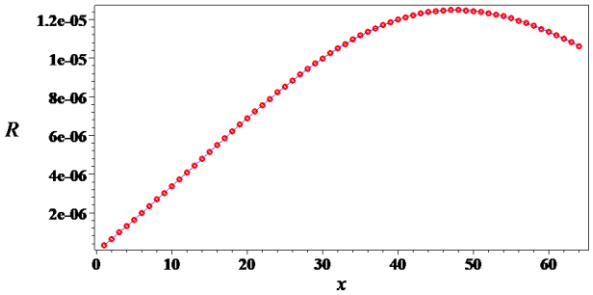

Fig.7. The plot of error between the exact solution of the equation (problem 1) and the numerical solution obtained with the application of the linear polynomial splines when we use 64 nodes

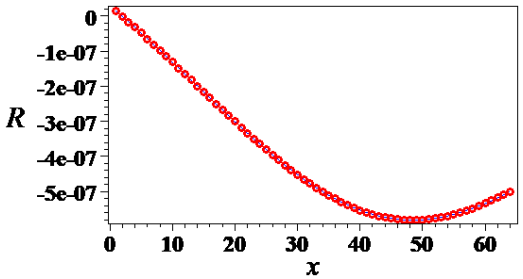

Fig.8. The plot of error between the exact solution of the equation (problem 1) and the numerical solution obtained with the application of the non-polynomial exponential splines of the third order approximation ($\varphi_1(x) = 1, \varphi_2(x) = \exp(x), \varphi_3(x) = \exp(2x)$) when we use 64 nodes

When we use the non-polynomial splines ($\varphi(x) = \sin(x)$) with the 128 nodes on the interval [0,1] we receive the plot of the error which is shown in Fig.10. When we use the polynomial splines of the third order approximation with the 64 nodes on the interval [0,1] we receive the plot of the error which is shown in Fig.11. When we use the polynomial splines of the third order approximation with the 128 nodes on the interval [0,1] we receive the plot of the error which is shown in Fig.12.

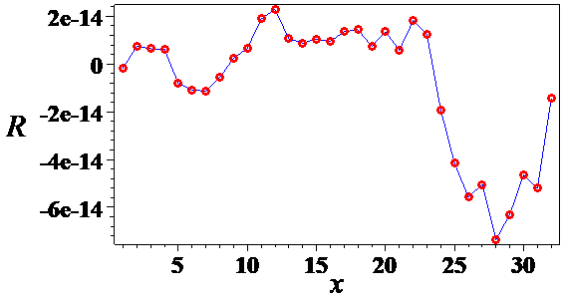

Fig.9. The plot of error between the exact solution of the equation (problem 1) and the numerical solution obtained with the application of the non-polynomial splines of the third order approximation ($\varphi_1(x) = 1, \varphi_2(x) = \exp(-x), \varphi_3(x) = \exp(-2x)$) when we use 32 nodes

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 polynomial splines, and the non-polynomial splines of the third order approximation

n	The error obtained with the use the polynomial splines	The error obtained with the use the trigonometric splines	The error obtained with the use the non-polynomial splines $\varphi_1(x) = 1, \varphi_2(x) = \exp(x), \varphi_3(x) = \exp(2x))$
32	$0.396 \cdot 10^{-7}$	$0.792 \cdot 10^{-7}$	$0.235 \cdot 10^{-6}$
64	$0.247 \cdot 10^{-8}$	$0.496 \cdot 10^{-8}$	$0.148 \cdot 10^{-7}$
128	$0.163 \cdot 10^{-9}$	$0.310 \cdot 10^{-9}$	$0.933 \cdot 10^{-9}$

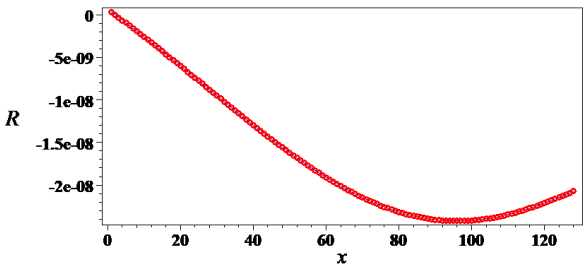

Fig.10. The plot of error between the exact solution of the equation (problem 1) and the numerical solution obtained with the application of the non-polynomial splines ($\varphi_1(x) = 1\ \varphi_2(x) = \sin(x), \varphi_3(x) = \cos(x)$) when we use 128 nodes

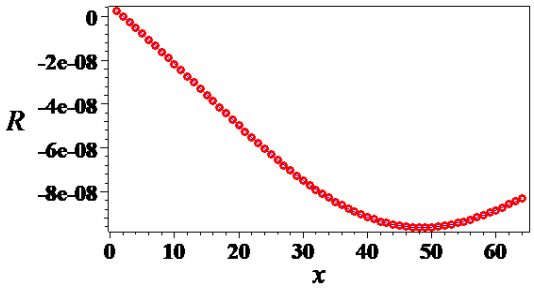

Fig.11. The plot of error between the exact solution of the equation (problem 1) and the numerical solution obtained with the application of the polynomial splines of the third order approximation when we use 64 nodes.

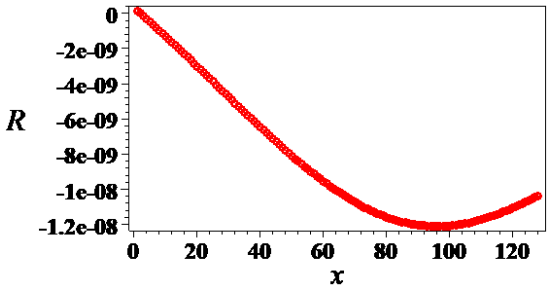

Fig.12. The plot of error between the exact solution of the equation (problem 1) and the numerical solution obtained with the application of the polynomial splines of the third order approximation when we use 128 nodes.

Note that to calculate the integral on the interval $[x_j, x_{j+1}]$, we can use a quadrature formula, for example the formula of the middle rectangles. We have in this case: $h = x_{j+1} - x_j$,

$$V_j(x) \approx \tilde{u}(x_{j-1}) hK\left(x, x_j + \frac{h}{2}\right) \omega_{j-1}^L\left(x_j + \frac{h}{2}\right)$$

$$+\tilde{u}(x_j) hK(x, x_j + h/2)\, \omega_j^L(x_j + h/2) +$$

$$\tilde{u}(x_{j+1}) hK(x, x_j + h/2)\, \omega_{j+1}^L(x_j + h/2),$$

$$Q_j(x) \approx \tilde{u}(x_j) hK\left(x, x_j + \frac{h}{2}\right) \omega_j^R\left(x_j + \frac{h}{2}\right)$$

$$+\tilde{u}(x_{j+1}) hK\left(x, x_j + \frac{h}{2}\right) \omega_{j+1}^R\left(x_j + \frac{h}{2}\right)$$

$$+\tilde{u}(x_{j+2}) hK\left(x, x_j + \frac{h}{2}\right) \omega_{j+2}^R\left(x_j + \frac{h}{2}\right).$$

Fig.13 shows the plot of error between the exact solution of the equation (problem 1) and the numerical solution obtained with the application of the polynomial splines of the third order approximation and the middle rectangles formula when we use 128 nodes.

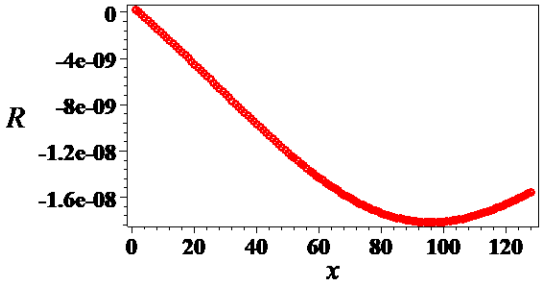

Fig.13. The plot of error between the exact solution of the equation (problem 1) and the numerical solution obtained with the application of the polynomial splines of the third order approximation when we use 128 nodes.

3 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 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:

$$U(x) = u(x_j)\omega_j(x) + u(x_{j+1})\omega_{j+1}(x), \quad x \in [x_j, x_{j+1}], \tag{5}$$

Now transforming the integral $\int_{x_j}^{x_{j+1}} K(x, s, u(s))ds$ using formula (5), we obtain

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

Using formulas (3), (4) we obtain the approximate solution with the lesser error.

Here we take the nonlinear Volterra 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 equation from paper [8] and compare the results with the result obtained with splines. First, in problem 4 we consider problem 6 from paper [8].

Problem 2. We take the equation

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

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

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

At first, we build a grid of nodes on the interval [0,1]:

$$x_k = kh, k = 0,1, \dots n, h = \frac{1}{n}.$$

To solve the problem, we need the following expressions:

$$W^k(x) = \int_{x_k}^{x_{k+1}} K(x,s)(u_k\omega_k(s) + u_{k+1}\omega_{k+1}(s) + u_{k+2}\omega_{k+2}(s))^2\, ds;$$

$$W_1^k(x) = \int_{x_k}^{x_{k+1}} K(x,s)(u_{k-1}\omega_{k-1}(s) + u_k\omega_k(s) + u_{k+1}\omega_{k+1}(s))^2\, ds;$$

Hereinafter $u_k \approx u(x_k)$. Then we sequentially calculate u_k. We have $u_0 = \tilde{u}(x_0)$: $\tilde{u}(x_0) = f(x_0)$,

To determine u_1 u_2, we solve the system of equations $V_1 = 0, V_2 = 0$:

$$V_1 = u_1 - W^0(x_1) - f(x_1),$$
$$V_2 = u_2 - W^0(x_2) - W_1^1(x_2) - f(x_2),$$

Next, we sequentially find u_m by solving the equations:

$$u_3 - \sum_{j=1}^{2} W_1^j(x_3) - W^0(x_3) - f(x_3) = 0,$$
$$u_m - \sum_{j=1}^{m-1} W_1^j(x_m) - W^0(x_m) - f(x_m) = 0.$$

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 polynomial splines of the third order approximation which is shown in Fig.14. 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 third order approximation polynomial splines which is shown in Fig.15.

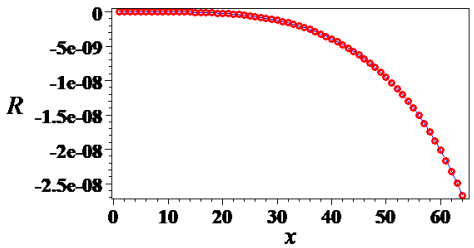

Fig.14. The plot of error between the exact solution of the equation (problem 2) and the numerical solution obtained with the application of the polynomial splines of the third order approximation when we use 64 nodes

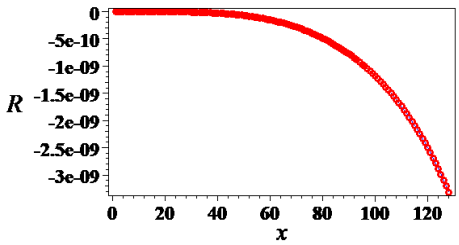

Fig.15. The plot of error between the exact solution of the equation (problem 2) and the numerical solution obtained with the application of the polynomial splines of the third order approximation when we use 128 nodes

Table 5 shows the maximum of the error in absolute value between the exact solution of the equation (*Problem 2)* and the numerical solution obtained with the application of the linear polynomial splines of the second order approximation (column 2). The numerical solution obtained with the application the polynomial splines of the third order approximation ($\varphi(x) = sin(x)$) is presented in column 3. The numerical solution obtained with the application the non-polynomial splines of the second order approximation ($\varphi(x) = exp(-x)$) is given in column 4. 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 2 and the numerical solution obtained with the application of the linear polynomial splines, non-polynomial splines

n	The error obtained with the use the linear polynomial splines of the second order approximation	The error obtained with the use the polynomial splines of the third order approximation	The error obtained with the use the non-polynomial splines of the second order approximation $\varphi(x) = exp(-x)$
32	$0.116 \cdot 10^{-4}$	$0.216 \cdot 10^{-6}$	$0.856 \cdot 10^{-5}$
64	$0.290 \cdot 10^{-5}$	$0.267 \cdot 10^{-7}$	$0.214 \cdot 10^{-5}$
128	$0.724 \cdot 10^{-6}$	$0.332 \cdot 10^{-8}$	$0.536 \cdot 10^{-6}$

When we use the non-polynomial splines of the second order approximation $\varphi(x) = \exp(-x)$ with the 64 nodes on the interval [0,1] we receive the plot of the error which is shown in Fig.16. When we use the non-polynomial splines of the second order approximation $\varphi(x) = \exp(-x)$ splines with the 128 nodes on the interval [0,1] we receive the plot of the error which is shown in Fig.17.

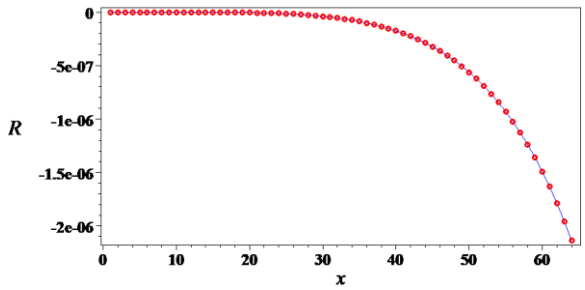

Fig.16. The plot of error between the exact solution of the equation (problem 2) and the numerical solution obtained with the application of the polynomial splines of the second order approximation when we use 64 nodes

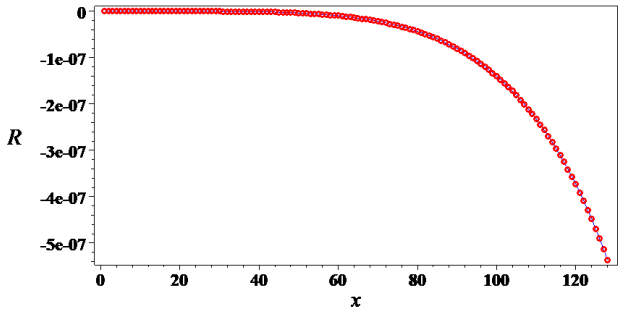

Fig.17. The plot of error between the exact solution of the equation (problem 2) and the numerical solution obtained with the application of the polynomial splines of the second order approximation when we use 128 nodes

4 Conclusion

In this paper, polynomial and non-polynomial splines of the third 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 third 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 third order of approximation for the Volterra integral equations of the second kind can give an appropriate result, better than the local interpolation splines of the second order of approximation. But it is quite possible that the splines of the second order of the approximation order will give a smaller error (for example, if the absolute value of the maximum of the second derivative of the function turns out to be less than the absolute value of the maximum of the third derivative). The use of various methods allows us to verify the result. The non-polynomial spline can sometimes provide the result better than the polynomial splines.

References:

[1] J.Xie, M.Yi, Numerical Research of Nonlinear System of Fractional Volterra-Fredholm Integral-Differential Equations via Block-Pulse Functions and Error Analysis, *Journal of Computational and Applied Mathematics*, Vol.345, 2019, pp. 159-167.

[2] K.Maleknejad, J.Rashidinia, T.Eftekhari, Numerical Solution of Three-Dimensional Volterra-Fredholm Integral Equations of the First and Second Kinds Based on Bernstein's Approximation, *Applied Mathematics and Computation*, Vol.339, 2018, pp. 272-285.

[3] N.Negarchi, K.Nouri, Numerical Solution of Volterra-Fredholm Integral Equations Using the Collocation Method Based on a Special Form of the Mëuntz-Legendre Polynomials, *Journal of Computational and Applied Mathematics*, Vol.344, 2018, pp. 15-24.

[4] 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.

[5] R.I. Esa, A.J. Saleh, Numerical treatment of first order Volterra integro-differential equation using non-polynomial spline functions, *Iraqi Journal of Science*, 2020, pp. 114-121.

[6] M.N.Sahlan, Convergence of approximate solution of mixed Hammerstein type integral equations, *Boletim da Sociedade Paranaense de Matematica*, Vol.38, No 2, 2020, pp. 61-74.

[7] K.Maleknejad, J.Rashidinia, H. Jalilian, Non-polynomial spline functions and quasi-linearization to approximate nonlinear volterra integral equation, *Filomat*, Vol.32, No 11, 2018, pp. 3947-3956.

[8] 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.

[9] I. G. Burova, N. S. Domnin, A. E. Vezhlev, A. V. Lebedeva, A. N. Pakulina, Nonlinear Fredholm Integral Equation of the Second Kind with Quadrature Methods, *Journal of WSEAS transactions on mathematics*, Vol. 17, 2018

[10] I.G. Burova, G.O.Alcybeev, Application of Splines of the second order approximation to Volterra integral equations of the second kind. Applications in Systems, Theory and Dynamical Systems, *International Journal of Circuits, Systems and Signal Processing*, Vol. 15, 2021, pp. 63-71.

[11] I.G.Burova, On Left Integro-Differential Splines and Cauchy Problem, *International Journal of Mathematical Models and Methods in Applied Sciences*, Vol.9, 2015, pp. 683-690.

[12] I.G.Burova, T.O.Evdokimova, On Construction Third Order Approximation Using Values of Integrals, *WSEAS Transactions on Mathematics*, Vol.13, 2014, pp. 676-683.

[13] Yuri K. Demjanovich, On Complexity of Adaptive Splines, *International Journal of Circuits, Systems and Signal Processing*, Vol. 14, 2020, pp. 607-615.

[14] Yuri K. Demjanovich, Tatjana A.Safonova, Mikhail A.Terekhov, The generalized Haar spaces and their adaptive decomposition, *International Journal of Circuits, Systems and Signal Processing*, Vol. 14, 2020, pp. 548-560.

[15] 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.

[16] 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, 2019, pp. 104-111.

[17] Galina Mehdiyeva, Vagif Ibrahimov, Mehrib Imanova, Application of the Finite Differences Methods to Computation of Definite Integrals, WSEAS Transactions on System, Vol. 19, 2020, pp 1-6.

[18] 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.

[19] I.G.Burova, V.M.Ryabov, On the Solution of Fredholm Integral Equations of the First Kind, *WSEAS Transactions on Mathematics*, Vol.19, 2020, pp. 699-708.