Specifics of using Local Approximations to Solve Fredholm Integral Equations of the Second Kind

I. G. BUROVA, N. A. LEBEDINSKAYA, D. M. LEBEDINSKI, Y. A. PATSUKEVICH
Department of Computational Mathematics,
Department of Parallel Algorithms,
St. Petersburg State University,
7/9 Universitetskaya Embankment, St.Petersburg, 199034,
RUSSIA

Abstract: - This paper examines the numerical solution of a Fredholm integral equation of the second kind using local polynomial splines with an approximation of the ninth order. In this case, the support of the basis splines has eight adjacent grid intervals. Different variants of local spline approximation are employed for the initial, middle, and final segments of the integration interval. The construction of computational schemes for solving the Fredholm integral equation of the second kind using these splines is discussed. Results of numerical experiments on function approximation and the solution of Fredholm integral equations are presented and compared with the Newton-Cotes rules and the results from other papers.

Key-Words: – Local splines, approximation, interpolation, Fredholm integral equation of the second kind.

Received: May 19, 2026. Revised: July 21, 2026. Accepted: August 23, 2026. Published: November 24, 2026.

1 Introduction

Interest in constructing high-precision spline ap proximations and applying them to the solution of integral equations remains high. We note the following papers.

Paper [1] gives a survey of the use of Chebyshev polynomials in the solution of integral equations.

The collocation methods based on cubic B-splines are developed in [2] to construct numerical solutions to Fredholm integral equations of the first and second kind.

In [3], a numerical method for solving nonlinear Fredholm-Hammerstein integral equations by using a multiquadric quasi-interpolation method is developed.

Paper [4] presents an innovative, semi-analytical approach for solving linear Fredholm integral equations (FIEs) of the first and second kind.

In paper [5], the Bernstein Piecewise Polynomials Method (BPPM), Integral Mean Value Method (IMVM), and other methods such as the Taylor Series Method (TSM), and the Least Square Method (LSM) are used to solve integral equations of the second kind numerically.

In paper [6], a new spline method, based on quintic spline functions is developed for solving Fredholm integral equations of the second kind.

In paper [7], optimal quadrature formulas with derivatives in the space of differentiable functions using the Sobolev method is used for solving Fredholm integral equations of the second kind.

In paper [8], new spline quasi-interpolants for the sharp approximation of data in one and several dimensions is introduced.

In paper [9], a method based on the approximation of the integral operator by quasi-interpolating the density function using Gaussian kernels is suggested.

Paper [10] discusses the solving of the integral equation with using the constructed optimal quadrature formula and trapezoidal rule.

Paper [11] considers the numerical solution of the Fredholm integral equation of the second kind using local polynomial splines of the fourth and fifth order of approximation.

Currently, machine learning is frequently applied to solve a wide variety of tasks, some of which are explored in the following. Study [12] evaluated the influential parameters in predicting earthquake-induced slope displacements by developing a multivariate adaptive regression splines (MARS) model — a machine learning algorithm — based on the results of two-dimensional dynamic nonlinear finite element simulations available in the literature.

In [13], a solution to the problem of end-to-end learning for raw waveform signals by introducing learnable continuous time-frequency atoms is proposed.

Paper [14] compares machine-learning methods and cubic splines on the sparsity of training data they can handle, especially when training samples are noisy. It is shown that, given very sparse data, cubic splines constitute a more precise interpolation method than either deep neural networks or multivariate adaptive regression splines.

Paper [15] explores some of the shortcomings of the current machine learning framework and reasons for the gap between approximation theory and machine learning practice.

In [16], splines were used to solve a Fredholm integral equation.

Artificial intelligence is, first and foremost, a computer program that can be adapted for data processing and pattern recognition. When solving integral equations, artificial intelligence can help characterize the kernel and the right-hand side of the equation including identifying singularities, intervals of smoothness, and a priori properties of the solution. Subsequently, appropriate tools can be selected to solve the problem. In the case of spline approximations, suitable basis functions can be chosen. Once the framework of the approximate solution has been computed using spline approximations, an approximate solution with the required degree of smoothness can be constructed.

This paper proposes a method for solving Fredholm integral equations of the second kind based on ninth-order spline approximations.

2 Ninth-order Approximation Spline Construction

2.1 Ninth-order Approximations at the Beginning of the Interpolation Interval

Let a and b be real. Note that a and b can be either finite or infinite numbers.

Let the values f_j of a certain continuous function $f(x)$ at the ordered grid nodes $\{x_j\}$ on the interval $[a,b]$:

$$a = x_0 < x_1 < \cdots < x_n = b$$

be given.

The authors of this paper have previously examined the construction of local polynomial and non-polynomial splines - both of zero and non-zero level - with approximation orders up to the seventh degree. A key advantage of local splines is that the function approximation is constructed independently on each grid interval $[x_j, x_{j+1}]$ as a sum of products involving function values at grid nodes and the basis functions.

It should also be noted that approximating a function over a finite interval requires the use of boundary-minimal splines. We define boundary-minimal interpolation splines as those that utilize function values only at the grid nodes within the interval $[a,b]$. In this context, we can employ so-called left-sided, right-sided, and centered boundary-minimal splines. Consequently, adjacent grid intervals may utilize approximations of varying orders and properties; for instance, adjacent intervals might employ splines that are exact on algebraic polynomials alongside those that are exact on trigonometric polynomials.

We separately consider spline approximations at the beginning, middle, and near the end of the approximation interval $[a,b]$. Note that, with the same order of approximation, these splines have different constants in the error estimate.

By increasing the approximation order of the splines that are applied near the boundaries of the interpolation interval $[a,b]$, we can balance the approximation errors across the entire interval $[a,b]$.

We consider the approximation $F(x)$ with continuous splines of the ninth order of accuracy at the beginning of the of the interval $[a,b]$ on each grid interval $[x_j, x_{j+1}]$. These basis splines in the polynomial case are obtained by solving a system of relations (approximation identities):

$$F(x) = f(x), \qquad f(x) = 1, x, \ldots., x^8.$$

We consider approximations with continuous splines of the ninth order on each grid interval $[x_j, x_{j+1}]$, $j = 0, 2, 3, 4, 5, 6, 7, 8$, in the form:

$$F(x) = \sum_{k=j}^{j+8} f(x_k)\, w_k(x)\,, x \in [x_j, x_{j+1}],$$

where the ninth-order approximation basis splines can be defined by the expressions:

$$w_j(x) = \prod_{i=1}^{8} \frac{x - x_{j+i}}{x_j - x_{j+i}}, x \in [x_j, x_{j+1}],$$

$$w_{j+1}(x) = \frac{x - x_j}{x_{j+1} - x_j} \prod_{i=2}^{8} \frac{x - x_{j+i}}{x_{j+1} - x_{j+i}},$$
$$x \in [x_j, x_{j+1}],$$

$$w_{j+2}(x) = \frac{(x - x_j)(x - x_{j+1})}{(x_{j+2} - x_j)(x_{j+2} - x_{j+1})} \prod_{i=3}^{8} \frac{x - x_{j+i}}{x_{j+2} - x_{j+i}},$$
$$x \in [x_j, x_{j+1}],$$

$$w_{j+3}(x) = \prod_{i=0}^{2} \frac{x - x_{j+i}}{x_{j+3} - x_{j+i}} \prod_{i=4}^{8} \frac{x - x_{j+i}}{x_{j+3} - x_{j+i}},$$
$$x \in [x_j, x_{j+1}],$$

$$w_{j+4}(x)=\prod_{i=0}^{3}\frac{x-x_{j+i}}{x_{j+4}-x_{j+i}}\ \prod_{i=5}^{8}\frac{x-x_{j+i}}{x_{j+4}-x_{j+i}},\quad x\in[x_j,x_{j+1}],$$

$$w_{j+5}(x)=\prod_{i=0}^{4}\frac{x-x_{j+i}}{x_{j+5}-x_{j+i}}\ \prod_{i=6}^{8}\frac{x-x_{j+i}}{x_{j+5}-x_{j+i}},\quad x\in[x_j,x_{j+1}],$$

$$w_{j+6}(x)=\prod_{i=0}^{5}\frac{x-x_{j+i}}{x_{j+6}-x_{j+i}}\ \prod_{i=7}^{8}\frac{x-x_{j+i}}{x_{j+6}-x_{j+i}},\quad x\in[x_j,x_{j+1}],$$

$$w_{j+7}(x)=\prod_{i=0}^{6}\frac{x-x_{j+i}}{x_{j+7}-x_{j+i}}\ \prod_{i=8}^{8}\frac{x-x_{j+i}}{x_{j+7}-x_{j+i}},\quad x\in[x_j,x_{j+1}],$$

$$w_{j+8}(x)=\prod_{i=0}^{7}\frac{x-x_{j+i}}{x_{j+8}-x_{j+i}},x\in[x_j,x_{j+1}].$$

We call these basis splines (and the spline approximation with them) as the left-side basis splines (and the related spline approximation) as they are considered only near the left boundary of the interval $[a,b]$. Note that supp $w_j=[x_{j-8},\ x_{j+1}]$.

2.1.1 The Error of Approximation

The error of approximation on interval $[x_j,x_{j+1}]$ can be given with the relation:

$$|f(x)-F(x)|\le K\,h^9\parallel f^{(9)}\parallel_{[x_j,x_{j+8}]},$$

where $K=4930/9!\approx 0.0136$.

Proof: This error estimate is easily derived using classical Lagrange interpolation theory [11].

2.2 Approximation Error with the Middle-Sided Splines

We consider the approximation using continuous splines of the ninth order of approximation on each grid interval $[x_j,x_{j+1}]$ (where $j=-4,-3,-2,-1,0,1,2,3,4$) separately, in the form:

$$F(x)=\sum_{k=j-4}^{j+4}f(x_k)\,w_k(x),$$

where the ninth-order approximation basis splines can be defined by the expressions:

$$w_{j-4}(x)=\prod_{i=-3}^{4}\frac{x-x_{j+i}}{x_{j-4}-x_{j+i}},x\in[x_j,x_{j+1}],$$

$$w_{j-3}(x)=\frac{x-x_{j-4}}{x_{j-3}-x_{j-4}}\prod_{i=-2}^{4}\frac{x-x_{j+i}}{x_{j-3}-x_{j+i}},\quad x\in[x_j,x_{j+1}],$$

$$w_{j-2}(x)=\prod_{i=-4}^{-3}\frac{x-x_{j+i}}{x_{j-2}-x_{j+i}}\ \prod_{i=-1}^{4}\frac{x-x_{j+i}}{x_{j-2}-x_{j+i}},\quad x\in[x_j,x_{j+1}],$$

$$w_{j-1}(x)=\prod_{i=-4}^{-2}\frac{x-x_{j+i}}{x_{j-1}-x_{j+i}}\ \prod_{i=0}^{4}\frac{x-x_{j+i}}{x_{j-1}-x_{j+i}},\quad x\in[x_j,x_{j+1}],$$

$$w_{j}(x)=\prod_{i=-4}^{-1}\frac{x-x_{j+i}}{x_{j}-x_{j+i}}\ \prod_{i=1}^{4}\frac{x-x_{j+i}}{x_{j}-x_{j+i}},\quad x\in[x_j,x_{j+1}],$$

$$w_{j+1}(x)=\prod_{i=-4}^{0}\frac{x-x_{j+i}}{x_{j+1}-x_{j+i}}\ \prod_{i=2}^{4}\frac{x-x_{j+i}}{x_{j+1}-x_{j+i}},\quad x\in[x_j,x_{j+1}],$$

$$w_{j+2}(x)=\prod_{i=-4}^{1}\frac{x-x_{j+i}}{x_{j+2}-x_{j+i}}\ \prod_{i=3}^{4}\frac{x-x_{j+i}}{x_{j+2}-x_{j+i}},\quad x\in[x_j,x_{j+1}],$$

$$w_{j+3}(x)=\prod_{i=-4}^{2}\frac{x-x_{j+i}}{x_{j+3}-x_{j+i}}\ \prod_{i=4}^{4}\frac{x-x_{j+i}}{x_{j+3}-x_{j+i}},\quad x\in[x_j,x_{j+1}],$$

$$w_{j+4}(x)=\prod_{i=-4}^{3}\frac{x-x_{j+i}}{x_{j+4}-x_{j+i}},x\in[x_j,x_{j+1}].$$

On each grid interval $[x_j,x_{j+1}]$, assuming a uniform grid with step h, we obtain the following error estimate, [11]:

$$|f(x)-F(x)|\le K\,h^9\parallel f^{(9)}\parallel_{[x_{j-4},x_{j+4}]},$$

where $K_1=195/9!\approx 0.000537$.

2.3 Ninth-order Approximations near the End of the Interpolation Interval

Approximation by continuous splines of the ninth order of approximation is considered at the end of the approximation interval on each grid interval $[x_j,x_{j+1}]$, $j=n-7,n-6,n-5,n-4,n-3,n-2,n-1$ separately in the form:

$$F(x)=\sum_{k=j-7}^{j+1}f(x_k)\,w_k(x),x\in\ [x_j,x_{j+1}],$$

where:

$$w_{j-7}(x)=\prod_{i=-6}^{1}\frac{x-x_{j+i}}{x_{j-7}-x_{j+i}},x\in[x_j,x_{j+1}],$$

$$w_{j-6}(x)=\frac{x-x_{j-7}}{x_{j-6}-x_{j-7}}\prod_{i=-5}^{1}\frac{x-x_{j+i}}{x_{j-6}-x_{j+i}},\quad x\in[x_j,x_{j+1}],$$

$$w_{j-5}(x)=\frac{x-x_{j-7}}{x_{j-5}-x_{j-7}}\frac{x-x_{j-6}}{x_{j-5}-x_{j-6}}\prod_{i=-4}^{1}\frac{x-x_{j+i}}{x_{j-5}-x_{j+i}},\quad x\in[x_j,x_{j+1}],$$

$$w_{j-4}(x)=\prod_{i=-7}^{-5}\frac{x-x_{j+i}}{x_{j-4}-x_{j+i}}\prod_{i=-3}^{1}\frac{x-x_{j+i}}{x_{j-4}-x_{j+i}},\quad x\in[x_j,x_{j+1}],$$

$$w_{j-3}(x)=\prod_{i=-7}^{-4}\frac{x-x_{j+i}}{x_{j-3}-x_{j+i}}\prod_{i=-2}^{1}\frac{x-x_{j+i}}{x_{j-3}-x_{j+i}},\quad x\in[x_j,x_{j+1}],$$

$$w_{j-2}(x)=\prod_{i=-7}^{-3}\frac{x-x_{j+i}}{x_{j-2}-x_{j+i}}\prod_{i=-1}^{1}\frac{x-x_{j+i}}{x_{j-2}-x_{j+i}},\quad x\in[x_j,x_{j+1}]$$

$$w_{j-1}(x)=\prod_{i=-7}^{-2}\frac{x-x_{j+i}}{x_{j-1}-x_{j+i}}\prod_{i=0}^{1}\frac{x-x_{j+i}}{x_{j-1}-x_{j+i}},\quad x\in[x_j,x_{j+1}]$$

$$w_j(x)=\prod_{i=-7}^{-1}\frac{x-x_{j+i}}{x_j-x_{j+i}}\prod_{i=1}^{1}\frac{x-x_{j+i}}{x_j-x_{j+i}},\quad x\in[x_j,x_{j+1}].$$

$$w_{j+1}(x)=\prod_{i=-7}^{0}\frac{x-x_{j+i}}{x_{j+1}-x_{j+i}},\qquad x\in[x_j,x_{j+1}].$$

On each grid interval $[x_j,x_{j+1}]$, assuming a uniform grid with step size h, we obtain the following error estimate (see [11]):

$$|f(x)-F(x)|\le K\,h^9\parallel f^{(9)}\parallel_{[x_{j-7},x_{j+1}]},$$

where K_2=4930/9! ≈ 0.0136.

It should be noted that the ratio of the constants in the approximation error estimates is $K_2/K_1\approx 25$.

2.4 Trigonometric Splines Application

The use of trigonometric splines is of particular interest. They are derived using a method analogous to that used for polynomials, but they satisfy an accuracy condition for trigonometric functions. Below, we provide the formulas for trigonometric splines used for midpoint approximations. We consider the approximation using continuous trigonometric splines of the ninth order of approximation on each grid interval $[x_j,x_{j+1}]$ (where j = -4, -3, -2, -1, 0, 1, 2, 3, 4) separately, in the form:

$$F(x)=\sum_{k=j-4}^{j+4}f(x_k)\,w_k(x),$$

where the ninth-order approximation basis splines can be defined by the expressions:

$$w_{j-4}(x)=\prod_{i=-3}^{4}\frac{\sin\left(\frac{x}{2}-\frac{x_{j+i}}{2}\right)}{\sin(x_{j-4}/2-x_{j+i}/2)},\quad x\in[x_j,x_{j+1}],$$

$$w_{j-3}(x)=\frac{\sin(x/2-x_{j-4}/2)}{\sin(x_{j-3}/2-x_{j-4}/2)}\prod_{i=-2}^{4}\frac{\sin(x/2-x_{j+i}/2)}{\sin(x_{j-3}/2-x_{j+i}/2)},\quad x\in[x_j,x_{j+1}],$$

$$w_{j-2}(x)=\prod_{i=-4}^{-3}\frac{\sin(x/2-x_{j+i}/2)}{\sin(x_{j-2}/2-x_{j+i}/2)}\prod_{i=-1}^{4}\frac{\sin(x/2-x_{j+i}/2)}{\sin(x_{j-2}/2-x_{j+i}/2)},\quad x\in[x_j,x_{j+1}],$$

$$w_{j-1}(x)=\prod_{i=-4}^{-2}\frac{\sin(x/2-x_{j+i}/2)}{\sin(x_{j-1}/2-x_{j+i}/2)}\prod_{i=0}^{4}\frac{\sin(x/2-x_{j+i}/2)}{\sin(x_{j-1}/2-x_{j+i}/2)},\quad x\in[x_j,x_{j+1}],$$

$$w_j(x)=\prod_{i=-4}^{-1}\frac{\sin(x/2-x_{j+i}/2)}{\sin(x_j/2-x_{j+i}/2)}\prod_{i=1}^{4}\frac{\sin(x/2-x_{j+i}/2)}{\sin(x_j/2-x_{j+i}/2)},\quad x\in[x_j,x_{j+1}],$$

$$w_{j+1}(x)=\prod_{i=-4}^{0}\frac{\sin(x/2-x_{j+i}/2)}{sin(x_{j+1}/2-x_{j+i}/2)}\prod_{i=2}^{4}\frac{\sin(x/2-x_{j+i}/2)}{\sin(x_{j+1}/2-x_{j+i}/2)},\quad x\in[x_j,x_{j+1}],$$

$$w_{j+2}(x)=\prod_{i=-4}^{1}\frac{\sin(x/2-x_{j+i}/2)}{\sin(x_{j+2}/2-x_{j+i}/2)}\prod_{i=3}^{4}\frac{\sin(x/2-x_{j+i}/2)}{\sin(x_{j+2}/2-x_{j+i}/2)},\quad x\in[x_j,x_{j+1}],$$

$$w_{j+3}(x)=\prod_{i=-4}^{2}\frac{\sin(x/2-x_{j+i}/2)}{\sin(x_{j+3}/2-x_{j+i}/2)}\prod_{i=4}^{4}\frac{\sin(x/2-x_{j+i}/2)}{\sin(x_{j+3}/2-x_{j+i}/2)},\quad x\in[x_j,x_{j+1}],$$

$$w_{j+4}(x)=\prod_{i=-4}^{3}\frac{\sin\left(\frac{x}{2}-\frac{x_{j+i}}{2}\right)}{\sin\left(\frac{x_{j+4}}{2}-\frac{x_{j+i}}{2}\right)},\ x\in[x_j,x_{j+1}].$$

On each grid interval $[x_j,x_{j+1}]$, assuming a uniform grid with step h, we obtain the following error estimate (see [11]): on a uniform grid, the approximation error for these splines is O(h^9).

2.5 Numerical Experiments

Figure 1, Figure 2, and Figure 3 show the approximation errors for the function $\exp(-x^2)$ on the interval $[-1,1]$. Figure 1 shows the plot of the approximation error for the function $\exp(-x^2)$ with equidistant grid nodes and $n=11$. Figure 2 shows the plot of the approximation error for the function $\exp(-x^2)$ with equidistant grid nodes

($h = 2/n$ and $n = 17$). Figure 3 shows the plot of the approximation error for the function $\exp(-x^2)$ with non-equidistant grid nodes.

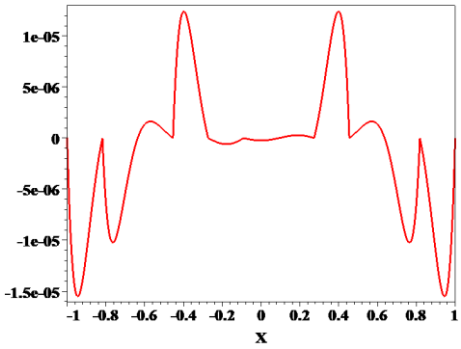

Fig. 1: Plot of the approximation error for the function $\exp(-x^2)$ with equidistant grid nodes and $n = 11$.
Source: created by the authors.

The approximation error plot for the function $\exp(-x^2)$ with equidistant grid nodes and $n = 17$ is shown in Figure 2. For the fifth grid interval, the error obtained when using left-sided splines, as well as midpoint-based splines, is presented.

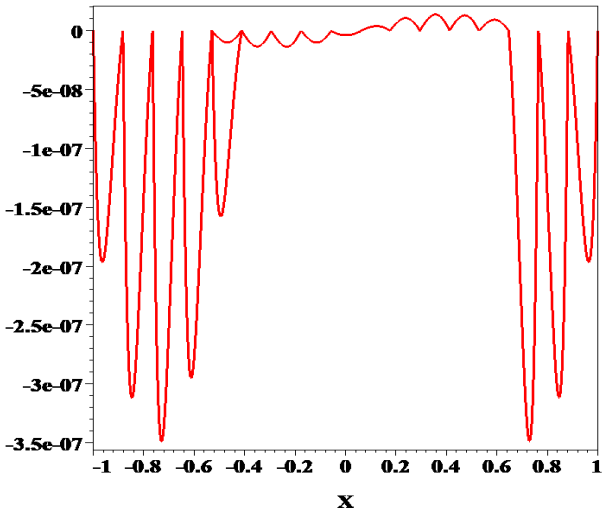

Fig. 2. Plot of the approximation error for the function $\exp(-x^2)$ with equidistant grid nodes ($h = 2/n$ and $n = 17$).
Source: created by the authors.

Our spline approximation method allows for the use of a non-uniform grid. Let $n = 18, n_1 = 24, h_1 = \frac{2}{n_1}$

The grid nodes can be constructed as follows:
for k from 0 to 5 do x[k]:=-1.+k*h1;od;
for j from 1 to 7 do x[5+j]:=x[4]+2.*j*h1;od;
for k from 1 to 6 do x[12+k]:=x[12]+k*h1;od;

The plot with the approximation errors for the function $\exp(-x^2)$ with non-equidistant grid nodes and $n = 18$ is shown in Figure 3.

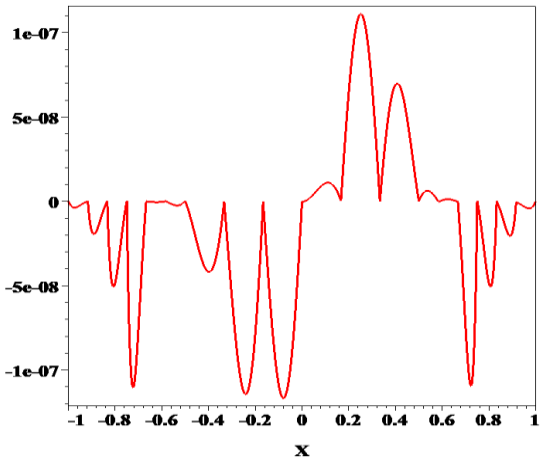

Fig. 3. Plot of the approximation error for the function $\exp(-x^2)$ with non-equidistant grid nodes.
Source: created by the authors.

Thus, a suitable choice of interpolation nodes makes it possible to reduce the approximation error.

3 The Application of Splines to Solve the Fredholm Integral Equation

Consider the Fredholm integral equation of the second kind:

$$u(x) - \int_a^b K(x,t)u(t)dt = f(x), \qquad (1)$$

assuming that the kernel $K(x,t)$ is continuous on the rectangle $[a,b] \times [a,b]$. We assume that the right-hand side of equation (1) is continuous on the interval $[a,b]$ and we obtain the solution to the equation within the class of functions continuous on $[a,b]$.

Selecting a grid of nodes x_k on $[a,b]$.

Let us rewrite the integral as follows:

$$\int_a^b K(x,t)u(t)dt = \sum_{k=0}^{n-1} \int_{x_k}^{x_{k+1}} K(x,t)u(t)dt.$$

Thus, using the spline approximation on each grid interval, we have the expression:

$$\int_a^b K(x,t)u(t)dt = \sum_{k=0}^{n-1} \int_{x_k}^{x_{k+1}} K(x,t)u(t)dt \approx$$
$$\sum_{k=0}^{3} \int_{x_k}^{x_{k+1}} K(x,t)U_1(t)dt + \sum_{k=4}^{n-5} \int_{x_k}^{x_{k+1}} K(x,t)U_2(t)dt$$

$$+\sum_{k=n-4}^{n-1}\int_{x_k}^{x_{k+1}} K(x,t)U_3(t)dt,$$

where:

$$U_1(t)=\sum_{i=k}^{k+8} u(t_i)\,w_i(t)\,,$$

$$U_2(t)=\sum_{i=k-4}^{k+4} u(t_i)\,w_i(t),$$

$$U_3(t)=\sum_{i=k-7}^{k+1} u(t_i)\,w_i(t)\,.$$

By replacing the unknown function with an approximation using the aforementioned splines, the problem reduces to calculating the integral of the product of the kernel and the basis function:

$$\int_{x_k}^{x_{k+1}} K(x,t)w_i(t)dt\,.$$

To approximate the integral over each grid interval, we use a Newton-Cotés quadrature formula with the appropriate algebraic degree of precision, or a Gaussian-type formula if a weight function contains a singularity. Here we use the Newton-Cotés quadrature formula:

$$\int_a^b g(y)dy \approx (b-a)\sum_{k=0}^{m} B_k g(y_k), \qquad (2)$$

where:

$$B_k=\frac{(-1)^{m-k}}{k!\,(m-k)!\,m}\int_0^{\mathrm{m}}\frac{t(t-1)\ldots(t-m)}{t-k}\,dt,$$

$$y_k=a+k\,h, k=0,1,\ldots m, h=(b-a)/m.$$

When the quantity of nodes is equal to 8 ($m=7$), we obtain the coefficients*:*
B_0:=751/17280; B_1=3577/17280;
B_2=1323/17280; B_3=2989/17280;
B_4=2989/17280; B_5=B_2; B_6=B_1; B_7=B_0.

In the middle of the interpolation interval $[a,b]$, we can use lower-order approximations (e.g., eighth or even seventh order) compared to those used near the boundaries of the interval $[a,b]$. It is significant that, in this case, midpoint splines are used; these midpoint splines have a smaller constant in the approximation error term.

Discarding the approximation error of the integral, we obtain $u_j=u(x_j)$.

3.1 Newton-Cotés Quadrature Formula Application

To solve the integral equation:

$$u(x)-\int_a^b K(x,t)u(t)dt\;=f(x),$$

we can use a simple or a composite Newton-Cotés quadrature formula with N nodes. To do this, we divide the interval $[a,b]$ into n equal grid intervals and apply the Newton-Cotés rule with $m+1$ nodes to each grid interval. This results in a system of linear algebraic equations (SLAE) of the order $N=1+\;+m\,n$.

3.2 The Gauss Quadrature Formula

We can also use the Gauss quadrature formula:

$$\int_{-1}^{1} f(x)dx \approx \sum_{k=1}^{n} A_k f(x_k)$$

which uses the roots of a Legendre polynomial as its nodes x_k:

$$P_n(x)=\frac{1}{2^n n!}\frac{d^n(x^2-1)^n}{dx^n}$$

and the coefficients:

$$A_k=\frac{2}{(1-x_k^2)[P'_n(x_k)]^2}, \qquad k=1,\ldots,n.$$

The error of the calculation of the integral is as follows:

$$R(f)=\frac{f^{(2n)}(\eta)}{(2n)!}\int_{-1}^{1}\omega^2(x)dx,$$

$$\eta\epsilon[-1,1],\ \omega(x)=(x-x_1)\cdot\ldots\cdot(x-x_n).$$

3.3 About Stability

As is well known, the convergence of a numerical method for solving integral equations of the second kind depends on two conditions: approximation and stability.

Below, we construct a numerical method for finding the solution to an integral equation and analyze its stability.

$$u(x)-\int_a^b K(x,t)u(t)dt\;=f(x).$$

Let's assume that $|K(x,t)|<r<1$.

3.3.1 Newton-Cotes Formula Application

First, we replace the integral with a Newton–Cotes quadrature formula (2).

Theorem 1*. [17] If the number of nodes $n+1$ in the Newton-Cotes formula (2) is odd, and the function g has a continuous derivative of order $n+2$ on the interval $[a,b]$, then there exists a point θ within the interval $[a,b]$ such that the following equality holds for the remainder $R(f)$:*

$$R(f)=\frac{f^{(n+2)}(\theta)}{(n+2)!}\int_a^b x\,w(x)dx.$$

Introduce a variable t and set $x = a + th$; then we obtain $w(x) = h^{n+1}t(t-1)\dots(t-n)$.

Theorem 2. *If the number of nodes n+1 in the Newton–Cotes formula (2) is odd, and the function g has a continuous derivative of order n+1 on the interval [a, b], then there exists a point θ within the interval [a, b] such that the following equality holds for the remainder R(f):*

$$R(f) = \frac{f^{(n+1)}(\theta)}{(n+1)!}\int_a^b w(x)dx.$$

Let us take a positive integer n: $a = 0$ and $b = 1$. Set $h = 1/n$ and construct the solution values $u_j^{[h]} = (u_0, u_1, \dots u_n)$ on the grid $x_k = kh, k = 0,1,\dots,n$. In this case, x_k are the nodes of the Newton–Cotes formula.

To construct the numerical method, we substitute x_k for x:

$$u(x_k) - \int_a^b K(x_k,t)u(t)dt = f(x_k). \qquad (3)$$

Replacing the integral with a sum using the Newton–Cotes quadrature formula, we obtain:

$$u_j - \sum_{k=0}^{n} B_k\, u_k\, K(x_j, x_k) = f(x_j),$$
$$j = 0,1,\dots,n.$$

Suppose $f^{[h]} = (f_0, f_1, \dots f_n), f_j = f(x_j), \quad |u_s| = \max_i |u_i|$ in $u^{[h]} = (u_0, u_1, \dots u_n)$. We have:

$$|f(x_s)| = \left| u_s - \sum_{k=0}^{n} B_k\, u_s\, K(x_s, x_k) \right| \geq$$
$$\left| |u_s| - r\,|u_s| \sum_{k=0}^{n} |B_k| \right| = |u_s|(1 - rB),$$

where

$$\sum_{k=0}^{n} |B_k| = B.$$

It is known that for $n < 8$, the coefficients in the Newton-Cotes formula are positive; therefore, for $a = 0$ and $b = 1$, we have $B = 1$.

Thus, the application of the simple Newton–Cotes formula is stable when $n < 8$. For $n > 8$, both positive and negative coefficients appear, which leads to significant computational instability. This can be observed when solving the examples presented in the Numerical Experiments section. In such calculations, the number of digits in the mantissa must be substantially greater than the number of grid nodes. We have the inequality

$$\|u^{[h]}\| \leq \frac{1}{1-r}\|f^{[h]}\|.$$

So we have stability when $n < 8$.

3.3.2 Gaussian Quadrature Formula Application

Now let us replace the integral with a sum using the Gaussian quadrature formula.

In this case we get:

$$|f(x_s)| \geq$$
$$\left| |u_s| - r\,|u_s| \sum_{k=0}^{n} |A_k| \right| = |u_s|(1-r),$$

Therefore:

$$\|u^{[h]}\| \leq \frac{1}{1-r}\|f^{[h]}\|.$$

Thus, the application of the simple Gaussian quadrature formula gives a stable method.

3.4 Examples

Example 1. Consider the integral equation from paper [6]:

$$u(x) = f(x) + \int_0^1 K(x,t)u(t)dt,$$

where:
$K(x,t) = \exp(t - x - 12)$,
$f(x) = \cos(x) - \frac{1}{2}\exp(-x-12)\,(\exp(1)\sin(1)$
$+\exp(1)\cos(1) - 1)$,
and the exact solution is as follows:

$$u(x) = \cos(x).$$

To solve the integral equation, we first apply our spline approximation method and then solve the same equation using the Newton-Cotés rule (applying first the simple rule and then the composite rule).

Figure 4 shows the errors obtained using the spline method when $n = 12$. Figure 5, Figure 6, and Figure 7 show the errors obtained using the Newton-Cotes method. In Figure 4, Figure 5, Figure 6 and Figure 7, the x-axis represents the nodes.

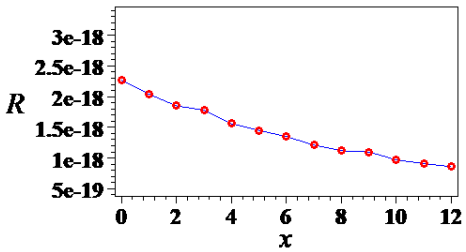

Fig. 4: The plot of the error of a solution to Example 1 obtained with the spline method
Source: created by the authors.

Now, let us apply the compound Newton-Cotés method. For $n = 10$ and $m = 7$, we obtain $N = 71$ equations. Table 1 and Table 2 present the absolute approximation errors at points t_i for $n = 1$ and $n = 12$, respectively.

Table 1. Absolute approximation errors at points t_i for $n=1$

t_i	$n = 1, m = 7$
0	5.52891e-14
0.14	4.78506e-14
0.29	4.15223e-14
0.43	3.59712e-14
0.57	3.11973e-14
0.71	2.69784e-14
0.86	2.32037e-14
1	2.02061e-14

Source: created by the authors.

Figure 5 shows the plot of the error of a solution to Example 1 obtained with the Newton-Cotés rule when $n = m = 7$. Note that the condition number of the system of equations is 9.966.

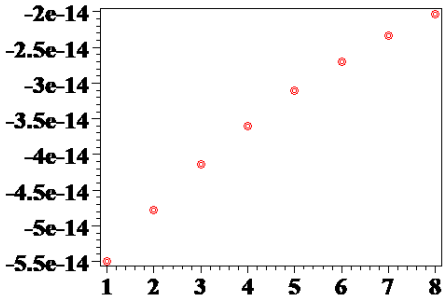

Fig. 5: The plot of the error of a solution to Example 1 obtained with the Newton-Cotés rule when $m = 7$
Source: created by the authors.

Figure 6 shows the plot of the error of a solution to Example 1 obtained with the simple Newton-Cotés rule when 30 nodes were taken and Digits=16. Figure 7 shows the plot of the error of a solution to Example 1 obtained with the simple Newton-Cotés rule when 30 nodes were taken and Digits=30.

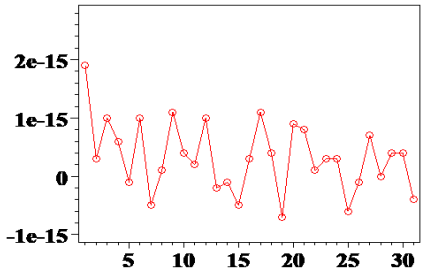

Fig. 6: The plot of the error of a solution to Example 1 obtained with the simple Newton-Cotés rule when 30 nodes were taken and Digits=16
Source: created by the authors.

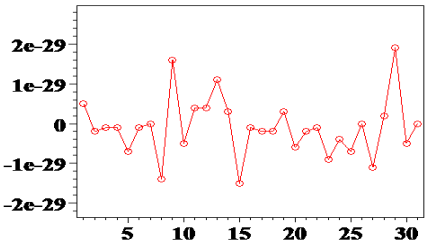

Fig. 7: The plot of the error of a solution to Example 1 obtained with the simple Newton-Cotés rule when 30 nodes were taken and Digits=30.
Source: created by the authors.

Table 2. Absolute approximation errors at points t_i for $n = 12, m = 7$

t_i	$n = 12, m = 7$
0	0
0.08	5.55112e-16
0.17	2.22045e-16
0.25	1.11022e-16
0.33	1.11022e-16
0.42	5.55112e-16
0.5	4.44089e-16
0.58	1.11022e-16
0.67	5.55112e-16
0.75	9.99201e-16
0.83	8.88178e-16
0.92	1.11022e-16
1	0

Source: created by the authors.

Note that the approximate solution using the spline method was obtained with the Maple software package (using `Digits= 20`), while the Newton-Cotés method was implemented in C++ (using type double). The calculation in Maple gave us the same result when 'Digits=15'. In both cases, the system matrix is dense. The spline method involved solving $n + 1$ equations, while the

Newton-Cotés method required solving N equations. When $n = 12$, $m = 7$, the application of the composite Newton-Cotés rule requires us to solve $N = 85$ equations. The condition number of the matrix does not exceed 2 when using the spline method.

The maximum solution error for the method proposed in [6] does not exceed 2.22e-8.

Example *2*. Consider the integral equation from paper [6]:

$$u(x) = f(x) + \int_{-1}^{1} K(x,t)u(t)dt,$$

where:

$$K(x,t) = \exp(t^2 - t - x^2 - 3),$$

$f(x) = \exp(-x^2)(\exp(-4) - \exp(-2) + 1)$,
and the exact solution is:

$$u(x) = \exp(-x^2).$$

Figure 8, Figure 9 and Figure 10 show the errors obtained with the spline method; note that the node numbers are plotted along the x-axis.

The plot of the error of a solution to Example 2 when $n = 12$ is given in Figure 8.

The plot of the error of a solution to Example 2 when $n = 32$ is given in Figure 9.

The plot of the error of a solution to Example 2 when $n = 64$ is given in Figure 10.

The condition number of the matrix does not exceed 2 when using the spline method.

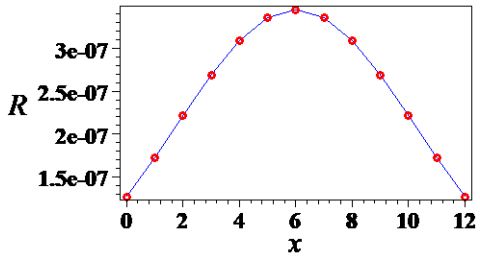

Fig. 8: The plot of the error of a solution to Example 2 when $n = 12$
Source: created by the authors.

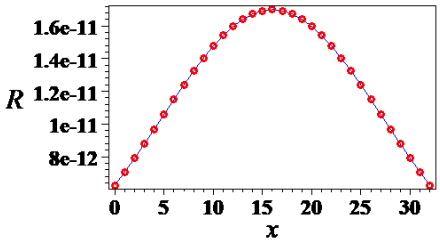

Fig. 9: The plot of the error of a solution to Example 2 when $n = 32$.
Source: created by the authors.

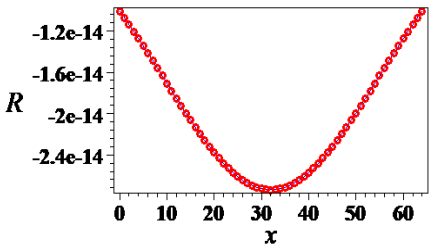

Fig. 10: The plot of the error of a solution to Example 2 when n = 64.
Source: created by the authors.

The maximum solution error for the method proposed in [6] does not exceed 5.1e-8.

Table 3 presents the errors of the solution to Example 2 when the simple Newton-Cotés rule was used and $n = 1$, $m = 7$.

Table 4 presents approximation errors (in absolute value) at points t_i for $n = 12, m = 7$ (C++, using type double).

Table 3. Absolute approximation errors at points t_i for $n = 1, m = 7$

t_i	$n=1$
-1	$4.27 \cdot 10^{-9}$
-5/7	$6.97 \cdot 10^{-9}$
-3/7	$9.67 \cdot 10^{-9}$
-1/7	$1.14 \cdot 10^{-8}$
1/7	$1.14 \cdot 10^{-8}$
3/7	$9.67 \cdot 10^{-9}$
5/7	$6.97 \cdot 10^{-9}$
1	$4.27 \cdot 10^{-9}$

Source: created by the authors.

The plot of the error of a solution to Example 2 is given in Figure 11 when the simple Newton-Cotés rule was used and $n = 30$, Digits=16.

The plot of the error of a solution to Example 2 is given in Figure 12 when the simple Newton-Cotés rule was used and $n = 30$, Digits=30.

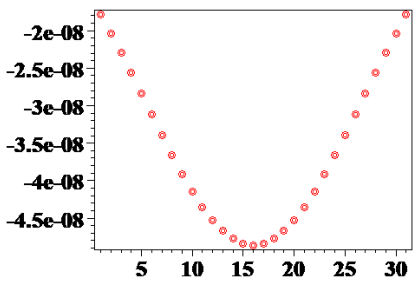

Fig. 11: The plot of the error of a solution **to** Example 2 when the simple Newton-Cotés rule was used and $n = 30$, Digits=16, and the Condition number=0.447 10^9
Source: created by the authors.

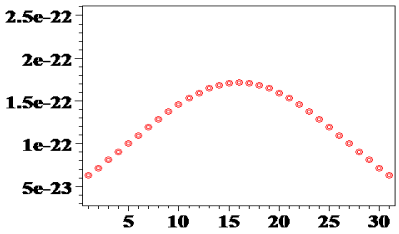

Fig. 12: The plot of the error of a solution to Example 2 when the simple Newton-Cotés rule was used and n = 30, Digits=30, and the Condition number=0.447 10^9
Source: created by the authors.

Table 4. Absolute approximation errors at points t_i for n = 12, m = 7

t_i	*n=12*
-1	$5.55 \cdot 10^{-17}$
-5/6	$5.55 \cdot 10^{-17}$
-4/6	$1.11 \cdot 10^{-16}$
-3/6	0
-2/6	$3.33 \cdot 10^{-16}$
-1/6	$2.22 \cdot 10^{-16}$
0	$2.22 \cdot 10^{-16}$
1/6	$2.22 \cdot 10^{-16}$
2/6	$1.11 \cdot 10^{-16}$
3/6	$2.22 \cdot 10^{-16}$
4/6	$2.22 \cdot 10^{-16}$
5/6	$1.11 \cdot 10^{-16}$
1	$5.55 \cdot 10^{-17}$

Source: created by the authors.

Below, we present a table of coefficients for the Newton-Cotes formula which were used when solving Example 2:

$B_0 = B_{30} = 0.01464645406231278$
$B_1 = B_{29} = 0.1814802933672462$
$B_2 = B_{28} =$ -0.6882336410075590
$B_3 = B_{27} = 4.689152168464531$
$B_4 = B_{26} =$ -23.60243283978322
$B_5 = B_{25} = 100.8882786721750$
$B_6 = B_{24} =$ -360.9257862506090
$B_7 = B_{23} = 1099.353268775373$
$B_8 = B_{22} =$ -2876.521258800876
$B_9 = B_{21} = 6523.869347742281$
$B_{10} = B_{20} =$ -12910.86087604653
$B_{11} = B_{19} = 22418.03554224070$
$B_{12} = B_{18} =$ -34294.60669755750
$B_{13} = B_{17} = 46364.94428538871$
$B_{14} = B_{16} =$ -55514.4645012251
$B_{15} = 58941.38756925335$

Example 3
Let us consider the integral equation from the article [16].

$$u(x) - 1/\pi \int_{-1}^{1} \frac{u(t)dt}{(1+(x-t)^2)} = f(x)$$

with the exact solution $u(x) = 1 + x/2 + x^2/4$.

The error of the solution obtained with our splines of the ninth order is given in Figure 13.

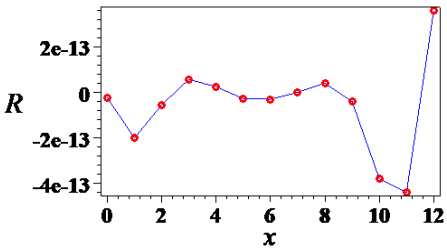

Fig. 13: The plot of the error of a solution to Example 3 when the spline method was used and n = 12, Digits=15
Source: created by the authors.

The error of the solution obtained with the Gausses method is given in Figure 14. The error of the solution obtained with the Newton-Cotes method is given in Figure 15 ($n = 12$, Digits=15) and Figure 16 ($n = 64$, Digits=60).

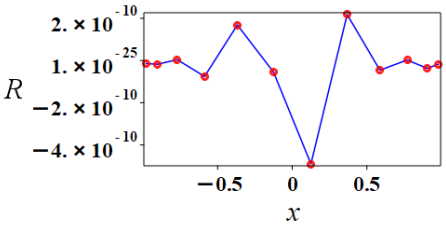

Fig. 14: The plot of the error of a solution to Example 3 when the Gausses method was used and n = 12, Digits=15
Source: created by the authors.

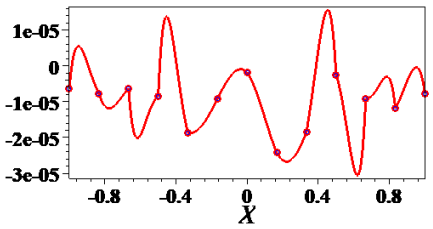

Fig. 15: The plot of the error of a solution to Example 3 when the Newton-Cotes method was used and $n = 12$, Digits=15. Here we connected the points with our splines
Source: created by the authors.

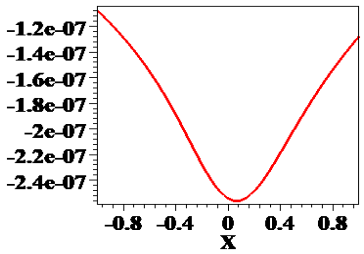

Fig. 16: The plot of the error of a solution to Example 3 when the Newton-Cotes method was used and $n = 64$, Digits=60
Source: created by the authors.

Example 4. Now we solve the integral equation from paper [16]:

$$u(x) - \int_0^1 \frac{\exp(xt)u(t)dt}{2} = f(x)$$

with the exact solution $u(x) = \exp(-x)\cos(x)$.

The error of the solution obtained with our splines of the ninth order when $n = 12$ is given in Figure 17, the error of the solution obtained when $n = 32$ is given in Figure 18, and the error of the solution obtained when $n = 128$ is given in Figure 19.

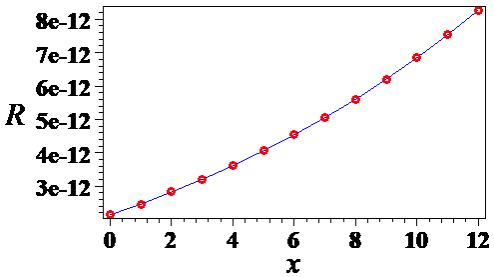

Fig. 17: The plot of the error of a solution to Example 4 when the spline method was used and n = 12, Digits=15
Source: created by the authors.

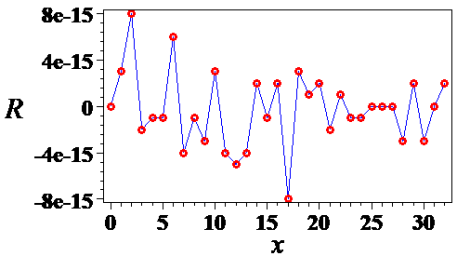

Fig. 18: The plot of the error of a solution to Example 4 when the spline method was used and n = 32, Digits=15
Source: created by the authors.

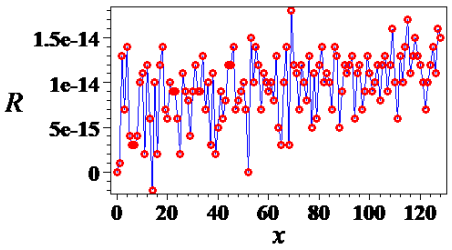

Fig. 19: The plot of the error of a solution to Example 4 when the spline method was used and n = 128, Digits=15
Source: created by the authors.

Example 5. The next integral equation is from paper [16]:

$$u(x) = f(x) + 1/\pi \int_{-\pi}^{\pi} K(x,t)u(t)dt,$$

Here the kernel is as follows:

$$K(x,t) = \frac{30}{100 - 64\cos^2\left(\frac{x}{2} + \frac{t}{2}\right)},$$

the exact solution of the integral equation is $u(x)$= $(17/2 + 128/17\cos(2x)$, and the function $f(x)$ is $f(x) = 25 - 16\sin(x)^2$.

The plot of the error of a solution to Example 5 when the spline method was used and $n = 12$ is given in Figure 20. The plot of the error of a solution to Example 5 when the spline method was used and $n = 128$ is given in Figure 21.

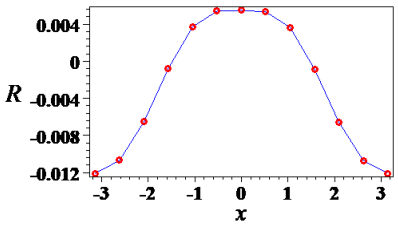

Fig. 20: The plot of the error of a solution to Example 5 when the spline method was used and $n = 12$, Digits=15
Source: created by the authors.

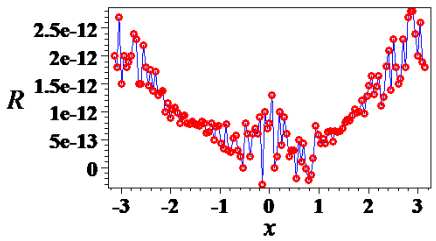

Fig. 21: The plot of the error of a solution to Example 5 when the spline method was used and $n = 128$, Digits=15
Source: created by the authors.

4 Conclusion

The specifics of applying various quadrature formulas to the solution of integral equations are discussed in numerous works. We highlight the following, published in recent years:

For solving Fredholm integral equations of the second kind, the use of Gaussian-type quadrature formulas—both simple and composite—yields good results. Spline approximations are advantageous due to the ease with which the framework of an approximate solution can be constructed; subsequently, they can be used to derive a continuous or continuously differentiable solution while maintaining the same order of approximation.
The numerical experiments demonstrate that, in the absence of a singularity in the kernel, both spline approximations and Newton-Cotés formulas yield good results; however, the compound Newton-Cotés method requires solving a system with a significantly larger number of equations. The spline approximation method for solving integral equations can easily be generalized to any continuous spline approximations of maximal defect and arbitrary approximation order.

In future papers, we will consider the case of kernels with singularities.

Acknowledgement:
The authors acknowledge Saint-Petersburg State University for research project 157438120.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work, the author(s) utilized artificial intelligence (AI) technologies solely for the purpose of linguistic refinement and grammatical correction. Following the application of these tools, the author(s) comprehensively reviewed, verified, and edited the resulting text. The author(s) maintain full accountability and responsibility for the accuracy, integrity, and final content of the publication.

References:

[1] Robert Piessens, Computing integral transforms and solving integral equations using Chebyshev polynomial approximations, *Journal of Computational and Applied Mathematics*, 121, 2000, pp. 113–124.

[2] J. Rashidinia1, E. Babolian, Z. Mahmoodi, Spline Collocation for Fredholm Integral Equations, *Mathematical Sciences*, Vol. 5, No. 2, 2011, pp. 147-158.

[3] Ezzati, Reza & Shakibi, Katayoon, On Approximation and Numerical Solution of Fredholm-Hammerstein Integral Equations Using Multiquadric Quasi-interpolation, *Communications in Numerical Analysis*, 2012. 1-10.doi: 10.5899/2012/cna-00112.

[4] R. S. Shamsah, S. Jain, P. Agarwal and S. Momani, Analytic methods to solve Fredholm integral equations, Montes Taurus J. *Pure Appl. Math*. 7 (2), 1–19, 2025; Article ID: MTJPAM-D 24-00121.

[5] Yilmaz, Bulent, and Yasin Cetin, Numerical Solutions of the Fredholm Integral Equations of the Second Type, *New Trends in Mathematical Sciences*, Vol. 5, No. 3, July 2017, pp. 284-92, [Online] https://izlik.org/JA94DZ47HN (Accessed Date: January 10, 2026).

[6] K. Maleknejad, J. Rashidinia, H.Jalilian, *Computational methods for differential equations (CMDE)*, Vol. 9, No. 1, 2021, pp. 211-224, DOI: 10.22034/cmde.2019.31983.1492.

[7] Kholmat M. Shadimetov, Farxod A. Nuralieva, Shaxobiddin S. Kuziev, Numerical solution Fredholm integral equation of the second kind by the optimal quadrature method, *Filomat*, 39:32, 2025, 11653–11667.

[8] Zhilin Li, Kejia Pan, Juan Ruiz, Dionisio F. Yáñez, Fully accurate approximation of piecewise smooth functions using corrected B-spline quasi-interpolants, *Computational and Applied Mathematic*s, 2024, 43:165 https://doi.org/10.1007/s40314-024-02651-4.

[9] F. Muller and W. Varnhorn, On approximation and numerical solution of Fredholm integral equations of second kind using quasi-interpolation, *Appl. Math. Comput*, 217, 2011, pp.6409–6416.

[10] Abdullo Hayotova,Samandar Babaeva, The numerical solution of a Fredholm integral equations of the second kind by the weighted optimal quadrature formula, *Results in Applied Mathematic*s**,** vol. 24, November 2024, 100508.

[11] I. G. Burova, Fredholm Integral Equation and Splines of the Fifth Order of Approximation, *WSEAS Transactions on Mathematics*, Vol.21, 2022, pp.260-270, DOI: 10.37394/23206.2022.21.31.

[12] Alireza Eskandarinejad, David Arellano, Rouzbeh Nazari, Mohammad Reza Nikoo, Regression Splines (MARS) analysis

and complementary machine learning approaches in predicting earthquake‐induced slope displacements, *Earth Science Informatics*, 2025, 18:398, https://doi.org/10.1007/s12145-025-01899-6.

[13] Randall Balestriero, Romain Cosentino, Herv´e Glotin, Richard Baraniuk, Spline Filters For End-to-End Deep Learning, *Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden*, PMLR 80, 2018.

[14] Poulinakis, K.; Drikakis, D.; Kokkinakis, I.W.; Spottswood, S.M. Machine-Learning Methods on Noisy and Sparse Data. *Mathematics*, 2023, 11, 236, https://doi.org/10.3390/ math11010236.

[15] Hrushikesh N. Mhaskara, Efstratios Tsoukanisa, Ameya D. Jagtap, An approximation theory perspective on machine learning, *Neural Networks* 200, 2026, 108841.

[16] Mohamed Addam Imad ElBarkan, Computational automation for solving Pocklington's integro-differential equation based on Fredholm integral equations and-spline interpolants, *Afrika Matematika*, 2026, 37:51 https://doi.org/10.1007/s13370-026-01446-6.

[17] Krylov, V. I. Approximate calculation of integrals. Russian edition: Moscow, 1959. English translation by A. H. Stroud: Macmillan, 1962.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

- I. G. Burova presented theoretical and empirical results regarding the use of spline approximations.
- N. A. Lebedinskaya and D.M.Lebedinsky conducted experiments using Newton-Cotés formulas.
- E. Patsukevich conducted a literature review.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

The authors acknowledge Saint-Petersburg State University for research project 157438120.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

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