Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 5 Issue 12, December - 2018

High-accuracy Numerical Solution of the Second-kind Integral Equations in the Mathematica Environment

Shapeev V.P.
Khristianovich Institute of Theoretical and Applied Mechanics, Siberian Branch of the Russian Academy of Sciences, Novosibirsk 630090, Russia;
Novosibirsk National Research University, Novosibirsk 630090, Russia
vshapeev@gmail.com

Vorozhtsov E.V.
Khristianovich Institute of Theoretical and Applied Mechanics, Siberian Branch of the Russian Academy of Sciences, Novosibirsk 630090, Russia
vorozh@itam.nsc.ru

***Abstract*— A p-version of the collocation method for the numerical solution of the Fredholm integral equations of the second kind has been proposed and implemented. In the given implementation, the possibilities have been realized for the variation of the polynomial degree in the polynomial representation of the approximate solution of equations and the variation of the number of nodes of the employed Gauss quadrature formula to affect the solution accuracy. The influence of the number of collocation points used for the solution approximation and of the number of nodes of the Gauss quadrature formula on the condition number of the system of linear algebraic equations to the solution of which the construction of the approximate solution reduces, and on its accuracy has been investigated by numerical solution of the examples, which include the examples presented in well-known publications. The proposed algorithm has been implemented in the language of the program package Mathematica. In all considered examples, the proposed version of the collocation method has enabled us to reach the accuracy of the solution of equations, which is close to the level of the machine rounding errors. The computer code implementing the proposed p-version has proved to be compact, and the method turned out economical: the machine time required for the solution of problems considered in the work did not exceed 3 seconds of the CPU time of a personal computer. The algorithm has been described, which makes it possible to assess the accuracy of the approximate solution by the proposed p-version of the method in the cases where the exact solution of the integral equation is unknown.**

Keywords—Fredholm integral equation of the second kind, collocation method, condition number, Gauss quadrature.

I. INTRODUCTION

At present, the integral equations have gained widespread acceptance at the investigation and solution of many problems in physics (electrodynamics [1, 2], superconductivity [3], solid state physics [4]), geophysics (gravimetry, seismic prospecting [5, 6]), computerized tomography [7], chemistry [8], mathematical biology [9], digital image processing [10], etc.

Until now, several numerical methods were developed and analyzed for solving the Fredholm integral equations of the second kind: these are, for example, various versions of the Galerkin methods [11, 12, 13, 14, 15, 16, 17], meshless methods [18]. In addition, the methods of resolvent, degenerate kernels, Fourier transform, and several other [19] are the conventional methods applied mainly at the solution of the integral equation of the second kind.

The methods for solving the integral equations differ from one another primarily by a technique for approximating the solution and by the corresponding discretization technique — the passage from the infinitely dimensional functional space, in which the exact solution of the equation is located, to a finite-dimensional functional space in which its approximation (the approximant) is sought for.

The collocation method is used in practice, first of all, for solving the ordinary differential equations [20, 21] and the one-dimensional integral equations. This method was served as a basis at the development of the COLSYS program package for solving the systems of ordinary differential equations [22], which is also at present a good tool for obtaining high-accuracy solutions of differential equations and is used for solving applied problems. The collocation method is applied also for the numerical solution of the Fredholm integral equations of second kind [19, 23, 24, 25, 26].

One of the advantages of the collocation method is the simplicity of its computer implementation and the possibility of reaching in a sufficiently simple way a high approximation order of the equations. This enables one, respectively, to reach high solution accuracy. In addition, this method possesses a possibility to describe the boundary conditions for the

approximate problem directly at the points lying at the region boundary when solving a multi-dimensional boundary-value problem in an irregular region [27]. This circumstance is very important for the goal of reaching a high accuracy of the solution.

As a rule, the condition number of a system of linear algebraic equations (SLAE) to be solved increases with the increasing size of the SLAE arising as a result of the original problem discretization at the solution of differential and integral equations by numerical methods, and this is a shortcoming of the collocation method too. The possibilities of the numerical solution of such problems are extended significantly when the collocation method is combined with the method of least squares. This has been shown in [27, 28, 29, 30, 31] and in a number of other works.

One can also extend the capabilities of the collocation method by using a “gridless” version — the p-version of the method in which, in contrast with the (grid) h-version, the entire region of the numerical solution of the problem consists of a single interval or of a single cell as in the spectral method. Therefore, such versions of the methods are frequently termed also pseudo-spectral methods.

In the p-version of the collocation method, a high accuracy of the approximation is realized fairly simply by increasing the number of basis elements of the employed functional space. In the case of the solution of integral equations, the accuracy of the solution approximation, which depends on the smallness of the error of the employed quadrature formula, also increases owing to the use of high-accuracy quadrature formulas, for example, the Gauss formula. It will be shown in the following that the both above factors are easily implemented on computer in a compact program.

The following fact is substantial here: with the increasing number of basis elements and nodes of the quadrature formula, the size of the corresponding SLAE of the approximate problem grows *more slowly* than in its grid version — h-version at the grid step refinement. One can efficiently use this property of the p-version.

The application of the collocation method for the numerical solution of integral equations has been known for a long time [19, 32], however, its possibilities for obtaining high-accuracy solutions are far from exhausted, and this is shown below in this article.

In the present work, a realization of the p-version of the collocation method for the numerical solution of the Fredholm integral equations of the second kind in the environment of the *Mathematica* system [33] is described. In this version, the solution is sought in the form of a polynomial of a sufficiently high degree in the case of using polynomial spaces.

It is shown by the examples that one can obtain by the proposed numerical method the numerical solution of problems with an error close to the error of the arithmetic operations on computer.

The method has been implemented in a computer code written in the language of the program package *Mathematica*. In the p-version of the algorithm, the entire computational interval represented a “single cell” of the computational grid in which the solution was sought in the form of a polynomial of a sufficiently high degree.

It is known that in the case of the approximation of smooth curves by polynomials, the location of approximation nodes in the roots of orthogonal polynomials (of Chebyshev, Legendre, etc.) often yields a much smaller error than in the case of an equidistant location of nodes (collocation points). In this connection, the Gauss quadrature formula with the number of nodes from 6 to 45 was employed in the developed *Mathematica* program at the computation of the integral entering the integral equation. This program used the zeroes of the Legendre polynomials as the nodes. Besides, the collocation points were also specified in the roots of the Legendre polynomials the degree of which depended on the degree of the polynomial approximating the solution.

The corresponding program in the language of the *Mathematica* system has proved to be fairly compact also owing to the fact that this system has such convenient built-in functions as the function for solving a system of linear algebraic equations, the function for calculating all eigenvalues of the matrix, and many other functions which are useful at the realization of numerical methods. The program makes provision for varying the number of basis elements in the polynomial space in which the approximate solution is constructed and varying the number of nodes of the employed Gauss quadrature formula. By varying the above parameters at the use of the program one can reach an increased accuracy of the solution of equations under consideration. In the case when the user does not need increased solution accuracy there is a possibility of choosing the values of these parameters, which ensure the needed accuracy. A reduction of the values of these parameters leads to a reduction of the solution accuracy and a speed-up of the program work. It is, however, well known that in many cases, besides other tools and means employed by the researchers, a high accuracy of computations, devices, and measurements contributed to new discoveries and the general progress of science. For example, U.J.J. Le Verrier had discovered the planet Neptune with the aid of an increased accuracy of computations. It is, therefore, important to have a possibility of solving the problems with an increased accuracy.

The results of numerical experiments with the p-version of the algorithm are presented below in the tables and by the graphs from the analysis of which one can judge about the capabilities of the proposed

method. To understand the behavior of the solution error depending on the condition number of the SLAE to the solution of which the solution of the given equation has been reduced the tables contain also a column presenting the condition number values.

II. Description of the Collocation Method for the Numerical Solution of Integral Equations

We will consider both the linear and nonlinear Fredholm equations of the second kind. The linear second-kind Fredholm equation has the following form:

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

The functions $f(x)$ and $g(x)$ are assumed analytic in the interval $[a, b]$. The kernel $K(x, s)$ is assumed analytic in the region $[a, b]\times[a, b]$. The parameter λ and the interval $[a, b]$ are assumed given. It is also assumed that $-\infty < a < b < \infty$. The existence and uniqueness results for (1) may be found in [32, 34].

The nonlinear second-kind Fredholm equation has the following form:

$$u(x)-\lambda\int_a^b G(x,s,u(s))ds = f(x), \quad x\in[a,b]. \qquad (2)$$

The function $G(x,s,u(s))$ is assumed analytic in the region $[a, b]\times[a, b]$. The assumptions for $f(x)$, the parameter λ, and the interval $[a, b]$ are the same as in the case of (1). The existence and uniqueness results for (2) may be found in [35, 19, 32, 4].

Let us describe the p-version of the algorithm for solving (1) by the collocation method. Let x_c be the center of the interval $[a, b]$ that is $x_c = (a + b)/2$, and let $h = (b – a)/ 2$. It is convenient, by analogy with [2, 3], to introduce the new independent variable y by formula $y = (x – x_c)/h$. It is obvious that $y\in[-1,1]$ at $x\in[a,b]$. Let $U(y) = u(x_c + hy)$. Let us make in (1) the substitution $x = x_c + hy$. As a result, (1) takes the following form:

$$g(x_c+hy)U(y)-\lambda h\int_{-1}^{1} K(x_c+hy,x_c+h\xi,u(x_c+h\xi))U(\xi)d\xi = F(y), \quad (3)$$

where $y\in[-1,1]$, $F(y) = f(x_c + hy)$.

The approximate solution of (3) is sought in the interval $–1 ≤ y ≤ 1$ in the form of a polynomial:

$$U(y)=\sum_{k=1}^{n} b_k y^{k-1}, \qquad (4)$$

where b_k are the unknown coefficients and n is a given number of the degrees of freedom. It is assumed that the degree of polynomial (4) $N_p = n – 1 > 0$. Substituting expression (4) in (3), we obtain an algebraic equation containing n unknown coefficients $b_1,…,b_n$. To find these coefficients let us specify $N_c = n$ collocation points y_j in the interval [-1,1],], $j = 1,…,N_c$.

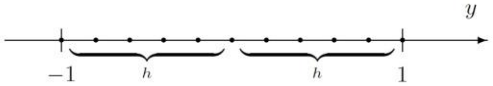


Fig. 1. *Locations of collocation points.*

In the case of using the roots of the Legendre polynomials as collocation points we at first specify two collocation points at the boundaries $y = –1$ and $y = 1$, and the remaining $N_c – 2$ collocation points are found as the zeroes of the Legendre polynomial of the degree $N_c – 2$. It is also possible to specify the collocation points in the nodes of a uniform grid $y_j = –1 + 2(j – 1)/(N_c – 1)$, $j = 1,…,N_c$. Fig. 1 shows the case when $N_c = 11$.

After the substitution of numerical values of the collocation point coordinates in the collocation equations, they become linear algebraic equations. In this way, the following SLAE is obtained for determining the solution vector column $\mathbf{X} = (b_1,…,b_n)$

$$\mathbf{AX} = \mathbf{\Phi}, \qquad (5)$$

where $\mathbf{A}$ is an $n×n$ matrix. In the present work, this system was solved by using the built-in function **`LinearSolve[A,Φ]`** of the system *Mathematica*. The performance of the developed compact program of the p-version of the collocation method is demonstrated in the following by several examples.

III. Results of Numerical Experiments

Example 1. In the cases when it is necessary to elucidate the capabilities and the limits of the applicability of a new numerical method for solving the problems under study it is useful to use smooth solutions for the method verification, which are obtained by the method of “manufactured solutions” (MMS), which was originally developed by Salary and Knupp [36]. An excellent overview of the method is given in [37]. The motivation for MMS lies in the difficulty of obtaining analytical solutions to the governing equations in many physical contexts. MMS proposes that instead of attempting to find a solution to the governing equation, one modifies the equation to match a solution of ones choosing.

We consider here a fairly simple test example of (1), for which we set $g(x) = 1$ and $K(x,s) = 1$. According to the MMS let us specify the exact solution in the form

$$u(x) = x^5e^x, \qquad x\in[a, b]. \qquad (6)$$

and set $a = –1$, $b = 1$. In this case, as we will see below, there is no need in the developed p-version of the method to take a large number of nodes in the Gauss formula. To find the corresponding right-hand side we substitute solution (6) in (1). We obtain the following expression for $f(x)$:

$$f(x) = x^5e^x - \frac{326\lambda}{e} + 44\lambda e.$$

The computations by the collocation method were done in the *Mathematica* system at $\lambda = 1$ in (1). In this case, $x_c = 0$, $h = 1$.

TABLE I. ERROR $||U - U_h||_C$ AS A FUNCTION OF PARAMETERS N AND N_{Gauss} IN EXAMPLE 1.

No.	n	N_{Gauss}	$\|\|u - u_h\|\|_C$	$\kappa(\mathbf{A})$	CPU time, sec.
1	9	6	4.969e–4	4.578e2	0.19
2	11	8	4.010e–6	2.684e3	0.23
3	14	16	9.753e–10	3.902e4	0.38
4	15	10	4.864e–11	9.200e4	0.39
5	15	16	4.864e–11	9.200e4	0.45
6*	20	16	1.155e–14	7.746e6	0.22
7	20	—	9.132e–14	7.737e6	5.64

The goal of the series of computations the results of which are presented in Table I was the investigation of the influence of the parameters n and N_{Gauss} on the accuracy of the result, where N_{Gauss} is the number of nodes of the Gauss quadrature formula for approximate computation of the integral. The tables of the numerical values of the nodes x_i and weights w_i, i = 1,…, M for M = 2,…,64 are available on the site https://pomax.github.io/bezierinfo/legendre-gauss.html. The site also contains the *Mathematica* program for calculating the x_i and w_i. The symbol * marks in Table I a row with the result, which is the best in terms of accuracy.

The condition number $\kappa(\mathbf{A})$ of the matrix $\mathbf{A}$ was computed by the formula $\kappa(\mathbf{A}) = \left(\max_i |\lambda_i| / \min_i |\lambda_i|\right)^{0.5}$, where λ_i, i = 1,…,n are the eigenvalues of the matrix $\mathbf{A}^T\mathbf{A}$, where the superscript T denotes the transposition operation.

All numerical computations described in the present work were done on a desktop computer with an Intel processor with tact frequency 3 GHz. The extreme right column of Table I presents the CPU time for each run on the above computer.

The dash in the third column of Table I means that in this numerical experiment, the integral entering (3) was calculated exactly by using the antiderivative. In all such cases when it is possible to calculate the integral in a closed analytic form in the software system *Mathematica*, there is no need to use the approximate quadrature formula. However, as one can see in Table I, without the use of the quadrature formula (row 1 in Table I), the CPU time was required, which proved to be 25 times larger than at the use of the Gauss quadrature formula (row 6* in Table I). This is explained by that the closed form of integral entering (3) involves an analytic expression whose length exceeds several times the integrand size. As a result, the computation of the closed-form expression requires the execution of a large number of arithmetic operations, which exceed significantly the number of arithmetic operations in the given case in comparison with the Gauss quadrature. A situation has occurred that from the viewpoint of minimizing the CPU time, an analytic expression for the integral requires a much larger CPU time than the application of the numerical, approximate procedure for computing the value of the integral.

It follows further from a comparison of rows Nos. 6* and 7 of Table I that at the use of the Gauss quadrature formula, the collocation solution error proved to be by the factor of about 8 less than in the case of the application of the analytic formula for the integral entering (3). This means that the error of the quadrature formula at chosen parameters of the formula is close to the rounding errors on computer, and the error in both cases of the integral computation is determined by the number of arithmetic operations needed for computing its value. Thus, we arrive at a conclusion that from the viewpoint of the speed of the problem solution and the accuracy of the approximate solution, it is recommended to apply the Gauss quadrature formula to reach a high solution accuracy, including the case of an error close to the rounding errors on computer independently of the possibility of computing the integral in (3) in closed analytic form. It is natural that when the above solution accuracy is not required one can restrict oneself to a small number of the nodes and weights in the quadrature formula and a small number of collocation points, and as a result, the CPU time will then be smaller. One can also note that the condition number of the SLAE to be solved grows with the increasing number of unknowns n in the polynomial representation (4) of the approximate solution. However, its growth proves to be non-fatal to impede the solution obtaining with an error of the order 10^{-14} even at n = 20. For example, in Table I, the SLAE size has increased by a factor of nearly two at a passage from the first row to the row No. 6, and the solution accuracy has increased by a factor of more than 10^{10}. A similar conclusion may be drawn from the analysis of the solutions of the second and third examples presented below.

The computations in the arithmetic of floating-point machine numbers are used in the program package *Mathematica*, which has been implemented in the C language. This package employs the machine numbers having the length “long double”'. This length is somewhat larger than the length “double precision”. This circumstance contributes to the feasibility of obtaining very small solution errors. At the use of the “double precision” numbers, the solution accuracy attainable on computer will differ from the accuracy reached here nearly by two decimal orders. This is practically not so essential at such small magnitudes of the error, which were achieved here.

We have also carried out a numerical experiment, in which the collocation points were placed at n = 20 in the nodes of a uniform grid in the interval [–1, 1]. The error $||u - u_h||$ = 2.630e^{-13} was obtained, which is by the factor of 23 larger than in the case of placing the collocation points in the nodes of a non-uniform grid lying in the zeroes of the Legendre polynomial of the 16th degree (cf. row No. 6* of Table I).

Finally, a series of computation was done in which the endpoints y = –1 and y = 1 of the equation solution interval were eliminated from the set of collocation

TABLE II. THE VALUES OF THE ROOTS y_L VS. n.

No.	n	N_{Gauss}	y_L
1	9	6	±0.968160
2	14	16	±0.986284
3	20	16	±0.993129
4	30	24	±0.996893
5	42	35	±0.998400

TABLE III. INFLUENCE OF THE ELIMINATION OF ENDPOINTS FROM THE SET OF COLLOCATION POINTS ON THE APPROXIMATE SOLUTION ACCURACY

No.	n	N_{Gauss}	$\|\|u-u_h\|\|_C$	$\|\|u-u_h\|\|_{C,w.e.p.}$	δ
1	9	6	4.969e–4	5.805e–4	1.168
2	14	16	9.753e–10	1.287e–9	1.320
3	20	16	1.155e–10	1.199e–10	1.038
4	30	24	1.205e–14	1.155e–14	0.959
5	42	35	1.199e–14	1.199e–14	1.000

points, and all collocation points were computed as the roots of the Legendre polynomial of the nth degree, see Table III. In Table II, y_L are the roots of this polynomial, which are the closest to the endpoints $y = \pm 1$. In Table III, $||u - u_h||_{C,w.e.p.}$ is the approximate solution error, which was obtained without using the endpoints in the set of collocation points, $\delta = ||u - u_h||_{C,w.e.p.} / ||u - u_h||_C$. One can see that in the interval $9 \le n \le 20$, the error $||u - u_h||_{C,w.e.p.}$ decreases, but still remains higher than the error $||u - u_h||_C$. In addition, the quantity δ decreases in the interval $14 \le n \le 30$. This effect is explained by that with increasing degree of the Legendre polynomial, its roots closest to the endpoints of the equation solution interval tend to these points, see the fourth column in Table II. One can draw the following general conclusion from this table: if the values n and N_{Gauss} are not very large, then it is advisable to include the endpoints $y = \pm 1$ in the set of collocation points to obtain an increased accuracy of the result. However, starting from some sufficiently high values of n and N_{Gauss}, the presence of endpoints in the set of collocation points does already not lead to an increase in the accuracy of the numerical result obtained by the proposed collocation method.

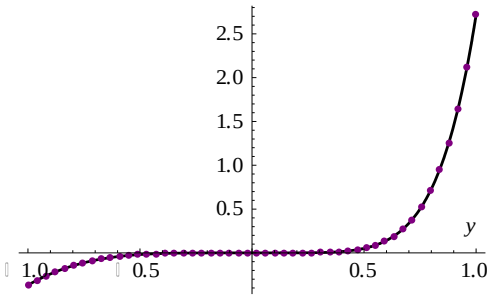
It is reasonable to specify the collocation points with the elimination of endpoints in the cases when the kernel $K(x,s)$ has a singularity at one or the both endpoints.

Fig. 2 shows the results of the computations by the collocation method at $n = 20$ and $N_{Gauss} = 16$. The exact solution $u(x)$ is depicted by the solid line in Fig. 2, (*a*), and the numerical solution u_h is plotted by a dotted line. As is seen in Table 1 and in Fig. 2, (*b*), $||u - u_h||_C$ does not exceed $1.2 \cdot 10^{-14}$.

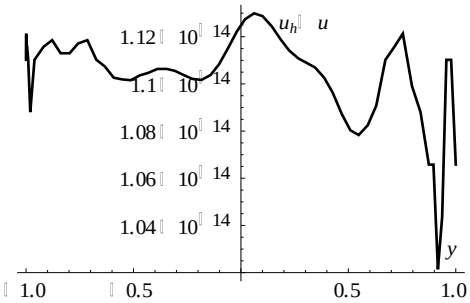
Example 2. Consider (1) at the following expressions for functions $g(x)$, $K(x,s)$, and $f(x)$:

TABLE IV. ERROR $||u - u_h||_C$ AS A FUNCTION OF PARAMETERS N AND N_{Gauss} IN EXAMPLE 2.

No.	n	N_{Gauss}	$\|\|u-u_h\|\|_C$	$\kappa(\mathbf{A})$	CPU time, sec.
1	9	6	1.179e–5	4.432e2	0.14
2	11	8	3.177e–9	2.589e3	0.24
3	14	16	5.016e–13	3.650e4	0.20
4	15	10	7.672e–14	8.853e4	0.22
5	15	16	1.660e–14	8.853e4	0.30
6*	20	16	1.332e–15	7.307e6	0.31



(*a*)



(*b*)

Fig. 2. *Results of solving the integral equation of Example 1 at n = 20 and N_{Gauss} = 16.*

$$g(x)=1,\ K(x,s)=se^{s}\sin(x+s),$$
$$a=-1,\quad b=1,\quad \lambda=1,$$
$$f(x)=e^{x}\sin(x)-\frac{\lambda}{16}\{e^{2}[2\cos(x)-\sin(2+x)-2\cos(2+x)]-$$
$$e^{-2}[3\sin(x-2)-6\cos(x)+2\cos(x-2)]\}.$$

The exact solution has the following form: $u(x) = e^x \sin(x)$. Table IV presents the value of the error norm $||u - u_h||_C$, which was computed by using 1000 points distributed uniformly in the region $-1 \le x \le 1$. It follows from a comparison of Tables I and IV that at the same values of n and N_{Gauss}, the numerical solution accuracy is higher in Example 2 than in Example 1. To explain this effect we present the following well-known estimate [38] the norm of the error of the SLAE (5) solution:

$$\frac{\|\delta\mathbf{X}\|}{\|\mathbf{X}\|}\le\frac{\kappa(\mathbf{A})}{1-\kappa(\mathbf{A})\dfrac{\|\delta\mathbf{A}\|}{\|\mathbf{A}\|}}\left(\frac{\|\delta\mathbf{\Phi}\|}{\|\mathbf{\Phi}\|}+\frac{\|\delta\mathbf{A}\|}{\|\mathbf{A}\|}\right). \qquad (7)$$

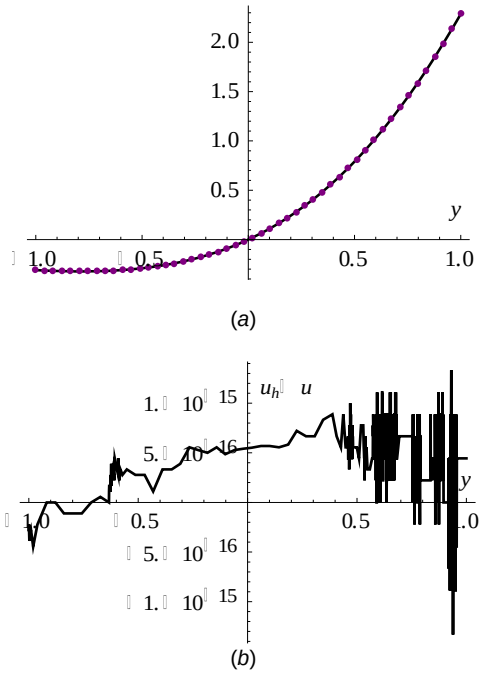


Fig. 3. *Results of solving the integral equation of Example 2 at n = 20 and N_{Gauss} = 16.*

Here $\delta\mathbf{A}$ is the error of the computation of matrix **A**, $\delta\mathbf{\Phi}$ is the error of computing the right-hand side of the equation. This estimate is valid under the following conditions: **A** is a square matrix and $\kappa(\mathbf{A})\|\delta\mathbf{A}\|\cdot\|\mathbf{A}\|^{-1} < 1$. One can see from (7) that a smaller condition number $\kappa(\mathbf{A})$ leads to a smaller error of the SLAE solution. As one can see from a comparison of Tables I and IV, in the case of Example 2, the condition number $\kappa(\mathbf{A})$ is less than in the case of Example 1. The symbol * in Table IV marks a row with the result, which is the best in terms of accuracy.

Fig. 3 shows the numerical results obtained at n = 20, N_{Gauss} = 16. The exact solution is shown in Fig. 3, (*a*) by solid line, and the numerical solution obtained by the proposed method is shown by a dotted line. As one can see in Table IV and Fig. 3, (*b*), the maximum error $|u_h - u|$ does not exceed $1.4\cdot10^{-15}$.

Example 3. Consider (1) at the following expressions for functions $g(x)$, $K(x,s)$, and $f(x)$:

$$g(x)=\cos(x),\quad K(x,s)=\frac{3}{6.4\pi\cos^2\left(\frac{x+s}{2}\right)-1},$$

$$f(x)=g(x)\left(\frac{17}{2}+\frac{128}{17}\cos 2x\right)+\frac{33}{2}-16\sin^2 x-\frac{128}{17}\cos 2x,\quad a=-\pi, b=\pi,\quad \lambda=1.$$

The exact solution has the following form: $u(x)=\frac{17}{2}+\frac{128}{17}\cos 2x$. The given example was considered in detail in [19], where a coincidence with the exact solution was obtained by the computations on the computer BESM-6 in the first six digits of the mantissa of floating-point decimal numbers. Therefore, the numerical solution error reached in [19] was of the order of 10^{-6}. The version of the collocation method proposed in the present work has enabled us to reach the accuracy which is by seven decimal orders higher than the accuracy reached in [19].

TABLE V. ERROR $\|u - u_h\|_C$ AS A FUNCTION OF PARAMETERS N AND N_{GAUSS} IN EXAMPLE 3.

No.	n	N_{Gauss}	$\|u-u_h\|_C$	κ(**A**)	CPU time, sec.
1	9	6	8.404e–1	1.120e3	0.22
2	14	16	2.624e–3	2.309e5	0.31
3	20	16	3.712e–4	5.540e7	0.47
4	30	24	1.115e–7	5.054e8	1.08
5	36	30	5.957e–10	8.151e8	1.14
6	42	35	8.750e–12	5.153e8	1.49
7*	48	40	6.892e-13	8.650e8	2.06
8	56	45	5.301e-12	1.557e9	2.86

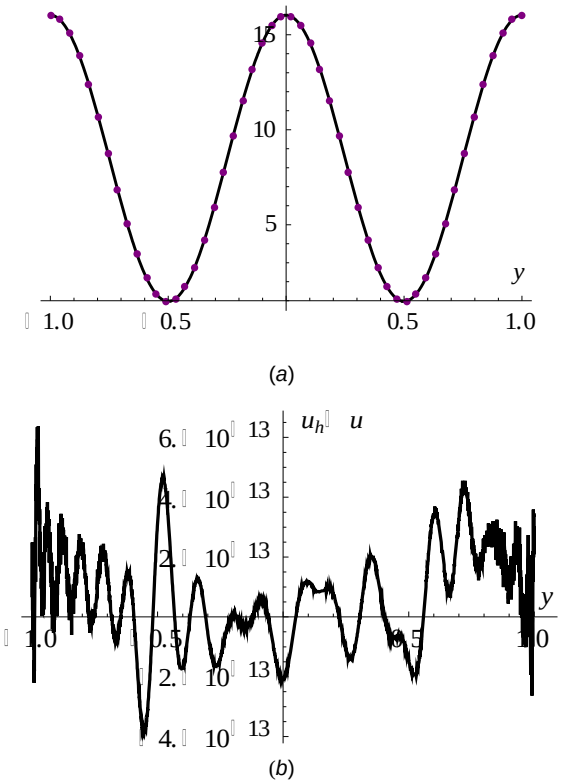


Fig. 4. *Results of solving the integral equation of Example 3 at n = 48 and N_{Gauss} = 40.*

Table V presents the error $\|u - u_h\|_C$, which was computed at 1001 points taken uniformly in the region $-\pi \le x \le \pi$. This example makes increased demands for numerical solution than two foregoing examples, first of all, because of a more complex behavior of the equation kernel. Besides, two periods of the function cos(2*x*) are contained in the interval [–π, π], which leads to an increase in the gradients in comparison with the function sin(*x*), and in Example 2, only part of the period of this function was considered. As one can see in Table V, the condition number of the SLAE corresponding to Example 3 is much larger than in Examples 1 and 2. Symbol * marks in Table V the row with the most accurate result. For the reasons indicated above, the given example required a much

TABLE VI. ERROR $||u - u_h||_C$ AS A FUNCTION OF PARAMETERS N AND N_{Gauss} IN EXAMPLE 4.

No.	n	N_{Gauss}	$\|\|u - u_h\|\|_C$	$\kappa(\mathbf{A})$	CPU time, sec.
1	9	6	8.222e–8	7.571e2	0.14
2	14	16	1.066e–14	5.721e4	0.26
3*	20	16	4.441e–15	1.082e7	0.42
4	30	24	5.329e–15	4.108e8	0.89

TABLE VII. THE ABSOLUTE ERROR $|u(x) - u_h(x)|$ AT PARTICULAR POINTS IN SOLUTION OF EXAMPLE 4 AT $N = 20$, $N_{Gauss} = 16$.

x	Our method	Method in [25]	Method in [26]
0	1.33e-15	6.85e-06	1.65e-04
0.2	6.66e-16	1.02e-05	4.69e-04
0.4	8.88e-16	1.52e-05	3.39e-04
0.6	1.78e-15	2.27e-05	5.31e-04
0.8	2.66e-15	3.39e-05	1.53e-03
1	8.88e-16	5.06e-05	1.10e-03

larger number of collocation points and the nodes of the quadrature formula than in the first two examples to reach a solution accuracy which is comparable in terms of the order of magnitude with the magnitude of rounding errors.

The exact solution is depicted in Fig. 4 (*a*) by a solid line, and the numerical solution obtained by the proposed method is shown by a dotted curve. As one can see in Table V and Fig. 4 (*b*), the error $||u - u_h||_C$ does not exceed $7 \cdot 10^{-13}$.

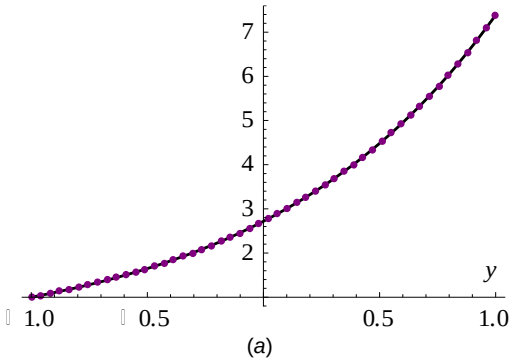
Example 4. The following linear integral equation was considered in the work [25]:

$$u(x) + \frac{1}{3}\int_0^1 e^{2x-\frac{5}{3}s} u(s)ds = e^{2x+\frac{1}{3}}, \quad x \in [0,1]. \qquad (8)$$

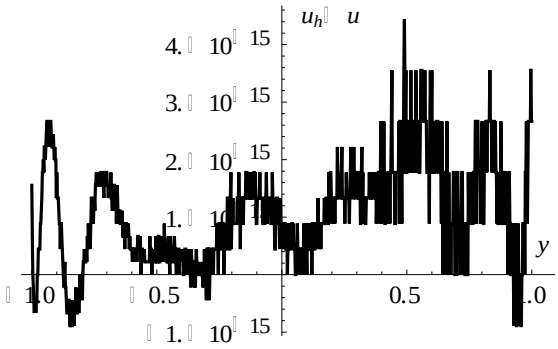
Consequently, one must set in (1) $g(x) = 1$, $\lambda = -\frac{1}{3}$, $a = 0$, $b = 1$, $K(x,s) = e^{2x-\frac{5}{3}s}$, $f(x) = e^{2x+\frac{1}{3}}$. We assume further in (3): $x_c = \frac{1}{2}$, $h = \frac{1}{2}$. The exact solution of (8) is as follows: $u(x) = e^{2x}$.

Table VI presents the error $||u - u_h||_C$, which was computed at 1001 points taken uniformly in the region $0 \le x \le 1$. One can see in this table that the obtaining of a high-accuracy numerical solution of the example under consideration by the method proposed here has required the use of fairly modest means, namely: the number of collocation points $n = 14$ and the number of nodes $N_{Gauss} = 16$ in the Gauss quadrature formula (see row 2 in Table VI, from which one can see that the accuracy of the result has been achieved, which is approximately equal to the machine arithmetic accuracy). Symbol * marks in Table VI the row with the most accurate result.

Table VII presents a comparison of the value of the absolute error $|u(x) - u_h(x)|$ in the solution of equation



(*a*)



(*b*)

Fig. 5. *Results of solving the integral equation of Example 4 at n = 20 and N_{Gauss} = 16.*

of Example 4 by the proposed method, by the spline-collocation method from the work [25], and by the method from the work [26], in which the triangular orthogonal functions were used. One can see from this table that the method proposed here ensures an error, which is by 9 – 11 decimal orders less than in the case of method [25].

We believe that there are the following two main reasons for a lower accuracy of method [25] in comparison with our method: (i) in the method [25], an approximating polynomial only of the third degree is used in each cell of a uniform grid; (ii) at the endpoints $x = 0$ and $x = 1$, the boundary conditions $s''(0) = 0$, $s''(1) = 0$ are used for the closure of the algorithm for constructing a cubic B-spline $s(x)$. At the same time, $u''(0) = 4$, $u''(1) = 4e^2 \approx 29.5562$ in accordance with the exact solution. Note that the boundary conditions $s''(0) = 0$, $s''(1) = 0$ are not required in the original formulation of the problem of solving equation (8) and, thus, are the artificial additional conditions, which are far from the real behavior of the solution at endpoints.

At the same time, the method proposed here does not need any additional boundary conditions for its implementation. Owing to this, it involves a potential possibility of obtaining the numerical solution with a high accuracy.

The exact solution is depicted in Fig. 5 (*a*) by a solid line, and the numerical solution obtained by the proposed method is shown by a dotted curve. As one can see in Table VI and Fig. 5 (*b*), the error $||u - u_h||_C$ does not exceed $5 \cdot 10^{-15}$.

Example 5. Consider the following nonlinear integral equation [25]:

TABLE VIII. ERROR $||U - U_h||_C$ AS A FUNCTION OF PARAMETERS N AND N_{GAUSS} IN EXAMPLE 5.

No.	n	N_{Gauss}	N_{it}	$\|\|u - u_h\|\|_C$	$\kappa(\mathbf{A})$	CPU time, sec.
1	9	6	6	3.541e–5	5.420e2	0.22
2	14	16	5	2.097e–12	4.231e4	0.36
3*	20	16	10	7.772e–16	8.250e6	0.87
4	30	24	11	7.772e–16	6.488e8	2.69

$$u(x) = \sin(\pi x) + \frac{1}{5}\int_0^1 \cos(\pi x)\sin(\pi s)u^3(s)ds, \quad x \in [0,1]. \quad (9)$$

Thus, $a = 0$, $b = 1$, $\lambda = 1/5$, $f(x) = \sin(\pi x)$ in (2). In the work [4], the existence and uniqueness theorems were proved for a slightly more general integral equation

$$u(x) = \lambda\int_a^b K(x,s)u(s)^p ds + f(x), \quad p \geq 2.$$

Let us make in (2) the substitution $x = x_c + hy$, where $x_c = (a + b)/2$ and $h = (b - a)/2$, and let $U(y) = u(x_c + hy)$. Equation (2) then takes the form

$$U(y) - \lambda h\int_{-1}^1 G(x_c + hy, x_c + h\xi, U(\xi))d\xi = F(y), \quad (10)$$

where $F(y) = f(x_c + hy)$.

At the solution of nonlinear integral equations, one has to use the iterative methods. Let $U^m(y)$, m = 0,1,2,…, be the sequential approximations of the solution of (10). In the case of (9), we take $U^0(y) = 0$. Then we immediately obtain from (9) the next approximation $U^1(y) = \sin[\pi(1 + y)/2]$.

To reduce (9) to a linear equation we now linearize the expression $[U^{m+1}(\xi)]^3$. Take the function $\varphi(z) = z^3$. Retaining only the first two terms in the Taylor series expansion of $\varphi(z)$, we obtain

$$\varphi(z) \approx \varphi(z_0) + \varphi'(z_0)(z - z_0). \quad (11)$$

Let us substitute in (11) the values $z_0 = U^m$, $z = U^{m+1}$. We obtain the following linear expression from (11):

$$(U^{m+1})^3 \approx (U^m)^2(3U^{m+1} - 2U^m).$$

Substituting this expression in the integrand in (10) instead of $U(\xi)$, we obtain a linear integral equation for the solution of which we use (at each iteration) the same collocation method as in the foregoing four examples.

Denote the approximate solution obtained at the mth iteration (m = 2,3,…) as follows:

$$U^m(y) = \sum_{k=1}^n b_k^{(m)} y^{k-1}.$$

For the realization of the criterion for completion of iterations we introduce the quantity

$$\delta b^{m+1} = \max_{1 \leq k \leq n} \left| b_k^{(m+1)} - b_k^{(m)} \right|.$$

As the numerical experiments have shown, the quantity δb^{m+1} at first reduced with the increasing number of iterations m, however, at some $m = m_c > 1$, the inequality $\delta b^{m_c} < \delta b^{m_c+1}$ took place. As soon as this inequality arose, the iterations were stopped, and the solution obtained at the m_cth iteration was taken as the final numerical solution. Such a procedure for determining the moment of the iterations completion is termed the Garvik technique [34].

TABLE IX. THE ABSOLUTE ERROR $|U(x) - U_h(x)|$ AT PARTICULAR POINTS IN SOLUTION OF EXAMPLE 5 AT N = 20, N_{GAUSS} = 16.

x	Our method	Method in [25]	Method in [39]
0	6.66e-16	0	4.98e-02
0.2	4.44e-16	1.56e-07	4.03e-02
0.4	5.55e-16	5.99e-08	1.54e-02
0.6	0.0	6.90e-08	1.54e-02
0.8	4.44e-16	1.57e-07	4.03e-02
1	1.80e-16	1.94e-07	4.98e-02

The exact solution of (9) is as follows [25]:

$$u(x) = \sin(\pi x) + \frac{1}{3}\left(20 - \sqrt{391}\right)\cos(\pi x).$$

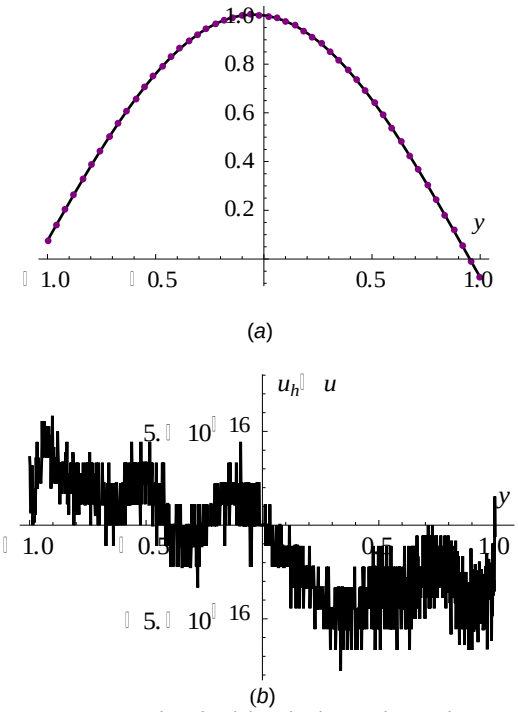


Fig. 6. *Results of solving the integral equation of Example 5 at n = 20 and* N_{Gauss} *= 16.*

In Table VIII, N_{it} is in fact the iteration number m_c. Symbol * marks in Table VIII the row with the most accurate result.

Table IX shows that the above-described version of the iteration-collocation method for solving the Fredholm nonlinear second-kind integral equation ensures the numerical solution error, which is by 8 – 9 decimal orders less than the error obtained by the spline-collocation method [25].

The exact solution is depicted in Fig. 6(*a*) by a solid line, and the numerical solution obtained by the proposed method is shown by a dotted curve.

Example 6. Consider the following nonlinear integral equation [40]:

$$u(x) = f(x) + \int_0^1 \frac{1}{x+s+u(s)}\,ds, \quad x \in [0,1]. \qquad (12)$$

The function $f(x)$ must be chosen in such a way that the function $u(x) = (x + 1)^{-1}$ satisfies the integral equation (12). The corresponding expression for $f(x)$ was found with the aid of the *Mathematica* system:

$$F(y) = \left(\begin{array}{l} \left(-2(-3+2y+y^2)\,\mathrm{Arctan}\left[\dfrac{y-5}{3\sqrt{15+2y-y^2}} \right] + \right. \\ \left. \sqrt{15+2y-y^2}\left(4-(3+y)\,\mathrm{Log}\left[\dfrac{2(4+y)}{3+y} \right] \right) \right) / \\ \left(2(3+y)\sqrt{15+2y-y^2} \right), \end{array} \right.$$

where $F(y)=f(x_c + hy)$, x_c = 1/2, and h = 1/2. The initial approximation $U^0(y)$ was obtained as follows: we have deleted the integral in (12), and as a result, $U^0(y) = F(y)$. The next approximations of the solution were computed by solving the linear equation (m = 0,1, 2,…)

$$U^{m+1}(y) = F(y) + h\int_{-1}^{1} \frac{1}{0.5(y+\xi+2)+U^m(\xi)}\,d\xi,\ y \in [-1,1].$$

In Table X, symbol * marks the row with the most accurate result. Table XI shows that the above-described version of the iteration-collocation method for solving the Fredholm nonlinear second-kind integral equation ensures the numerical solution error, which is by 7 – 8 decimal orders less than the error obtained by the extrapolation method [40].

The exact solution is depicted in Fig. 7(*a*) by a solid line, and the numerical solution obtained by the proposed method is shown by a dotted curve.

One can draw the following further conclusions from the analysis of the results of solving all examples: 1) the remainder term of solution approximation and the solution accuracy in the proposed numerical technique as well as in many other techniques applied in practice depends on solution gradients; 2) the ultimate attainable accuracy of the solution obtained on computer depends as in many other numerical methods also on the condition number of a SLAE to the solution of which the construction of the problem approximate solution reduces; 3) the attainable minimum size of the solution error depends on the both above factors. One can see from the analysis of the solution error graphs in all examples that when the solution error is close in terms of its order of magnitude to the computer rounding error, the oscillations appear on the graph. Their amplitude correlates with the magnitude of solution gradients. For example, the amplitude of the error oscillations in Example 2 is larger near the right end of the interval than near the left end. Their amplitude in Example 3 is much higher than in Example 2. The result presented in row 8 of Table V shows that at a further increase of the nodes number in the Gauss formula and the number of collocation

TABLE X. ERROR $\|U - U_h\|_C$ AS A FUNCTION OF PARAMETERS N AND N_{GAUSS} IN EXAMPLE 6.

No.	n	N_{Gauss}	N_{it}	$\|u - u_h\|_C$	$\kappa(\mathbf{A})$	CPU time, sec.
1	9	6	25	1.878e–6	1.298e3	0.12
2	14	16	37	7.765e–11	4.153e4	0.33
3	20	16	43	2.109e–15	8.126e6	0.52
4*	30	24	43	4.441e–16	3.105e8	1.17

TABLE XI. THE ABSOLUTE ERROR $|U(x) - U_h(x)|$ AT PARTICULAR POINTS IN SOLUTION OF EXAMPLE 6 AT N = 30, N_{GAUSS} = 24.

x	Our method	Method in [40]
0	2.22e-16	2.20e-8
1/6	3.33e-16	1.89e-8
1/4	3.33e-16	1.60e-8
1/3	0.0	1.40e-8
1/2	0.0	1.03e-8
2/3	1.11e-16	7.00e-9
3/4	1.11e-16	5.57e-9
5/6	1.11e-16	4.55e-9
1	3.33e-16	3.00e-9

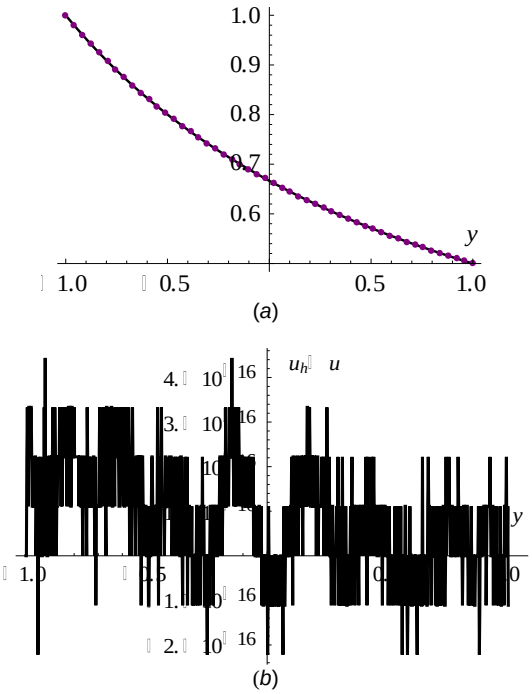


Fig. 7. *Results of solving the integral equation of Example 6 at n = 30 and* N_{Gauss} *= 24.*

points, there occurs such an accumulation of round-off errors that the numerical result is no longer refined.

The results obtained in the present work show the flexibility and capabilities of the proposed method: at the expense of a suitable choice of the approximant degree, the number of collocation points, and the way of their location in the integration interval, one can obtain a very high accuracy of the result, whose error

TABLE XII. THE RESIDUAL NORM $||E(x)||_C$ AS A FUNCTION OF PARAMETERS N AND N_{Gauss} IN EXAMPLE 3.

No.	n	N_{Gauss}	$\|\|\varepsilon(x)\|\|_C$
1	9	6	4.976e-1
2	14	16	2.707e-3
3	20	16	8.997e-5
4	30	24	1.115e-7
5	36	30	3.673e-11
6	42	35	1.528e-11
7*	48	40	1.517e-11
8	56	45	1.543e-11

is close in terms of its order of magnitude to the error of machine computations in the arithmetic of floating-point numbers in the *Mathematica* system.

IV. NUMERICAL SOLUTION VERIFICATION IN THE CASE OF THE UNKNOWN EXACT SOLUTION

We now consider a question about how one can judge approximately about the error of the numerical solution obtained by the method under consideration in the case when the exact solution of the linear equation (1) is unknown. Let us rewrite this equation in the form

$$Lu = g(x)u(x) - \lambda\int_a^b K(x,s)u(s)ds - f(x) = 0,\ x \in [a,b]. \quad (13)$$

At the substitution of the approximate solution $u_h(x)$ in (13), a nonzero residual $\varepsilon(x)$ arises in the right-hand side: $Lu_h(x) = \varepsilon(x)$. If this residual proves to be sufficiently small for all $x \in [a,b]$ and, in addition, its modulus will reduce with increasing degree $n - 1$ of the polynomial approximating the solution then this will testify both the accuracy of the obtained numerical solution and its convergence to the exact solution with increasing n. Subtracting from the both sides of the equation $Lu_h(x) = \varepsilon(x)$ the both sides of the equation $Lu(x) = 0$, it is easy to obtain the following equality: $L(u_h - u) = \varepsilon(x)$. From this we have: $||u - u_h|| \le ||L^{-1}||\cdot||\varepsilon||$. If the value $||L^{-1}||$ is bounded from above, then one can judge approximately from here about the magnitude of $||u - u_h||$ depending on the size of the residual ε. We illustrate the above-described procedure by Example 3. As one can see in Table XII, with the growth of n within the interval $9 \le n \le 48$, a rapid reduction of the value $||\varepsilon(x)||_C$ occurs, which is indicative of the convergence of the numerical solution to the exact one and confirms the correctness of the obtained approximate solution. But at $n = 56$, the residual value is already somewhat larger than at $n = 48$. This agrees with the fact that $||u - u_h||_C$ at this n is larger than at $n = 48$, see Table V. Fig. 8 shows the form of curves $\varepsilon = \varepsilon(x)$ for two pairs of the values (n, N_{Gauss}).

In the case of the nonlinear integral equation (2), we can investigate the behavior of the residual function

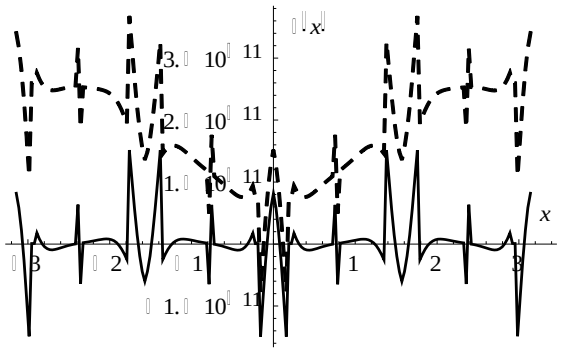


Fig. 8. The residual (Example 3) at n = 48, N_{Gauss} = 40 (solid line) *and at n = 36, N_{Gauss} = 30 (dashed line).*

$$\varepsilon(x) = u(x) - \lambda\int_a^b G(x,s,u_h(s))ds - f(x)$$

similarly to the linear case and compose a table similar to Table XII, and in this way, we can draw a conclusion whether the obtained approximate solution is correct. We omit these studies for the sake of brevity.

V. CONCLUSIONS

A p-version of the collocation method has been proposed for the numerical solution of the Fredholm integral equations of the second kind. The important peculiarities of the method are the possibilities for a considerable variation of the polynomial degree in the polynomial representation of the approximate solution and a considerable variation of the number of nodes of the Gauss quadrature formula for reaching a high accuracy of the solutions of equations.

The proposed p-version of the collocation method has been implemented in the language of the system *Mathematica*. The corresponding computer code proved to be sufficiently compact, and it has been shown here that it can ensure quite well a sufficient accuracy for a wide class of the Fredholm integral equations of the second kind. In all considered examples, the proposed version of the collocation method has enabled us to reach the accuracy close to the size of machine rounding errors. The method has proved economical from the viewpoint of the computer resources required for its realization: even in the most complex example from the examples handled in the work — Example 3, the CPU time needed for problem solution did not exceed 3 seconds of the computer work to solve the equation with an error close to the rounding errors of computer calculations.

The capabilities of the method have been shown by the examples of solving the specific equations. An algorithm has also been described here, which enables one to verify the accuracy of the approximate solution obtained by the proposed version of the collocation method.

Comparison of the above numerical solutions of the known linear and nonlinear integral equations with the known results obtained by other methods showed that the possibilities of the collocation method are far from exhausted.

REFERENCES

[1] U. van Rienen, Numerical Methods in Computational Electrodynamics. Berlin: Springer Verlag, 2013.

[2] A.B. Samokhin, Volumetric Integral Equations: Methods and Algorithms. Moscow: MIREA, 2011 [in Russian].

[3] J. Nee, "Nonlinear integral equation from the BCS gap equations of superconductivity", Nonlinear Anal. Real World Appl., vol. 11, pp. 190-197, 2010.

[4] J.A. Ezquerro and M.A. Hernandez-Veron, "On the existence of solutions of nonlinear Fredholm integral equations from Kantorovich's technique", Algorithms, vol. 10, No. 89, 2017; doi:10.3390/a10030089.

[5] M.M. Lavrentiev, A.V. Avdeev, M.M. Lavrentiev, Jr. and V.I. Priimenko, Inverse Problems of Mathematical Physics. Utrecht, Boston: VSP, 2003.

[6] T. Habashy, "Hybrid finite-difference integral equation solver for 3D frequency domain anisotropic electromagnetic problems," Geophysics, vol. 76 (2). Pp. F123-F137, 2011.

[7] A.N. Tikhonov, V.Ya. Arsenin, and A.A. Timonov, Mathematical Problems of Computer Tomography. Moscow: Nauka, 1987 [in Russian].

[8] K. Maleknejad and M. Alizadeh, "An efficient numerical scheme for solving Hammerstein integral equation arisen in chemical phenomenon," Procedia CS, vol. 3, pp. 361-364, 2011.

[9] O. Diekmann, "Thresholds and traveling for the geographical spread of infection," J. Math. Biol., vol. 6, pp. 109-130, 1978.

[10] W.K. Pratt, Digital Image Processing. 3rd ed. New York: Wiley-Interscience, 2001.

[11] I.G. Graham, "Galerkin methods for second kind integral equations with singularities," Math. Comp., vol. 39), pp. 519-533, 1982.

[12] C.A. Micchelli, Y. Xu, and Y. Zhao, "Wavelet Galerkin methods for second-kind integral equations," J. Comput. Appl. Math., vol. 86, pp. 251-270, 1997.

[13] Z. Chen,Y. Xu, and J. Zhao, "The discrete Petrov-Galerkin method for weakly singular integral equations," J. Integral Equations Appl., vol. 11, pp. 1-35, 1999.

[14] M. Huang, Wavelet Petrov-Galerkin Algorithms for Fredholm Integral Equations of the Second Kind. PhD thesis, Academy Science of China, 2003.

[15] W. Fang, Y. Wang, and Y. Xu, "An implementation of fast wavelet Galerkin methods for integral equations of the second kind," J. Sci. Comput., vol. 2, pp. 277-302, 2004.

[16] H. Cai and Y. Xu, "A fast Fourier-Galerkin method for solving singular boundary integral equations," SIAM J. Numer.Anal., vol. 46, pp. 1965-1984, 2008.

[17] G. Long, G. Nelakanti, and X. Zhang, "Iterated fast multiscale Galerkin methods for Fredholm integral equations of second kind with weakly singular kernels," Appl. Numer. Math., vol. 62, pp. 201-211, 2012.

[18] D. Mirzaei and M. Dehghan, "A meshless based method for solution of integral equations," Appl.Numer.Math., vol. 60, pp. 245-262, 2010.

[19] V.S. Verlan and V.S. Sizikov, Integral Equations: Methods Algorithms Programs. Kiev: Naukova Dumka, 1986 [in Russian].

[20] R.D. Russell and L.F. Shampine, "A collocation method for boundary value problems," Numerical Math., vol. 19, pp. 1-28, 1972.

[21] U.M. Ascher, R.M.M. Mattheij, and R.D. Russell, Numerical Solution of Boundary Value Problems for Ordinary Differential Equations. Philadelphia: SIAM, 1994.

[22] U. Ascher, J. Christiansen, and R.D. Russell, "Collocation software for boundary-value ODEs," ACM Trans. Math. Software, vol. 7 (2), pp. 209-222, 1981.

[23] Z. Chen, C.A. Micchelli, and Y.Xu, "Fast collocation methods for second kind integral equations," SIAM J. Numer. Anal., vol. 40, pp. 344-375, 2002.

[24] J. Rashidinia and M. Zarebnia, "Numerical solution of linear integral equations by using Sinc-collocation method," Appl. Math. Comput., vol. 168, pp. 806-822, 2005.

[25] N. Ebrahimi and J. Rashidinia, "Collocation method for linear and nonlinear Fredholm and Volterra integral equations," Appl. Math. Comput., vol. 270, pp. 156-164, 2015.

[26] E. Babolian, H.R. Marzban, and M. Salmani, "Using triangular orthogonal functions for solving Fredholm integral equations of the second kind," Appl. Math. Comput., vol. 201, pp. 452-464, 2008.

[27] V. Shapeev, "Collocation and least residuals method and its applications," EPJ Web of Conferences, vol. 108, No. 01009, 2016. DOI: 10.1051/epjconf/201610801009).

[28] V.I. Isaev and V.P. Shapeev, "High-accuracy versions of the collocations and least squares method for the numerical solution of the Navier-Stokes equations," Comput. Math. Math. Phys., vol. 50 (10), pp. 1670-1681, 2010.

[29] V.P. Shapeev, E.V. Vorozhtsov, V.I. Isaev, and S.V. Idimeshev, "The method of collocations and least residuals for three-dimensional Navier-Stokes equations," Vychisl. Metody Programm., vol. 14, pp. 306-322, 2013.

[30] E.V. Vorozhtsov and V.P. Shapeev, "On combining the techniques for convergence

acceleration of iteration processes during the numerical solution of Navier-Stokes equations," Vychisl. Metody Programm., vol. 18, pp. 80-102, 2017.

[31] D. Kharenko, C. Padovani, A. Pagni, G. Pasquinelli, and L. Semin, "Free longitudinal vibrations of bimodular beams: a comparative study," Int. J. Structural Stability and Dynamics, vol. 11 (1), pp. 23-56, 2011.

[32] K.E. Atkinson, The Numerical Solution of Integral Equations of the Second kind. Cambridge: Cambridge University Press, 1997.

[33] S. Wolfram, The Mathematica book, 5th ed. Champaign, IL: Wolfram Media, Inc., 2003.

[34] N.N. Kalitkin. Numerical Methods, 2nd ed. St. Petersburg: BKhV-Petersburg, 2011.

[35] A. Hammerstein, Nichtlineare Integralgleichungen nebst Anwendungen, Acta Math., vol. 54, No. 1, pp. 117-176, 1930.

[36] P. Knupp and K. Salari, Verification of Computer Codes in Computational Science and Engineering. Boca Raton: Chapman and Hall/CRC, 2002.

[37] P.J. Roache, "Code verification by the method of manufactured solutions," J. Fluids Eng., vol. 124, pp. 4-10, 2002.

[38] I.B. Petrov and A.I. Lobanov, Lectures on Computational Mathematics. Moscow: Binom, 2006 [in Russian].

[39] J. Saberi-Najafi and M. Heidari, "Solving nonlinear integral equations in the Urysohn form by Newton–Kantorovich-quadrature method," Comput. Math. Appl., vol. 60, pp. 2058-2065, 2010.

[40] J. Farzi, "Generalized extrapolation methods for solving nonlinear Fredholm integral equations," Math. Communs., vol. 19, pp. 375-390, 2014.