

Research Article

An Adaptive Approach to Solutions of Fredholm Integral Equations of the Second Kind

Nebiye Korkmaz and Zekeriya Güney

Department of Secondary Science and Mathematics Education, Muğla Sıtkı Koçman University, 48000 Muğla, Turkey

Correspondence should be addressed to Nebiye Korkmaz; nkorkmaz@mu.edu.tr

Received 5 May 2014; Revised 13 September 2014; Accepted 22 September 2014; Published 17 November 2014

Academic Editor: Chun-Gang Zhu

Copyright © 2014 N. Korkmaz and Z. Güney. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As an approach to approximate solutions of Fredholm integral equations of the second kind, adaptive hp-refinement is used firstly together with Galerkin method and with Sloan iteration method which is applied to Galerkin method solution. The linear hat functions and modified integrated Legendre polynomials are used as basis functions for the approximations. The most appropriate refinement is determined by an optimization problem given by Demkowicz, 2007. During the calculations L^2 -projections of approximate solutions on four different meshes which could occur between coarse mesh and fine mesh are calculated. Depending on the error values, these procedures could be repeated consecutively or different meshes could be used in order to decrease the error values.

1. Introduction and Preliminaries

The aim of this study is to find approximate solutions for the Fredholm integral equations of the second kind by applying adaptive refinement together with Galerkin method and Sloan iteration method. Our reason to apply adaptive refinement is to search for meshes on which we might obtain better approximations to the solution of the problems with the methods mentioned. In Section 2 we explained how to obtain a finer mesh from a given mesh, which is called coarse mesh, and how we construct the basis functions used for the approximation. In Section 3 we solved the problem given with (10) by Galerkin method and then in order to determine an optimal mesh we solved the optimization problem given with (18) for adaptive refinement. In Section 4, as the subsequent step, we iterate Galerkin method solution by Sloan iteration method and we solved the same optimization problem for this case in order to make adaptive refinement. Finally in Section 5 we presented some problem examples. In this section we will give some basic knowledge about two essential subjects that this study stands on: integral equations and adaptive refinement.

1.1. On Integral Equations. As the theory of integral equations has significant importance in mathematics, it is also closely

related to various fields of science. Many problems, such as ordinary and partial differential equations, problems of mathematical physics, can be laid out as integral equations. Hochstadt [1] mentioned that many existence and uniqueness results can be derived from the corresponding results from integral equations and there is almost no area of applied mathematics and mathematical physics where integral equations do not play a role. Many studies can be found that state some of these areas of usage of integral equations. Rahbar and Hashemizadeh [2] indicated to high applicability of integral equations in different areas of applied mathematics, physics, and engineering, and they particularly mentioned some areas in which these equations are widely used such as mechanics, geophysics, electricity and magnetism, kinetic theory of gases, hereditary phenomena in biology, quantum mechanics, mathematical economics, and queuing theory. We can sort some more examples of these areas of usage as follows: automatic control theory, network theory and the dynamics of nuclear reactors [3], acoustics, optics and laser theory, potential theory, radiative transfer theory, cardiology, fluid mechanics and statics [4], continuum mechanics, hereditary phenomena in physics and biology, renewal theory, radiation, optimization, optimal control systems, communication theory, population genetics, medicine and

mathematical problems of radiative equilibrium, the particle transport problems of astrophysics and reactor theory, steady state heat conduction, and fracture mechanics [5].

As Pachpatte [3] expressed, the beginning of the integral equations can be traced back to N. H. Abel who found an integral equation in 1812 starting from a problem in mechanics and in 1895 V. Volterra emphasized the significance of the theory of integral equations.

Lonseth [6] stated that Fredholm published his distinguished paper in Acta Mathematica [7] in 1903, in which he gave the first detailed account of the existence and multiplicity of solutions of the following two equations where the kernel $K(s, t)$ and $y(s)$ are known functions and the values of λ (proper values) are values to be determined such that a continuous solution $x(s) \neq 0$ exists:

$$\begin{aligned} x(s) - \int_0^1 K(s, t) x(t) dt &= y(s), \quad 0 \leq s \leq 1, \\ x(s) - \lambda \int_0^1 K(s, t) x(t) dt &= y(s), \quad 0 \leq s \leq 1. \end{aligned} \quad (1)$$

In his book Pachpatte [3] stated a few number of monographs that he accepted as an excellent account of integral equations may be found in the following: Burton [8], Corduneanu [9–11], Gripenberg et al. [12], Krasnoselskii [13], Miller [14], and Tricomi [15].

1.2. On Adaptive Refinement. In 1988 Babuška [16] published his work on the advances in the p and h - p versions of the finite element method and in this study he distinguished three versions of the finite element method (FEM) as follows: the h -version, the p -version, and the h - p version. The main idea in the h -version is to refine the size of the meshes while degrees of the polynomials used for approximation are kept fixed (usually $p = 1, 2$); in the p -version it is the opposite: size of the meshes kept fixed, but degrees of the polynomials used for approximation are increased. In the h - p version both changes are done simultaneously: size of the meshes are refined and the degrees of the polynomials used for approximation are increased. In [16] while the h -version of FEM is introduced to be the standard one, the other versions are said to be developed later and the first theoretical papers about the p -version and the h - p version which appeared in 1981 are given with [17] and [18], respectively.

In Demkowicz's book [19] which is about computation with hp -adaptive finite elements, he studied one- and two-dimensional elliptic and Maxwell problems and he mentioned two major components of the one-dimensional version of their hp -algorithm as fine grid solution and optimal mesh selection. For the first component, a given (coarse) mesh is refined in both h and p to obtain a corresponding fine mesh, and then the problem is solved on this fine mesh to find the fine mesh solution. In the latter component, he used this fine mesh solution to determine optimal mesh refinement of the coarse mesh, by minimizing the projection based interpolation error solving the following discrete optimization problem where u , $\Pi_{hp}u$, $\Pi_{hp_{opt}}u$, $N_{hp_{opt}}$, and N_{hp} denote,

respectively, the solution on the fine mesh, the interpolant of the fine grid solution on the original mesh, interpolant of the fine grid solution on the new optimal mesh to be determined, the corresponding number of degrees of freedom on the new optimal mesh to be determined and the corresponding number of degrees of freedom on the original mesh:

$$\frac{\|u - \Pi_{hp}u\|_{H^1(0,l)}^2 - \|u - \Pi_{hp_{opt}}u\|_{H^1(0,l)}^2}{N_{hp_{opt}} - N_{hp}} \longrightarrow \min. \quad (2)$$

Here the aim of the optimization problem is said to maximize the rate of decrease of the interpolation error. Asadzadeh and Eriksson [20] gave a few number of references [21–25] on solving integral equations with FEM in their paper in which they have chosen to work on the single layer potential problem for Laplace's equation with Neumann boundary conditions in order to be concrete. In their paper, the studies on solving integral equations with adaptive FEM in that period are given with [25–28]. Adaptive FEM are usually used to solve partial differential equations; but in literature these methods are also seen to be used for solving different type of problems in various branches of science such as hydrodynamics [29], optimal design [30], elliptic stochastic equations [31], parabolic problems [32], parabolic systems [33], elliptic problems [34], elliptic partial differential equations [35], elliptic boundary value problems [36, 37], electrostatics [38], electromagnetic problems [39], biological flows [40], and Laplace eigenvalue problem [41].

2. Refining a Finer Mesh from a Coarse Mesh and Construction of Basis Functions

2.1. Refining a Finer Mesh from a Coarse Mesh. Let $a = t_1 < t_2 < \dots < t_n = b$ be the node points of a given (finite element) mesh which is accepted as the coarse mesh and denote the list of these node points as follows:

$$L^c = [t_1 \ t_2 \ \dots \ t_n]. \quad (3)$$

For each $i \in \{1, 2, \dots, n-1\}$ the interval of the form $I_i = [t_i, t_{i+1}]$ is called an element (or finite element) having two parameters: element length $h_i = t_{i+1} - t_i$ and element local polynomial of order of approximation p_i ($p_i \geq 2$). What is meant by *element local polynomial of order of approximation* can be explained as follows: "If an element has element local polynomial of order of approximation of order p , it means the nonlinear base functions on that element are the polynomials of degree starting from 2 to p ." Let D^c be the list of $n-1$ numbers of element local polynomial order of approximation of the coarse mesh elements:

$$D^c = [p_1 \ p_2 \ \dots \ p_{n-1}]. \quad (4)$$

Firstly dividing each element of this mesh from the middle (h -refinement) and then increasing the element local polynomial order of approximation by 1 for each new element (p -refinement), we obtain a finer mesh having new lists of node points and the element local polynomial orders of approximation given below which are of length $2n-1$ and $2n-2$, respectively:

$$L^f = \left[t_1 \quad \frac{(t_1 + t_2)}{2} \quad t_2 \quad \cdots \quad t_{n-1} \quad \frac{(t_{n-1} + t_n)}{2} \quad t_n \right], \quad (5)$$

$$D^f = [p_1 + 1 \quad p_1 + 1 \quad p_2 + 1 \quad p_2 + 1 \quad \cdots \quad p_{n-1} + 1 \quad p_{n-1} + 1]. \quad (6)$$

2.2. Construction of Basis Functions. In this study, for each element of a mesh two kinds of basis functions are used: hat functions and bubble functions. The reasons why these are called so can be explained as follows: the linear base functions are called hat functions, because their shapes look like a hat and the nonlinear ones are called bubble functions, because they vanish at node points as bubbles. Considering the coarse mesh given with lists (3) and (4) we explain how to construct the basis functions on the coarse mesh as sample. Basis functions of any mesh can be built up similarly. Formulations of number of n hat functions belonging to coarse mesh are as follows:

$$\begin{aligned} \varphi_1(t) &= \begin{cases} \frac{(t_2 - t)}{(t_2 - t_1)}, & \text{if } t \in [t_1, t_2] \\ 0, & \text{otherwise,} \end{cases} \\ \varphi_i(t) &= \begin{cases} \frac{(t - t_i)}{(t_{i+1} - t_i)}, & \text{if } t \in [t_{i-1}, t_i] \\ \frac{(t_{i+1} - t)}{(t_{i+1} - t_i)}, & \text{if } t \in [t_i, t_{i+1}] \\ 0, & \text{otherwise} \end{cases} \quad (7) \\ &\quad i = 2, \dots, n-1, \\ \varphi_n(t) &= \begin{cases} \frac{(t - t_{n-1})}{(t_n - t_{n-1})}, & \text{if } t \in [t_{n-1}, t_n] \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

For the construction of bubble functions in the coarse mesh the integrated Legendre polynomials are combined with a function ψ mapping any closed interval (which is an element) $[t_i, t_{i+1}]$ to the domain of integrated Legendre polynomials that is given as follows:

$$\psi : [t_i, t_{i+1}] \longrightarrow [-1, 1], \quad \psi(t) = \frac{2t - (t_{i+1} + t_i)}{t_{i+1} - t_i}. \quad (8)$$

Let L_p denote the integrated Legendre polynomial of degree p . Bubble function of degree p in the i th ($i = 1, 2, \dots, n-1$) element that is used for approximation is defined as follows:

$$\begin{aligned} \varphi_{i,p}(t) &= \begin{cases} L_p \circ \psi(t), & \text{if } t \in [t_{i-1}, t_i] \\ 0, & \text{otherwise} \end{cases} \quad (9) \\ &\quad i = 2, \dots, n, \quad p \geq 2. \end{aligned}$$

3. Galerkin Method Solution and Adaptive Refinement by Using Demkowicz's Optimization

Firstly we calculate the Galerkin method approximate solution on the fine mesh L^f which was constructed in Section 2.

Then in order to decide the optimal mesh, we solve an optimization problem, given by Demkowicz [19], on each element of the fine mesh. For this we need L^2 -projections of the fine mesh solution on each element of L^f onto the corresponding element of the coarse mesh L^c and on the four possible optimal mesh refinements of the coarse mesh element. The possible four optimal mesh refinements are defined clearly in the latter parts.

3.1. Galerkin Method Solution on the Fine Mesh L^f . Consider the Fredholm integral equation of the second kind:

$$\lambda x(t) - \int_a^b K(t, s) x(s) ds = f(t), \quad a \leq t \leq b, \quad \lambda \neq 0, \quad (10)$$

and the fine mesh having lists (5) and (6). On this mesh the total number of basis functions is $2n-1+2(p_1+p_2+\dots+p_{n-1})$ which we denote by d_f . Let

$$u(t) = \sum_{j=1}^{d_f} c_j \phi_j(t), \quad t \in [a, b], \quad (11)$$

be the approximate solution we are looking for where $\phi_j(t)$ are the basis functions (hat functions and bubble functions) and $c_j(t)$ are the coefficients to be calculated for $j = 1, 2, \dots, d_f$. Substituting (10) in the residual function of the Galerkin method [42], we obtain the following equality:

$$r(t) = \sum_{j=1}^{d_f} c_j \left\{ \lambda \phi_j(t) - \int_D K(t, s) \phi_j(s) ds \right\} - y(t), \quad t \in D. \quad (12)$$

The residual function is required to satisfy the following equalities:

$$\langle r, \phi_i \rangle = 0, \quad i = 1, \dots, d_f. \quad (13)$$

Rearranging (13), for all $i \in \{1, \dots, d_f\}$

$$\begin{aligned} \sum_{j=1}^{d_f} c_j \left[\lambda \int_a^b \phi_j(t) \phi_i(t) dt - \int_a^b \int_a^b K(t, s) \phi_j(s) \phi_i(t) ds dt \right] \\ = \int_a^b f(t) \phi_i(t) dt, \end{aligned} \quad (14)$$

is obtained which is a system of equations and this system can be represented in the matrix form by using the matrices defined as follows:

$$\begin{aligned}
E &= \begin{bmatrix} E_{11} & \cdots & E_{d_f 1} \\ E_{21} & \cdots & E_{d_f 2} \\ \vdots & \vdots & \vdots \\ E_{d_f 1} & \cdots & E_{d_f d_f} \end{bmatrix}, \quad E_{ij} = \lambda \int_a^b \phi_j(t) \phi_i(t) dt, \\
K &= \begin{bmatrix} K_{11} & \cdots & K_{d_f 1} \\ K_{21} & \cdots & K_{d_f 2} \\ \vdots & \vdots & \vdots \\ K_{d_f 1} & \cdots & K_{d_f d_f} \end{bmatrix}, \quad K_{ij} = \int_a^b \int_a^b K(t, s) \phi_j(s) \phi_i(t) ds dt, \\
F &= \left[\int_a^b f(t) \phi_1(t) dt \quad \int_a^b f(t) \phi_2(t) dt \quad \cdots \quad \int_a^b f(t) \phi_{d_f}(t) dt \right]^T, \\
C &= [c_1 \quad c_2 \quad \cdots \quad c_{d_f}]^T.
\end{aligned} \tag{15}$$

The system (14) can be expressed as

$$(E - K) \cdot C = F. \tag{16}$$

Hence the coefficient matrix C is found as follows:

$$C = (E - K)^{-1} \cdot F. \tag{17}$$

By substituting these coefficients to the left hand side of equality (11), the desired approximate solution of the Galerkin method is obtained.

3.2. Adaptive Refinement by Using Demkowicz's Optimization for Galerkin Method Solution on the Fine Mesh L^f . As explained in the abstract, for adaptive refinement we solve an optimization problem which was originally used by Demkowicz [19] with Sobolev spaces for solving one- and two-dimensional elliptic and Maxwell problems. In this study instead of Sobolev norm, L^2 -norm is used and naturally inner products are L^2 -inner product. Under this choice our optimization problem turns into

$$\frac{\|u - \Pi_{hp} u\|^2 - \|u - \Pi_{hp_{opt}} u\|^2}{N_{hp_{opt}} - N_{hp}} \rightarrow \min. \tag{18}$$

For determining optimal mesh (18) is solved on each element of the fine mesh. For simplicity we solve the problem on element of the form $I = [a, b]$ as a representative element for all elements of the coarse mesh which were in the form $I_i = [t_i, t_{i+1}]$ ($i = 1, 2, \dots, n-1$). In other words we will start with a mesh consisting of just one element which is the interval itself. Let $D^c = [p]$ be the list of element local polynomial order of approximation of the element I^c . Refining this element in both h and p we get a fine mesh with the following new list of node points:

$$L^f = \left[a \quad \frac{a+b}{2} \quad b \right], \quad D^f = [p+1 \quad p+1]. \tag{19}$$

There are also other possible refinements (just in h or p or in both but with different choice of refinements in p) which produce the following four possible optimal mesh choices:

$$L^{\text{opt}} = [a \quad b], \quad D^{\text{opt}} = [p+1], \tag{20}$$

$$\text{or } L^{\text{opt}} = \left[a \quad \frac{a+b}{2} \quad b \right], \quad D^{\text{opt}} = [p \quad p], \tag{21}$$

$$\text{or } L^{\text{opt}} = \left[a \quad \frac{a+b}{2} \quad b \right], \quad D^{\text{opt}} = [p \quad p+1], \tag{22}$$

$$\text{or } L^{\text{opt}} = \left[a \quad \frac{a+b}{2} \quad b \right], \quad D^{\text{opt}} = [p+1 \quad p]. \tag{23}$$

Consider the fine mesh solution u given with (11). Firstly L^2 -projections of u on the coarse mesh $\Pi_{hp} u$ and on the optimal meshes $\Pi_{hp_{opt}} u$ are calculated. During these calculations we introduced some matrices and notations that we need. Let d_c and d_{opt} be the total number of basis functions and let ξ_j^c , ($j = 1, 2, \dots, d_c$) and ξ_j^{opt} , ($j = 1, 2, \dots, d_{opt}$) be the basis functions of the coarse and optimal mesh cases, respectively:

$$\Pi_{hp} u = \sum_{j=1}^{d_c} \xi_j^c \phi_j^c, \tag{24}$$

$$\Pi_{hp_{opt}} u = \sum_{j=1}^{d_{opt}} \xi_j^{\text{opt}} \phi_j^{\text{opt}}. \tag{25}$$

Let $\xi^c = [\xi_1^c \quad \xi_2^c \quad \cdots \quad \xi_{d_c}^c]^T$, $\xi^{\text{opt}} = [\xi_1^{\text{opt}} \quad \xi_2^{\text{opt}} \quad \cdots \quad \xi_{d_{opt}}^{\text{opt}}]^T$ be the coefficient matrices corresponding to (24) and (25), respectively. We calculated L^2 -projections of the Galerkin method solution (11) as Larson and Bengzon explained

in [43]. For calculating $\Pi_{hp}u$ given with (24) above, we define two matrices Z^c and M^c that we need as follows:

$$Z^c = \begin{bmatrix} Z_{11}^c & \cdots & Z_{1d_f}^c \\ Z_{21}^c & \cdots & Z_{2d_f}^c \\ \vdots & \vdots & \vdots \\ Z_{d_c1}^c & \cdots & Z_{d_cd_f}^c \end{bmatrix}, \quad Z_{ij}^c = \int_a^b \phi_i^c(s) \phi_j^f(s) ds,$$

$$M^c = \begin{bmatrix} M_{11}^c & \cdots & M_{1d_c}^c \\ M_{21}^c & \cdots & M_{2d_c}^c \\ \vdots & \vdots & \vdots \\ M_{d_c1}^c & \cdots & M_{d_cd_c}^c \end{bmatrix}, \quad M_{ij}^c = \int_a^b \phi_i^c(s) \phi_j^c(s) ds.$$
(26)

We obtain the coefficient matrix ξ^c as follows:

$$\xi^c = (M^c)^{-1} \cdot Z^c \cdot C. \quad (27)$$

Substituting these coefficients in (24) we obtain the L^2 -projection function $\Pi_{hp}u$. For calculating $\Pi_{hp_{opt}}u$ given with (25), we need two new matrices Z^{opt} and M^{opt} defined as follows:

$$Z^{opt} = \begin{bmatrix} Z_{11}^{opt} & \cdots & Z_{1d_f}^{opt} \\ Z_{21}^{opt} & \cdots & Z_{2d_f}^{opt} \\ \vdots & \vdots & \vdots \\ Z_{d_{opt}1}^{opt} & \cdots & Z_{d_{opt}d_f}^{opt} \end{bmatrix}, \quad Z_{ij}^{opt} = \int_a^b \phi_i^{opt}(s) \phi_j^f(s) ds,$$

$$M^{opt} = \begin{bmatrix} M_{11}^{opt} & \cdots & M_{1d_{opt}}^{opt} \\ M_{21}^{opt} & \cdots & M_{2d_{opt}}^{opt} \\ \vdots & \vdots & \vdots \\ M_{d_{opt}1}^{opt} & \cdots & M_{d_{opt}d_{opt}}^{opt} \end{bmatrix}, \quad M_{ij}^{opt} = \int_a^b \phi_i^{opt}(s) \phi_j^c(s) ds.$$
(28)

We obtain the coefficient matrix ξ^{opt} as follows:

$$\xi^{opt} = (M^{opt})^{-1} \cdot Z^{opt} \cdot C. \quad (29)$$

Substituting these coefficients in (25) we obtain the L^2 -projection function $\Pi_{hp_{opt}}u$.

We reformulate the right hand side of the optimization problem (18) in terms of the matrices we introduced before by using (27) and (29) as follows:

$$\left(C^T \cdot \left[(Z^{opt})^T \cdot ((M^{opt})^{-1})^T \cdot Z^{opt} - (Z^c)^T \cdot ((M^c)^{-1})^T \cdot Z^c \right] \cdot C \right) \times (d^{opt} - d^c)^{-1}.$$
(30)

The value (30) is calculated for each of the four possible optimal mesh refinements of the coarse mesh element.

The case giving the minimum value is replaced with that element of the coarse mesh. Starting from the first element of the coarse mesh we repeat this procedure for each coarse mesh element, respectively. Joining all these replaced elements at node points of the coarse mesh, we obtain adaptively refined new mesh which is the optimal mesh we are trying to achieve. During this process to guarantee the continuity of the approximate solution at node points of the coarse mesh, the boundary conditions at node points of the coarse mesh have to be fixed. In other words, the values of the fine mesh solution u and its L_2 -projections is forced to take the same value at the node points of the coarse mesh.

We start with the L^2 -projection function $\Pi_{hp}u$. If we equalize the coefficients of the L^2 -projection function $\Pi_{hp}u$ with fine mesh solution u at node points a and b of the sample element $I^c = [a, b]$ we get the following results:

$$\begin{aligned} \Pi_{hp}u(a) &= u(a) \\ \implies \xi_1^c \cdot \phi_1^c(a) + 0 &= c_1 \cdot \phi_1^f(a) + 0 \\ \implies \xi_1^c &= c_1, \end{aligned}$$
(31)

$$\begin{aligned} \Pi_{hp}u(b) &= u(b) \\ \implies \xi_2^c \cdot \phi_2^c(b) + 0 &= c_3 \cdot \phi_3^f(b) + 0 \\ \implies \xi_2^c &= c_3. \end{aligned}$$

In this case calculating the coefficients in the coefficient matrix ξ^c except ξ_1^c and ξ_2^c will be sufficient. Removing the first two columns of the matrices given with (26) and the first two elements of the matrix ξ^c and solving the remaining system brings the desired coefficients. Since the structure of the optimal element given with (20) is similar to the coarse mesh, the calculation for this case is similar with the one done for obtaining the coefficient matrix ξ^c . For this optimal case, $\xi_1^{opt} = c_1$ and $\xi_2^{opt} = c_3$. Deleting the first two columns of the matrices given with (28) and the first two elements of the matrix ξ^{opt} and solving the remaining system brings the remaining coefficients of the projection function $\Pi_{hp_{opt}}u$.

For the remaining three optimal cases given with (20), (21), and (22) we follow the same way:

$$\begin{aligned} \Pi_{hp_{opt}}u(a) &= u(a) \\ \implies \xi_1^{opt} \cdot \phi_1^{opt}(a) + 0 &= c_1 \cdot \phi_1^f(a) + 0 \\ \implies \xi_1^{opt} &= c_1, \end{aligned}$$
(32)

$$\begin{aligned} \Pi_{hp_{opt}}u(b) &= u(b) \\ \implies \xi_3^{opt} \cdot \phi_3^{opt}(b) + 0 &= c_3 \cdot \phi_3^f(b) + 0 \\ \implies \xi_3^{opt} &= c_3. \end{aligned}$$

For these cases of optimal meshes, the first and the third columns of matrices (28) and the first and the third elements (ξ_1^{opt} and ξ_3^{opt}) of the matrix ξ^{opt} are deleted and the remaining system is solved in order to get the remaining coefficients of the projection $\Pi_{hp_{\text{opt}}}$.

4. Sloan Iteration Solution and Adaptive Refinement by Using Demkowicz's Optimization

The Fredholm integral equation of the second kind given with formula (10) can be reformulated as

$$x = \frac{1}{\lambda} (f + z), \quad (33)$$

where $z = \mathcal{K}x = \int_a^b K(t, s)x(s)ds$, $t \in [a, b]$. Atkinson [44] defined iterated projection solution $\tilde{x}_n$ for a given projected method solution x_n as

$$\tilde{x}_n = \frac{1}{\lambda} (f + \mathcal{K}x_n), \quad (34)$$

and beside this he mentioned that although such iterations are found in the literature in many places, Sloan [45] first recognized the importance of doing one such iteration and in his honor $\tilde{x}_n$ is often called the Sloan iterate. We express (34) in a more clear and general way as follows:

$$\tilde{x}(t) = \frac{1}{\lambda} \left[\int_a^b K(t, s)x(s)ds + f(t) \right], \quad t \in [a, b]. \quad (35)$$

Substituting the Galerkin method solution on the fine mesh given with formula (11) to the right hand side of formula (35), the iterated solution on the fine mesh is obtained as follows:

$$\tilde{x}(t) = \frac{1}{\lambda} \left[f(t) + \sum_{j=1}^d c_j \int_a^b K(t, s)\phi_j(s)ds \right], \quad t \in [a, b]. \quad (36)$$

We solve the optimization problem given with (18) for iterated solution as it was solved for Galerkin method solution. In this case u in (18) is taken as the iterated solution $\tilde{x}$ given with (36).

We define two matrices Z^c and M^c needed in the calculation of $\Pi_{hp}u$ given with (24) as follows:

$$Z^c = \begin{bmatrix} Z_{11}^c & \cdots & Z_{d_f 1}^c \\ Z_{12}^c & \cdots & Z_{d_f 2}^c \\ \vdots & \vdots & \vdots \\ Z_{1d_c}^c & \cdots & Z_{d_f d_c}^c \end{bmatrix}, \quad Z^c = \int_a^b \int_a^b K(t, s)\phi_j^f(s)\phi_i^c(t)dsdt, \quad (37)$$

$$F^c = \left[\int_a^b f(t)\phi_1^c(t)dt \quad \int_a^b f(t)\phi_2^c(t)dt \quad \cdots \quad \int_a^b f(t)\phi_{d_c}^c(t)dt \right]^T.$$

We obtain the coefficient matrix ξ^c as

$$\xi^c = \frac{1}{\lambda} (M^c)^{-1} \cdot (F^c + Z^c \cdot C). \quad (38)$$

Substituting these coefficients in (24) we obtain the L^2 -projection function $\Pi_{hp}u$ for the iterated solution. We need two new matrices Z^{opt} and M^{opt} in the calculation of $\Pi_{hp_{\text{opt}}}u$ which are given as follows:

$$Z^{\text{opt}} = \begin{bmatrix} Z_{11}^{\text{opt}} & \cdots & Z_{d_f 1}^{\text{opt}} \\ Z_{12}^{\text{opt}} & \cdots & Z_{d_f 2}^{\text{opt}} \\ \vdots & \vdots & \vdots \\ Z_{1d_{\text{opt}}}^{\text{opt}} & \cdots & Z_{d_f d_{\text{opt}}}^{\text{opt}} \end{bmatrix}, \quad Z_{ij}^{\text{opt}} = \int_a^b \int_a^b K(t, s)\phi_j^f(s)\phi_i^{\text{opt}}(t)dsdt, \quad (39)$$

$$F^{\text{opt}} = \left[\int_a^b f(t)\phi_1^{\text{opt}}(t)dt \quad \int_a^b f(t)\phi_2^{\text{opt}}(t)dt \quad \cdots \quad \int_a^b f(t)\phi_{d_{\text{opt}}}^{\text{opt}}(t)dt \right]^T.$$

We obtain the coefficient matrix ξ_{opt} as follows:

$$\xi^{\text{opt}} = \frac{1}{\lambda} (M^{\text{opt}})^{-1} \cdot (F^{\text{opt}} + Z^{\text{opt}} \cdot C). \quad (40)$$

Substituting these coefficients in (25) we obtain the L^2 -projection function $\Pi_{hp_{\text{opt}}}u$ for the iterated solution.

As the right hand side of the optimization problem (18) was reformulated in matrix form for Galerkin method

TABLE 1: Error values of (42) in Example 1.

n	G_{L^2}	$G_{\max}$	S_{L^2}	$S_{\max}$
I	$3.807778247544568e + 00$	$3.962758499440351e + 00$	$6.142245995845852e - 02$	$2.691129516577462e - 02$
II	$3.681501530476621e - 01$	$4.734800657784639e - 01$	$1.436655818926428e - 05$	$5.651797188477303e - 06$
III	$3.980571071622163e - 02$	$7.073630464941472e - 02$	$1.437092335784317e - 05$	$5.601928975806914e - 06$
IV	$1.231396356019893e - 03$	$2.530443837131857e - 03$	$5.107562613500647e - 06$	$2.459469289561866e - 06$
V	$1.705913430341418e - 03$	$3.875831522837103e - 03$	$2.009899309359689e - 07$	$6.851382039485543e - 08$

TABLE 2: Error values of (42) in Example 1.

N, p	G_{L^2}	$G_{\max}$	S_{L^2}	$S_{\max}$
20, 3	$1.919678771148673e - 05$	$3.143456151022406e - 05$	$2.861019689160329e - 11$	$1.624300693947589e - 11$
30, 3	$8.285910426569432e - 06$	$1.378103734439584e - 05$	$8.932393923247606e - 14$	$5.240252676230739e - 14$
40, 3	$4.568972671388127e - 06$	$7.635657150117936e - 06$	$9.976748626959734e - 14$	$6.927791673660977e - 14$
50, 3	$2.892986173063029e - 06$	$4.818969607356394e - 06$	$9.852985133624295e - 14$	$6.750155989720952e - 14$

solution, the same will be done for the iterated solution. We reformulate the right hand side of the optimization problem (18) in terms of the matrices we introduced before by using (38) and (40) as follows:

$$\begin{aligned} & \frac{1}{\lambda^2} \left(\left((F^{\text{opt}})^T + C^T \cdot (Z^{\text{opt}})^T \right) \cdot \left((M^{\text{opt}})^{-1} \right)^T \right. \\ & \quad \cdot \left(F^{\text{opt}} + Z^{\text{opt}} \cdot C \right) - \left((F^c)^T + C^T \cdot (Z^c)^T \right) \\ & \quad \left. \cdot \left((M^c)^{-1} \right)^T \cdot (F^c + Z^c \cdot C) \right). \end{aligned} \quad (41)$$

As explained for Galerkin method solution at the end of Section 3.2, on each element of the coarse mesh expression (41) should be calculated for each of four possible optimal mesh refinement cases and the case giving the minimum value is replaced with that element of the coarse mesh. For the continuity of the approximate solution at node points of the coarse mesh, the boundary conditions at node points of the coarse mesh are fixed during the calculations in the same way as it was done for the Galerkin method solution.

5. Some Applications

Both methods are applied to some problems in [46] on Fredholm integral equations of the second kind with smooth kernel and discontinuous kernel and results are stated. For each example we have presented error values in two different kinds of tables. In the first kind, we give the error values of the consecutive solutions where n denotes the repeating order. Here resulting refined mesh is used as the coarse mesh for the later turn. In the second kind, we give the error values when we use N number of equidistant node points on coarse mesh and p as element local polynomial order of approximation at each element of the coarse mesh. In both tables while G_{L^2} and S_{L^2} denote the L^2 -errors, $G_{\max}$ and $S_{\max}$ denote the maximum absolute error at node points of the fine mesh (obtained from coarse mesh) of the Galerkin method solution and the iterated solution, respectively.

In this study our main and final goal is to reach better approximations by applying adaptive refinement together with Sloan iteration to Galerkin method solutions and to examine them. For this reason in all examples presented, we give two kinds of graphs with relative errors on log-log scale: one with relative error in L^2 -norm and one with relative error in maximum norm on y -axis and both with number of degrees of freedoms on the x -axis via Sloan iteration results. Graphs clearly show the decrease in relative error while number of degrees of freedoms are increasing. For simplicity log of number of degrees of freedoms is represented by “log(#dofs)” on the x -axis.

In order to illustrate the refinement process better we provide more details for the first example than for the latter ones: besides the error plots for the Sloan iteration we also add the corresponding graphs for the Galerkin method and show the mesh refinement for the five consecutive runs.

Example 1. The exact solution of the problem,

$$\begin{aligned} x(s) + \frac{1}{\pi} \int_{-\pi}^{\pi} \frac{0.3}{1 - 0.64 \cos^2((s+t)/2)} x(t) dt \\ = 25 - 16 \sin(s^2), \quad -\pi \leq s \leq \pi, \end{aligned} \quad (42)$$

is given with $x(t) = 17/2 + (128/7) \cos(2t)$. Let the coarse mesh be given with the lists $L = [-\pi, \pi]$, $D = [2]$. The problem is solved five times consecutively.

As we see in each row of Table 1 Sloan iteration cause a decrease in both G_{L^2} and $G_{\max}$ error values for each run. Also we see a general decrease in all error types, especially this is much more clear for errors via Sloan iteration given in the last two columns of the table.

When we look to the relative error graphs given with Figures 1 and 2, we see that relative errors via Sloan iteration get smaller values rather than the ones via Galerkin method.

In Table 2 we give the error for the computations starting from four different initial coarse meshes with 20, 30, 40, and 50 equidistant node points on the interval $[-\pi, \pi]$ and with initial element local polynomial order of $p = 2$. From this

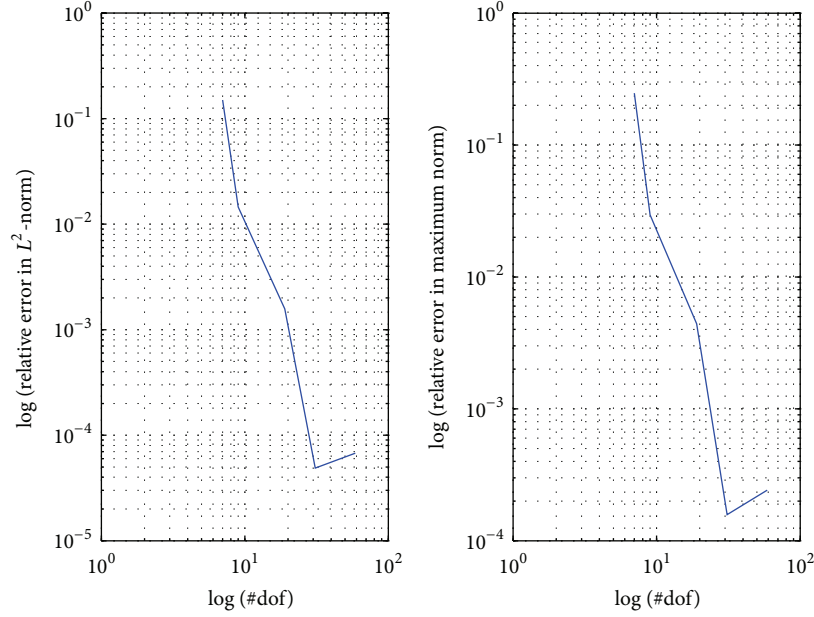


FIGURE 1: Convergence graphs of relative error in L^2 -norm and maximum norm via Galerkin method for (42) in Example 1.

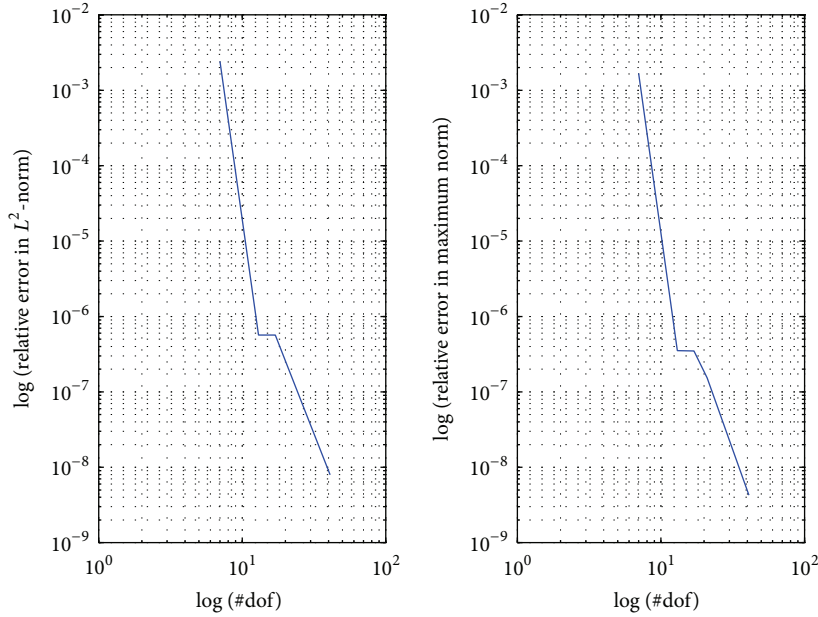


FIGURE 2: Convergence graphs of relative error in L^2 -norm and maximum norm via Sloan iteration for (42) in Example 1.

table we see that, at just one run of the procedure, we can reach much more smaller error values with lower element local polynomial order of approximation by just increasing the number of nodes. In [46] the error at nodes obtained by using Nyström method was given with the value $1.1e - 8$. Just for this example as a sample we give a graph including a diagram which shows us the mesh refinement step by step for the five consecutive runs via Galerkin method results and the lists of the obtained optimal mesh at the end of the five consecutive runs for both methods.

In Figure 3 we see the optimal mesh selections chosen by the optimization problem (18). In the first run it chose just to do p refinement and increase the element local polynomial order of approximation from 2 to 3 which corresponds to choice (20). In the second run it chose to make both h and p refinements together by choosing case (22). In the third run it chose to refine the first element as in (23) and the second one as in (20). On the fourth run while it chose to refine the first and third elements of level III by (22), it chose to refine the middle element by (20). Finally in the fifth run refining

TABLE 3: Error values of (45) in Example 2.

n	G_{L^2}	$G_{\max}$	S_{L^2}	$S_{\max}$
I	$1.154995991009992e - 05$	$4.800924769501891e - 05$	$9.079011385069973e - 06$	$2.077330804106659e - 05$
II	$1.144321554963263e - 05$	$7.856882758505712e - 05$	$4.481278123307043e - 06$	$1.430730253160206e - 06$
III	$2.260073176335741e - 05$	$7.192862655699961e - 05$	$3.073165554649129e - 07$	$3.282068198745547e - 07$
IV	$2.373416815667144e - 05$	$7.277671609520753e - 05$	$1.846336864089446e - 08$	$4.620430288371225e - 08$

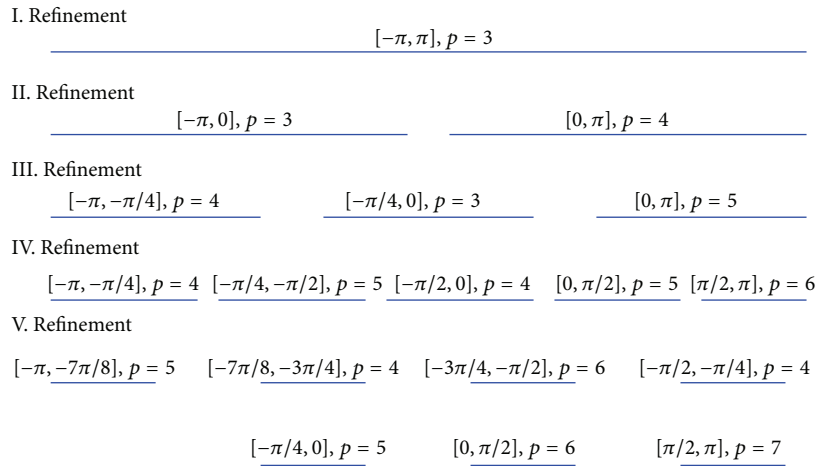


FIGURE 3: Representation of mesh refinement via Galerkin method for (42) in Example 1.

the first element of level IV by (23), the second, fourth, and fifth elements of level IV by (20), and the third element of level IV by (22), the optimal mesh at the end of five consecutive runs given below for the Galerkin method is obtained.

For Galerkin method solution,

$$\begin{aligned}
 L &= [-3.1416e + 00 \quad -2.7489e + 00 \quad -2.3562e + 00 \\
 &\quad -1.5708e + 00 \quad -7.8540e - 01 \quad 0 \\
 &\quad 1.5708e + 00 \quad 3.1416e + 00], \\
 D &= [5 \quad 4 \quad 6 \quad 4 \quad 5 \quad 6 \quad 7].
 \end{aligned} \quad (43)$$

For iterated solution we obtain the final optimal mesh as

$$\begin{aligned}
 L &= [-3.1416e + 00 \quad -1.5708e + 00 \quad -7.8540e - 01 \\
 &\quad 0 \quad 1.5708e + 00 \quad 2.3562e + 00 \quad 3.1416e + 00], \\
 D &= [5 \quad 5 \quad 4 \quad 5 \quad 5 \quad 4].
 \end{aligned} \quad (44)$$

Example 2. The exact solution of the problem

$$-\frac{x(s)}{2} - \int_0^1 \frac{0.1}{0.01 + (s-t)^2} x(t) dt = f(s), \quad 0 \leq s \leq 1, \quad (45)$$

is $x(t) = 0.06 - 0.8t + t^2$. Let the coarse mesh be given with lists $L = [0 \quad 1]$ and $D = [2]$. The problem is solved four times consecutively.

We observe from the rows of Table 3 that Sloan iteration decreases the G_{L^2} and $G_{\max}$ errors as we saw in Example 1.

Besides while the error values obtained by the consecutive runs via Galerkin method do not show much difference, the ones obtained by Sloan iteration are decreasing faster. The graphs given by Figure 4 for Example 2 clearly show the decrease in the relative errors in L^2 and maximum norms as expected.

Likewise we did in our first example, we used three different initial coarse meshes, having 30, 40, and 50 equidistant node points on the interval $[0, 1]$ and $p = 2$ as initial element local polynomial order of approximation at each element of these coarse mesh elements for all three situations. As in Example 1, we again see in Table 4 that we reach much more smaller error values with lower element local polynomial order of approximation by just increasing the number of nodes. In [46] the error at nodes obtained by using Nyström method was given with the value $6.6e - 6$.

Example 3. The solution of the equation

$$x(s) - \int_0^1 k(t, s) x(t) dt = \left(1 - \frac{1}{\pi^2}\right) \sin(\pi s), \quad 0 \leq s \leq 1, \quad (46)$$

with

$$k(t, s) = \begin{cases} s(1-t), & s \leq t \\ t(1-s), & t \leq s, \end{cases} \quad (47)$$

is $x(t) = \sin(\pi t)$. Let the coarse mesh be given with the lists $L = [0 \quad 1]$ and $D = [2]$. The problem is solved seven times consecutively.

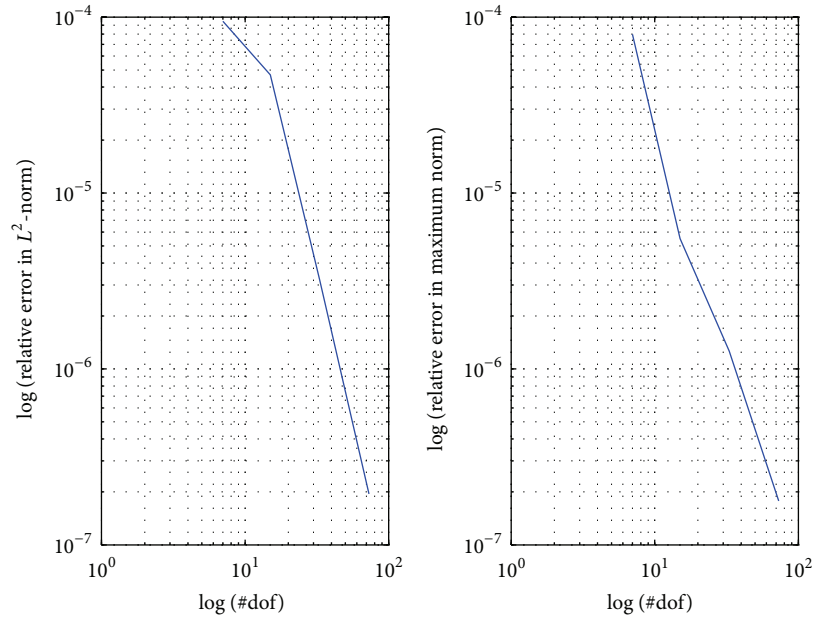
FIGURE 4: Convergence graphs of relative error in L^2 -norm and maximum norm via Sloan iteration for (45) in Example 2.

TABLE 4: Error values of (45) in Example 2.

N, p	G_{L^2}	$G_{\max}$	S_{L^2}	$S_{\max}$
30, 2	$3.827463454759924e-13$	$2.371158824843178e-12$	$3.827929692581457e-13$	$2.367273044256990e-12$
40, 2	$6.337936926296469e-14$	$3.875233467454109e-13$	$6.344645499378824e-14$	$3.914091273315989e-13$
50, 2	$1.595065414892088e-14$	$1.005862060310392e-13$	$1.598617548424624e-14$	$9.842127113302013e-14$

TABLE 5: Error values of (46) in Example 3.

n	G_{L^2}	$G_{\max}$	S_{L^2}	$S_{\max}$
I	$2.323175934794088e-03$	$4.178803217825712e-03$	$3.043325952187653e-03$	$2.499839591130204e-04$
II	$2.240147347334236e-03$	$1.853030198079919e-03$	$7.768764444034127e-04$	$6.367704404164343e-05$
III	$6.486006996900690e-04$	$1.540804529766398e-03$	$2.275583867630013e-04$	$2.260998286551796e-05$
IV	$2.737884048950595e-04$	$7.494545635317040e-04$	$1.076817056502688e-04$	$1.022158821584185e-05$
V	$4.309359697687976e-04$	$2.445614088301018e-03$	$8.482708369079025e-05$	$6.301343734582687e-06$
VI	$5.397336277740180e-04$	$3.256397951289958e-03$	$8.303708583183078e-05$	$5.990667999111743e-06$
VII	$6.252849507196142e-04$	$4.555052326156391e-03$	$4.721973545142185e-05$	$4.470476808404733e-06$

TABLE 6: Error values of (46) in Example 3.

N, p	G_{L^2}	$G_{\max}$	S_{L^2}	$S_{\max}$
30, 2	$1.248229331783191e-05$	$2.667447083548602e-06$	$1.110735027318169e-05$	$1.362431358398197e-06$
40, 2	$7.197918189051699e-06$	$6.660698562699352e-06$	$7.498309701859696e-06$	$8.492780161351021e-07$
50, 2	$4.754457518136059e-06$	$6.130650015201411e-06$	$5.508659819578954e-06$	$5.732876173780710e-07$

Table 5 shows that Sloan iteration causes a decrease in the error values obtained via Galerkin method and they are getting smaller in each seven consecutive runs. Graphs given by Figure 5 clearly show the decrease in both relative errors in L^2 and maximum norms as in the previous examples.

By using the same three different coarse meshes used in Example 2, finally we see from Table 6 that we can obtain smaller errors with lower element local polynomial order of

approximation by just increasing the number of nodes. In [46] the error at nodes obtained by using Nyström method was given with the value $1.7e-7$.

6. Conclusions

The two methods presented aimed to find and improve approximate solutions for Fredholm integral equations of

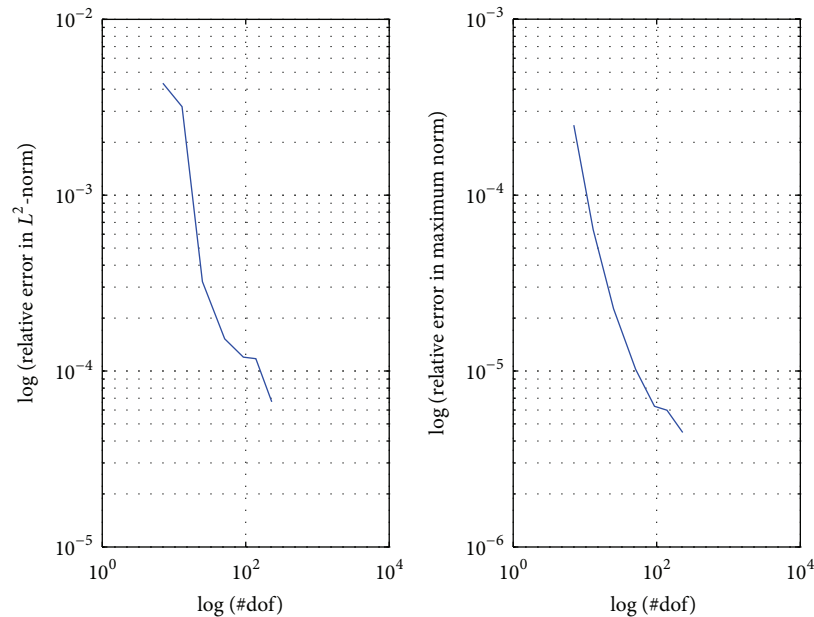


FIGURE 5: Convergence graphs of relative error in L^2 -norm and maximum norm via Sloan iteration for (46) in Example 3.

the second kind and to observe the effect of adaptive refinement on these solutions for both methods. Using polynomial type functions in the approximation of solutions for many kinds of problems in mathematics is one of the most usual methods. Generally in the whole solution interval, polynomials having the same degree are used. The main idea behind why we preferred to use adaptive refinement in our study is observing the changes in the results when we change this general approach. When we use adaptively refined meshes, this gives us a chance to use polynomials of different degree in different subintervals of the solution interval, which might cause obtaining better approximations. When comparing the maximum absolute error values at nodes of Examples 1 and 2 with the results in [46], we saw that our methods are able to reach much smaller error values by increasing the number of node points even by using polynomials having low degrees for these examples, which have smooth kernels. For Example 3 when the absolute error values at nodes are compared with the ones in [46] we see that they are getting closer to each other as we used more node points again by using polynomials of low degrees. In our study we also see that with Sloan iteration we obtain better approximations rather than Galerkin method as seen from the examples. The relative error graphs show us that Sloan iteration brings the expected decrease in relative error values of L^2 -error and maximum absolute error at node points of the fine mesh. The results showed that generally error values are better when we use polynomials of degree between 2 and 6, which is an advantage for decreasing the time we spend to solve the problems. Another advantage of our methods is that approximate solutions are found easily by using computer code written in Matlab. It is also observed that using polynomials with higher degrees can cause oscillations in the errors. The methods can

be improved not only to get better results, but also to solve other problem models by some modifications.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors appreciate helpful and valuable comments from all the referees to improve the paper.

References

- [1] H. Hochstadt, *Integral Equations*, Wiley-Interscience, New York, NY, USA, 1973.
- [2] S. Rahbar and E. Hashemizadeh, "A computational approach to the fredholm integral equation of the second kind," in *Proceedings of the World Congress on Engineering*, vol. 2, pp. 933–937, London, UK, July 2008.
- [3] B. G. Pachpatte, *Inequalities for Differential and Integral Equations*, Academic Press, London, UK, 1998.
- [4] C. T. H. Baker, *The Numerical Treatment of Integral Equations*, Clarendon Press, Oxford, UK, 1977.
- [5] W. Wang, "A new mechanical algorithm for solving the second kind of Fredholm integral equation," *Applied Mathematics and Computation*, vol. 172, no. 2, pp. 946–962, 2006.
- [6] A. T. Lonseth, "Approximate solutions of Fredholm-type integral equations," *Bulletin of the American Mathematical Society*, vol. 60, pp. 415–430, 1954.
- [7] I. Fredholm, "Sur une classe d'équations fonctionnelles," *Acta Mathematica*, vol. 27, no. 1, pp. 365–390, 1903.

- [8] T. A. Burton, *Volterra Integral and Differential Equations*, Academic Press, New York, NY, USA, 1983.
- [9] C. Corduneanu, *Integral Equations and Stability of Feedback Systems*, Academic Press, New York, NY, USA, 1973.
- [10] C. Corduneanu, *Principles of Differential and Integral Equations*, Chelsea, Bronx County, NY, USA, 2nd edition, 1977.
- [11] C. Corduneanu, *Integral Equations and Applications*, Cambridge University Press, Cambridge, UK, 1991.
- [12] G. Gripenberg, S. O. Londen, and O. Staffans, *Nonlinear Volterra and Integral Equations*, Cambridge University Press, Cambridge, UK, 1990.
- [13] A. M. Krasnoselskii, *Positive Solutions of Operator Equation*, Noordhoff Uitgevers, Groningen, The Netherlands, 1964.
- [14] R. K. Miller, *Nonlinear Volterra Integral Equations*, W. A. Benjamin, Menlo Park, Calif, USA, 1971.
- [15] F. G. Tricomi, *Integral Equations*, Interscience, New York, NY, USA, 1957.
- [16] I. Babuška, "Advances in the p and h - p versions of the finite element method. A survey," in *Numerical Mathematics Singapore 1988*, vol. 86 of *International Series of Numerical Mathematics*, pp. 31–46, Birkhäuser, Basel, Switzerland, 1988.
- [17] I. Babuska, B. A. Szabo, and I. N. Katz, "The p -version of the finite element method," *SIAM Journal on Numerical Analysis*, vol. 18, no. 3, pp. 515–545, 1981.
- [18] I. Babuška and M. R. Dorr, "Error estimates for the combined h and p -versions of the finite element method," *Numerische Mathematik*, vol. 37, no. 2, pp. 257–277, 1981.
- [19] L. Demkowicz, *Computing with hp -Adaptive Finite Elements, I. One- and Two-Dimensional Elliptic and Maxwell Problems*, Chapman & Hall/CRC, Boca Raton, Fla, USA, 2007.
- [20] M. Asadzadeh and K. Eriksson, "On adaptive finite element methods for Fredholm integral equations of the second kind," *SIAM Journal on Numerical Analysis*, vol. 31, no. 3, pp. 831–855, 1994.
- [21] K. Atkinson, *A Survey of Numerical Methods for the Solution of Fredholm Integral Equations of the Second Kind*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 1976.
- [22] Y. Ikebe, "The Galerkin method for the numerical solution of Fredholm integral equations of the second kind," *SIAM Review*, vol. 14, no. 3, pp. 465–491, 1972.
- [23] J. C. Nedelec, *Approximation des quations integrales en mecanique et physique*, Centre de Mathematiques, Ecole Polytechnique, Palaiseau, France, 1977.
- [24] I. H. Sloan, "A review of numerical methods for fredholm equations of the second kind," in *The Application and Numerical Solution of Integral Equations*, R. S. Anderssen, F. Hook, and M. de Lukas, Eds., Sijthoff and Noordhoff, Australian Nat. Univ., Canberra, Alphen aan den Rijn, The Netherlands, 1978.
- [25] W. L. Wendland, "On some mathematical aspects of boundary element methods for elliptic problems," in *The Mathematics of Finite Elements and Applications*, J. R. Whiteman, Ed., pp. 193–227, Academic Press, London, UK, 1985.
- [26] I. G. Graham, R. E. Shaw, and A. Spence, "Adaptive numerical solution of integral equations with application to a problem with a boundary layer," *Congressus Numerantium*, vol. 68, pp. 75–90, 1989.
- [27] E. Rank, I. Babuška, O. C. Zienkiewicz, J. Gago, and E. R. Oliveira, "Adaptivity and accuracy estimation for finite element and boundary integral element methods," in *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, pp. 79–94, John Wiley & Sons, New York, NY, USA, 1986.
- [28] D. H. Yu, "A posteriori error estimates and adaptive approaches for some boundary element methods," in *Boundary Element Methods*, C. A. Brebbia, W. L. Wendland, and G. Kuhn, Eds., pp. 241–256, Springer, Berlin, Germany, 1987.
- [29] M. J. Berger and P. Colella, "Local adaptive mesh refinement for shock hydrodynamics," *Journal of Computational Physics*, vol. 82, no. 1, pp. 64–84, 1989.
- [30] J. E. Schiermeier and B. A. Szabó, "Interactive design based on the p -version of the finite element method," *Finite Elements in Analysis and Design*, vol. 3, no. 2, pp. 93–107, 1987.
- [31] S. Larsen, *Numerical analysis of elliptic partial differential equations with stochastic input data [Ph.D. thesis]*, University of Maryland, College Park, Md, USA, 1986.
- [32] K. Eriksson and C. Johnson, "Adaptive finite element methods for parabolic problems I: a linear model problem," *SIAM Journal on Numerical Analysis*, vol. 28, no. 1, pp. 43–77, 1991.
- [33] P. K. Moore, "Comparison of adaptive methods for one-dimensional parabolic systems," *Applied Numerical Mathematics*, vol. 16, no. 4, pp. 471–488, 1995.
- [34] L. Demkowicz, W. Rachowicz, and P. Devloo, "A fully automatic hp -adaptivity," *Journal of Scientific Computing*, vol. 17, no. 1–4, pp. 117–142, 2002.
- [35] P. Houston and E. Süli, "A note on the design of hp -adaptive finite element methods for elliptic partial differential equations," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 2–5, pp. 229–243, 2005.
- [36] L. He and A. Zhou, "Convergence and complexity of adaptive finite element methods for elliptic partial differential equations," *International Journal of Numerical Analysis and Modeling*, vol. 8, no. 4, pp. 615–640, 2011.
- [37] R. H. W. Hoppe and M. Kieweg, "Adaptive finite element methods for mixed control-state constrained optimal control problems for elliptic boundary value problems," *Computational Optimization and Applications*, vol. 46, no. 3, pp. 511–533, 2010.
- [38] D. Pardo, L. Demkowicz, C. Torres-Verdín, and L. Tabarovskiy, "A goal-oriented hp -adaptive finite element method with electromagnetic applications. I. Electrostatics," *International Journal for Numerical Methods in Engineering*, vol. 65, no. 8, pp. 1269–1309, 2006.
- [39] E. P. Stephan, M. Maischak, and F. Leydecker, "An hp -adaptive finite element/boundary element coupling method for electromagnetic problems," *Computational Mechanics*, vol. 39, no. 5, pp. 673–680, 2007.
- [40] L. Botti, M. Piccinelli, B. Ene-Iordache, A. Remuzzi, and L. Antiga, "An adaptive mesh refinement solver for large-scale simulation of biological flows," *International Journal for Numerical Methods in Biomedical Engineering*, vol. 26, no. 1, pp. 86–100, 2010.
- [41] R. H. Hoppe, H. Wu, and Z. Zhang, "Adaptive finite element methods for the Laplace eigenvalue problem," *Journal of Numerical Mathematics*, vol. 18, no. 4, pp. 281–302, 2010.
- [42] K. E. Atkinson, *The Numerical Solution of Integral Equations of the Second Kind*, vol. 4 of *Cambridge Monographs on Applied and Computational Mathematics*, Cambridge University Press, Cambridge, UK, 1997.
- [43] M. B. Larson and F. Bengzon, *The Finite Element Method: Theory, Implementation, and Practice*, Springer, New York, NY, USA, 2010.

- [44] K. E. Atkinson, "A personal perspective on the history of the numerical analysis of Fredholm integral equations of the second kind," in *The Birth of Numerical Analysis*, pp. 53–72, World Scientific, Leuven, Belgium, 2008.
- [45] I. H. Sloan, "Improvement by iteration for compact operator equations," *Mathematics of Computation*, vol. 30, no. 136, pp. 758–764, 1976.
- [46] K. E. Atkinson and L. F. Shampine, "Algorithm 876: solving Fredholm integral equations of the second kind in Matlab," *ACM Transactions on Mathematical Software*, vol. 34, no. 4, pp. 1–20, 2008.