SOLVING VOLterra-FREDholM INtegral Equations BY Quadratic SPLine FUNCTION

Sarfraz Hassan Salim*, Rostam Karim Saeedb, Karwan Hama Faraj Jwamerc

*Department of Mathematics, College of Science, University of Sulaimani, Corresponding author: E-mail: sarfraz.salim@univsul.edu.iq

bDepartment of Mathematics, College of Science, Salahaddin University-Erbil, Iraq, E-mail: rostam. saeed@su.edu.krd

cDepartment of Mathematics, College of Science, University of Sulaimani, E-mail: karwan.jwamer@univsul.edu.iq

ABSTRACT

Using the quadratic spline function, this paper finds the numerical solution of mixed Volterra-Fredholm integral equations of the second kind. The proposed method is based on employing the quadratic spline function of the unknown function at an arbitrary point and using the integration method to turn the Volterra-Fredholm integral equation into a system of linear equations with respect to the unknown function. An approximate solution can be easily established by solving the given system. This is accomplished with the help of a computer program that runs on Python 3.9.
1. INTRODUCTION

Integral equations can be used to express a variety of mathematical physics topics. Some of these will be examined and treated explicitly as examples. It would be nearly impossible to compile a list of such applications. To say that integral equations play a role in practically every area of applied mathematics and mathematical physics is an understatement; thus, the literature on integral equations and their applications is extensive.

Many research have been conducted in recent years, with the results revealing the interaction of Fredholm integral equation, Volterra integral equation, mixed Volterra–Fredholm integral equation, and numerical part of these three types of integral equation.

In this work, we consider the linear mixed Volterra–Fredholm integral equations (MVFIEs) of the form:

\[ u(x) = f(x) + \lambda_1 \int_a^x K(x,t)u(t)dt + \lambda_2 \int_a^b L(x,t)u(t)dt, \quad (1) \]

where the functions \( f(x) \), and the kernels \( K(x,t) \) and \( L(x,t) \) are known \( L^2 \) analytic functions and \( \lambda_1, \lambda_2 \) are arbitrary constants, \( x \) is variable and \( u(x) \) is the unknown continuous function to be determined. Such equations arise in many applications in areas of physics, fluid dynamics, electrodynamics, and biology. Various formulations of boundary value problems, with Neumann, Dirichlet or mixed boundary conditions are reduced to such integral equations. They also provide mathematical models for the development of an epidemic and numerous other physical and biological problems.

It is well-known that the analytical solution of MVFIEs generally does not exist except for special cases, and thus, numerical method was the successful and effective method for solving these problems. Several numerical and approximate methods are used for solving MVFIEs such as Taylor polynomial by [16]; [15], least square method and Chebyshev polynomials by [5], Lagrange collocation method by [14], Series solution, successive approximation method and method of successive substitutions by [22], Trigonometric Functions and Laguerre Polynomials by [7], Touchard Polynomials (T-Ps) method by [1], Some iterative numerical methods by [12], Taylor polynomial by [6]. The reader can consult the following references for other information ([2], [3], [8], [9], [10], [11], [17], [18], [19], [20], [21], [24]) and the references therein.

We solved Equation (1) by linear spline function [23]. In this paper, Equation (1) studied by using quadratic spline function. The rest of this paper is organized as follows. In Section 2, we introduce our method for solving equation(1). In Section 3, we investigate several numerical examples, which demonstrate the effectiveness of our technique. In Section 4, some tentative conclusions will be given.
DESCRIPTION OF THE METHOD

In this section, we solve Equation (1) by using quadratic spline function [4], [25, P.151]. The unknown function \( u(x) \) in (1) approximated by the quadratic spline function \( Q(x) \). In the interval \([x_i, x_{i+1}]\) the quadratic spline function defined by the following formula:

\[
Q_i(x) = A_i(x)Q_i + B_i(x)Q_{i+1} + C_i(x)Q'_i, \tag{2}
\]

where \( A_i(x) = 1 - \frac{(x - t_i)^2}{h^2} \), \( B_i(x) = 1 - A_i(x) \), \( C_i(x) = \frac{(x - t_i)(t_{i+1} - x)}{h} \), and \( h = x_{i+1} - x_i \) for all \( i = 0, 1, \ldots, n - 1 \). Now substituting (2) in (1) and letting \( x = x_i \), we get

\[
Q_i = f_i + \lambda_1 \int_{x_i}^{x_i + h} K(x_i, t)Q(t)dt + \lambda_2 \int_{x_i}^{x_i + h} L(x_i, t)Q(t)dt
\]

\[
= f(x_i) + \lambda_1 \left[ \sum_{j=0}^{j=i-2} \int_{x_j}^{x_j + h} K(x_i, t)[A_j(t)Q_j + B_j(t)Q_{j+1} + C_j(t)Q'_j]dt \right] + \lambda_2 \left[ \int_{x_i}^{x_i + h} L(x_i, t)Q_0(t)dt + \int_{x_i}^{x_i + h} L(x_i, t)Q_1(t)dt + \cdots + \int_{x_{n-1}}^{x_{n-1} + h} L(x_i, t)Q_{n-1}(t)dt \right]
\]

\[
= f(x_i) + \lambda_1 \left[ \sum_{j=0}^{j=i-2} \int_{x_j}^{x_j + h} K(x_i, t)[A_j(t)Q_j + B_j(t)Q_{j+1} + C_j(t)Q'_j]dt \right] + \lambda_2 \left[ \int_{x_i}^{x_i + h} L(x_i, t)Q_0(t)dt + \int_{x_i}^{x_i + h} L(x_i, t)Q_1(t)dt + \cdots + \int_{x_{n-1}}^{x_{n-1} + h} L(x_i, t)Q_{n-1}(t)dt \right]
\]

By computing the integrals in the above equation using trapezoidal rule, we get
\[ Q_i = f_i + \frac{h}{2}(\lambda_1 K_{i0} + \lambda_2 L_{i0})Q_0 + h \sum_{j=1}^{i-1}(\lambda_1 K_{j1} + \lambda_2 L_{j1})Q_j + \frac{h}{2}(\lambda_1 (K_{ii} - 2K_{i,i-1}) + \lambda_2 L_{ii})Q_i \]  

(3)

for \( i = 0,1,\cdots,n \)

In this way, Equation (3) construct a system of linear equations with respect to the unknown function \( Q_j \). Briefly, this system can be rewritten as follows:

\[
CQ = F,
\]

(4)

where \( Q = \begin{bmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_n \end{bmatrix} \), \( F = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix} \), \( C = \begin{bmatrix} C_0 & C_1 & C_2 & \cdots & C_{n-1} & C_n \end{bmatrix} \),

\[
C_0 = \begin{bmatrix}
1 - \frac{\lambda_2 h}{2} L_{00} \\
-\frac{h}{2}(3\lambda_1 K_{10} + \lambda_2 L_{10}) \\
-\frac{h}{2}(\lambda_1 K_{20} + \lambda_2 L_{20}) \\
-\frac{h}{2}(\lambda_1 K_{30} + \lambda_2 L_{30}) \\
\vdots \\
-\frac{h}{2}(\lambda_1 K_{(n)0} + \lambda_2 L_{(n)0})
\end{bmatrix},
\]

\[
C_1 = \begin{bmatrix}
-\lambda_2 h L_{01} \\
1 - \frac{\lambda_2 h}{2}(K_{11} - 2K_{10} - \lambda_2 L_{11}) \\
-\lambda_2 h L_{11} \\
\ldots \\
-\lambda_2 h L_{n1}
\end{bmatrix},
\]

\[
C_2 = \begin{bmatrix}
-\lambda_2 h L_{02} \\
-\lambda_2 h L_{12} \\
1 - \frac{\lambda_2 h}{2}(K_{22} - 2K_{21} - \lambda_2 L_{22}) \\
-\lambda_2 h L_{22} \\
\ldots \\
-\lambda_2 h L_{n2}
\end{bmatrix},
\]

\vdots
In the sequel, making use of a standard rule to the resulting system yields an approximate solution of Equation (1) as \( Q_i(x) \) given by the Equation (2).

**NUMERICAL EXAMPLES**

In this section, we present three examples to illustrate the efficiency and the accuracy of the proposed method. The computed errors \( e_i \) are defined by \( e_i = |u_i - Q_i| \), where \( u_i \) is the exact solution of Equation (1) and \( Q_i \) is an approximate solution of the same equation. Also we compute Least square error (LSE) = \( \sum_{i=0}^{n} (u_i - Q_i)^2 \) and all computations are performed using Python program.

**Example 1** Consider Mixed Volterra-Fredholm integral equation

\[
 u(x) = -\frac{x^2}{2} - \frac{7x}{2} + 2 + \int_0^x u(t)dt + \int_0^x xu(t)dt. 
\]

The exact solution of this equation is given by \( u(x) = x + 2 \).

Table (1) demonstrates LSE obtained from applying our method on Example (1) for \( n = 5 \).
Table 1. The Numerical Results for Example (1) with $n = 5$.

| $x_i$ | $u_i$ | $Q_i$ | $|u_i - Q_i|$ | $|u_i - Q_i|^2$ |
|-------|-------|-------|---------------|---------------|
| 0     | 2     | 2     | 0             | 0             |
| 0.2   | 2.2   | 2.13629596 | 0.06370404    | 0.0040582     |
| 0.4   | 2.4   | 2.28579046 | 0.11420954    | 0.01304382    |
| 0.6   | 2.6   | 2.42850187 | 0.17149813    | 0.02941161    |
| 0.8   | 2.8   | 2.55956388 | 0.24043612    | 0.05780953    |
| 1.0   | 3     | 2.67597365 | 0.32402635    | 0.10499308    |
| LSE   |       |       | 2.09316234 $\times 10^{-1}$ |}

**Example 2** Consider Mixed Volterra-Fredholm integral equation

$$u(x) = 2\cos(x) - 1 + \int_0^x (x-t)u(t)dt + \int_0^\pi u(t)dt.$$  

The exact solution of this equation is given by $u(x) = \cos(x)$.

Table (2) demonstrates LSE obtained from applying our method on Example (2) for $n = 5$.

Table 2. The Numerical Results for Example (2) with $n = 5$.

| $x_i$ | $u_i$ | $Q_i$ | $|u_i - Q_i|$ | $|u_i - Q_i|^2$ |
|-------|-------|-------|---------------|---------------|
| 0     | 1     | 0.76338595 | 0.23661405    | 0.05598621    |
| $\pi$ | 0.59756842 | 0.21144858 | 0.011772      | 0.0447105     |
| $\frac{2\pi}{5}$ | 0.44912673 | 0.14010973 | 0.002637      | 0.01963074    |
Example 3 Consider Mixed Volterra-Fredholm integral equation

\[ u(x) = -\frac{x^5}{10} + 2x^3 - \frac{x^2}{2} + \frac{3x}{10} + \int_0^x (x+t)u(t)dt + \int_0^x (x-t)u(t)dt. \]

The exact solution of this equation is given by \( u(x) = 2x^3 + 1 \).

Table (3) demonstrates LSE obtained from applying our method on Example (3) for \( n = 5 \).

| \( \frac{3\pi}{5} \) | 0.01030637 | 0.31932337 | 0.009416 | 0.10196741 |
| \( \frac{4\pi}{5} \) | -0.54493082 | 0.26408617 | 0.021227 | 0.06974151 |
| \( \pi \) | -1.03436141 | -1.03436141 | 0.00118071 |
| LSE | | | | 2.93217071 \times 10^{-1} |

Table 3. The Numerical Results for Example (3) with \( n = 5 \).
Table 4. LSE for different values of \( n \) for Examples (1)-(3).

<table>
<thead>
<tr>
<th>( n )</th>
<th>Example 1</th>
<th>Example 2</th>
<th>Example 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>LES</td>
<td>( 4.1137035 \times 10^{-2} )</td>
<td>( 5.9176371 \times 10^{-3} )</td>
<td>( 1.8005313 \times 10^{-3} )</td>
</tr>
<tr>
<td>10</td>
<td>( 1.2014112 \times 10^{-1} )</td>
<td>( 4.5249249 \times 10^{-2} )</td>
<td>( 2.7925237 \times 10^{-2} )</td>
</tr>
<tr>
<td>20</td>
<td>( 8.8092531 \times 10^{-2} )</td>
<td>( 1.3833397 \times 10^{-2} )</td>
<td>( 4.3923665 \times 10^{-3} )</td>
</tr>
<tr>
<td>30</td>
<td>( 7.6633046 \times 10^{-4} )</td>
<td>( 2.0334663 \times 10^{-2} )</td>
<td>( 2.0334663 \times 10^{-2} )</td>
</tr>
<tr>
<td>40</td>
<td>( 3.9385354 \times 10^{-4} )</td>
<td>( 1.6040094 \times 10^{-1} )</td>
<td>( 1.9149630 \times 10^{-3} )</td>
</tr>
<tr>
<td>50</td>
<td>( 9.9939046 \times 10^{-4} )</td>
<td>( 9.9939046 \times 10^{-4} )</td>
<td>( 9.9939046 \times 10^{-4} )</td>
</tr>
</tbody>
</table>

CONCLUSION

The quadratic spline function is used in this paper to solve linear mixed Volterra-Fredholm integral equations, and it is a powerful numerical approach. The numerical results in the preceding section demonstrate that the proposed method can successfully tackle the Volterra-Fredholm type problem. Table (4) shows that the proposed method has extremely good stability; as \( n \) increases, the error decreases at first and then stabilizes. We also conclude that when the exact solution is a linear function, we have high accuracy. The present method can be easily extended to systems of mixed Volterra-Fredholm integral equations and systems of Volterra-Fredholm integro-differential equations. The current method may be simply extended to mixed Volterra-Fredholm integral equations and Volterra-Fredholm integro-differential equations.

REFERENCES


