

Numerical Solution of Mixed Volterra-Fredholm Integral Equations Using Linear Programming Problem

Pakhshan Mohammed Ameen Hasan*, Nejmaddin Abdulla Suleiman

Department of Mathematics, College of Education, Salahaddin University- Hawler, Kurdistan Region, Iraq

Abstract In this paper, a new technique is presented for solving linear mixed Volterra-Fredholm integral equations of the second kind. This technique is based on a polynomial of degree n and on the conversion of the integral equation to a linear programming problem, which will be solved by the Simplex method. For more illustration, an algorithm is suggested, and applied on several examples. The program is written in MATLAB (R2015a) to compute the results. To show the competency of the method and the accuracy of the results, comparison between the exact and the approximate solution are given by computing the absolute error and the least square error (L.S.E.).

Keywords Linear programming problem (LPP), Mixed linear Volterra-Fredholm integral equation of the second kind (MLV-FIE2nd), Simplex method

1. Introduction

Integral equations are found in different fields of science and several applications in approximation theory, fluid dynamics, electrodynamics, medicine, etc.

Recently, several works have been devoted to the existence of the solution of mixed type of Volterra-Fredholm integral equations [1, 3, 8]. The analytical solution of this type of integral equation is obtained in [1, 9, 11], while the numerical methods takes an important place in solving them [5, 7, 10, 14, 16, 17].

The linear mixed Volterra-Fredholm integral equation of the second kind (LMV-FIESK), which has the form

$$u(x) = f(x) + \int_0^x \int_a^b k(r, t)u(t)dt dr; a \leq x \leq b \quad (1)$$

is considered in this paper, where the free term f and the kernel k are known, while $u(x)$ is the unknown function which will be found.

In [3], Ibrahim, et al. used new iterative method for solving the mixed Volterra-Fredholm integral equations. Wang treated this problem in [6] by using Taylor collocation method. In [12], Shahooth solved the Volterra-Fredholm integral equations of the second kind by using Bernstein polynomials method. In addition, Ezzati and Najafalizadeh, used Cas wavelets for solving Volterra-Fredholm integral equations in [14].

Throughout this work, the central problem is the approximation of $u(x)$ by a function whose general form $u_n(x)$ depends on n parameters $\beta_1, \beta_2, \dots, \beta_n$. By choosing approximate values $\beta_j^*, j = 1, 2, \dots, n$ of these parameters according to some appropriate approximation criterion, we obtain a particular approximating function $u_n^*(x)$ of the equation (1).

2. Fundamental Concepts [2, 11]

2.1. Linear Programming Problem (LPP)

It is one of the most important optimization (maximization\ minimization) techniques developed in the field of operations research. More formally, linear programming is a technique for the optimization of a linear objective function (obj.fn.) subject to linear equality and/or inequality constraints, where the variables are nonnegative. In general, a linear programming model is defined as follows:

$$\text{max. or min. } Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

subject to

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n (\leq, =, \text{ or } \geq) b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n (\leq, =, \text{ or } \geq) b_2$$

⋮

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n (\leq, =, \text{ or } \geq) b_m$$

$$x_1, x_2, \dots, x_n \geq 0$$

The required smallest (or largest) value of the objective function is called the optimal value and the variables x_1, x_2, \dots, x_n that give the optimal solution are called the decision variables.

* Corresponding author:

pakhshansoran@gmail.com (Pakhshan Mohammed Ameen Hasan)

Published online at <http://journal.sapub.org/am>

Copyright © 2018 The Author(s). Published by Scientific & Academic Publishing

This work is licensed under the Creative Commons Attribution International

License (CC BY). <http://creativecommons.org/licenses/by/4.0/>

3. The Method

An important class of methods for solving integral equations involve substitution of a suitable approximation function $u_n(x)$ in place of the unknown solution $u(x)$. Here the key to effective approximating lies in the selection of the appropriate general form of $u_n(x)$ in the form

$$u_n(x) = \sum_{i=1}^n \beta_i x^{i-1} \quad (2)$$

The values of β_i are then determined using the transformation u_n which arises from substituting this function in the integral equation (1), which yields

$$\sum_{i=1}^n \beta_i x^{i-1} = f(x) + \int_0^x \int_a^b k(r, t) \sum_{i=1}^n \beta_i t^{i-1} dt dr + \epsilon_n(x) \quad (3)$$

Thus

$$\epsilon_n(x) = f(x) - \sum_{i=1}^n \beta_i \left(x^{i-1} - \int_0^x \int_a^b k(r, t) t^{i-1} dt dr \right) \quad (4)$$

Let

$$\varphi_i(x) = \left(x^{i-1} - \int_0^x \int_a^b k(r, t) t^{i-1} dt dr \right) \quad (5)$$

Thus equation (4) becomes:

$$\epsilon_n(x) = f(x) - \sum_{i=1}^n \beta_i \varphi_i(x) \quad (6)$$

One way of improving this technique is to select ($m > n$) points $x_j \in [a, b]$ and solve the over determined system of linear equations

$$\sum_{i=1}^n \beta_i x_j^{i-1} = u(x_j); j = 1, 2, \dots, m \quad (7)$$

This allows us to represent the function $u(x)$ on more than (n) points of $[a, b]$, when determining u_n^* . Further, for problem (7) a solution always exists in the following sense.

Let the residuals of equation (7) be defined as

$$\epsilon_n(x_j) = f(x_j) - \sum_{i=1}^n \beta_i \varphi_i(x_j); j = 1, 2, \dots, m \quad (8)$$

and

$$\rho = \sum_{j=1}^m |\epsilon_n(x_j)| \quad (9)$$

Consider the problem of determining

$$\|\epsilon_n^*\|_1 = \text{Minimum}_{\beta_1, \beta_2, \dots, \beta_n} \|\epsilon_n\|_1 \quad (10)$$

where

$$\|\epsilon_n^*\|_1 = \text{Minimum}_{\beta_1, \beta_2, \dots, \beta_n} \sum_{j=1}^m |\epsilon_n(x_j)| \quad (12)$$

By substituting equation (8) in equation (12), we get

$$\|\epsilon_n^*\|_1 = \text{Minimum}_{\beta_1, \beta_2, \dots, \beta_n} \sum_{j=1}^m |f(x_j) - \sum_{i=1}^n \beta_i \varphi_i(x_j)| \quad (13)$$

The quantity $\|\epsilon_n^*\|_1$ always exists and the optimal values of β_i yield a best l_1 (least-first-power) solution to equation (1).

Thus a general technique for determining l_1 solution to the LMV-FIESK is available which is based on solving the equivalent problem

$$\text{Minimize } \rho$$

subject to

$$\begin{cases} \rho + \sum_{i=1}^n \beta_i \varphi_i(x_j) \geq f(x_j) \\ \rho - \sum_{i=1}^n \beta_i \varphi_i(x_j) \geq -f(x_j) \end{cases}; j = 1, \dots, m. \quad (14)$$

β_i 's are unrestricted in sign.

The approximate values β_i^* of β_i which are calculated in (14) give an approximate solution of equation (1) as

$$u_n^*(x) = \sum_{i=1}^n \beta_i^* x^{i-1}$$

In practice we do not solve (14) exactly, but alternatively a corresponding discrete problem is solved using the Simplex method of Linear Programming.

4. The Algorithm

To find an approximate solution of (LMV-FIESK) perform the following steps:

Step 1: select two positive integers n and m .

Step 2: compute $f(x_j)$; $\forall j = 1, 2, \dots, m$.

Step 3: calculate $\varphi_i(x_j)$ in equation (5) for all $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, m$.

Step 4: use equation (9) to compute ρ , and then construct the objective function ($\text{Min. } \rho$) of the LPP.

Step 5: construct the constraints of the LPP from equation (14).

Step 6: use the Simplex method to find optimal approximate values β_i^* of β_i ; $i = 1, 2, \dots, n$.

Step 7: substitute these values of β_i^* 's; $i = 1, 2, \dots, n$ in the equation (2) to determine the approximate solution of equation (1).

5. Numerical Examples

In this section, several examples will be solved to show the accuracy of our approach.

Example 1. Consider the (MLV-FIE2nd)

$$u(x) = 2 + 4x - \frac{9}{8}x^2 - 5x^3 + \int_0^x \int_0^1 (r-t)u(t)dt dr$$

whose exact solution is $u(x) = 2 + 3x - 5x^3$.

Applying the algorithm of the described method with $n=4$ and $2m=8$ constraints will convert the integral equation to the following LPP:

$$\begin{aligned} \text{Minimize } \rho = & -4.2222222\beta_1 - 2.2777778\beta_2 \\ & -1.7962963\beta_3 - 1.5388889\beta_4 \end{aligned}$$

subject to

$$\begin{bmatrix} 5.2222222 & 2.2777778 & 1.7962963 & 1.5388889 \\ 5.3333333 & 2.6944444 & 1.9722222 & 1.6287037 \\ 5.3333333 & 3.0555556 & 2.3333333 & 1.9129630 \\ 5.2222222 & 3.3611111 & 2.8796296 & 2.6138889 \\ 3.2222222 & 2.2777778 & 1.7962963 & 1.5388889 \\ 3.1111111 & 1.8711111 & 1.6203704 & 1.4490741 \\ 3.1111111 & 1.5000000 & 1.2592593 & 1.1648148 \\ 3.2222222 & 1.1944444 & 0.7129630 & 0.4638889 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} \leq \begin{bmatrix} 9.5833333 \\ 10.6065481 \\ 10.2685000 \\ 7.4581852 \\ 5.5833333 \\ 4.5602185 \\ 4.8981481 \\ 7.7083333 \end{bmatrix}$$

where $\beta_1, \beta_2, \beta_3$, and β_4 are unrestricted in sign, which will be solved by the Simplex method to get the values of β_i 's for $i = 1, 2, 3, 4$.

Thus

$$\beta_1 = 2.0000741, \beta_2 = 2.9997596, \beta_3 = 0, \text{ and } \beta_4 = -4.9998956$$

Putting these values in equation (2) produces the approximate solution

$$u_n(x) = 2.0000741 + 2.9997596x - 4.9998956x^3$$

In this case the least square error (L.S.E.) is $(2.815477e - 08)$.

Increasing the value of m and taking $m=6>n$, will give a LPP with 4 unknowns and 12 constraints. Use the Simplex method to get the values of β_i 's, then put them in equation (2) to get the approximate solution

$$u_n(x) = 1.99999991 + 3.00000060x - 0.00000113x^2 - 4.99999938x^3$$

In Table (1), the real solution and the corresponding approximate solution using (LPP) are obtained; also the values of the absolute error and (L.S.E) are presented.

Table 1. The LPP results compared with exact solutions for $n=4$ and $m=6$

x_j	Exact solution $u(x_j)$	$u_n(x_j)$ (LPP)	Absolute Error $ u(x_j) - u_n(x_j) $
0	2	1.9999999100	9.00e-08
0.1	2.295	2.2949999595	4.05e-08
0.2	2.560	2.5599999902	9.80e-09
0.3	2.765	2.7650000058	5.78e-09
0.4	2.880	2.8800000100	1.00e-08
0.5	2.875	2.8750000066	6.63e-09
0.6	2.720	2.7199999994	9.00e-08
0.7	2.385	2.3849999921	7.92e-09
0.8	1.840	1.8399999884	1.16 e-08
0.9	1.055	1.0549999921	7.78 e-09
1.0	0	0.0000000070	7.00 e-09
L.S.E	1.032024809e - 14		

Example 2. The integral equation

$$u(x) = \sin(x) - \frac{1}{2}(x^2 + 2x) + \int_0^x \int_0^{\frac{\pi}{2}} (1 + rt)u(t)dt dr,$$

has the exact solution $u(x) = \sin(x)$.

Assume that the approximate solution is of the form

$$u_n(x) = \sum_{i=1}^n \beta_i x^{(i-1)}$$

By following the steps of the algorithm and choosing different values of n and m , we will get the results of L.S.E. that are listed in Table (2).

Table 2. The L.S.E. of Example 2 for different values of n and m

(n, m)	(4, 6)	(5, 8)	(6, 10)
L.S.E	2.7827618e-04	6.2680573e-05	8.5878406e-10

while, taking $n=8$ and $m=10$, gives the approximate solution

$$u_n(x) = -0.000000175 + 1.000002546x + 0.000000979x^2$$

$$-0.166705619x^3 + 0.000075868x^4 + 0.008302601x^5 - 0.000034394x^6 - 0.000170867x^7$$

Table (3) presents the comparison between the exact and the numerical solutions with $n=8$, and $m=10$.

Table 3. The comparison between the exact and the numerical solutions

x_j	Exact solution $u(x_j)$	Approximate solution $u_n(x_j)$	Absolute Error $ u(x_j) - u_n(x_j) $
0	0	1.1100000e-07	1.11e-07
0.1	0.1105170918	0.1105173029	2.11e-07
0.2	0.2442805516	0.2442804182	1.33 e-07
0.3	0.4049576422	0.4049576839	4.16e-08
0.4	0.5967298790	0.5967299999	1.21e-07
0.5	0.8243606353	0.8243605244	1.11e-07
0.6	1.0932712802	1.0932710072	2.73e-07
0.7	1.4096268952	1.4096268228	7.25e-08
0.8	1.7804327427	1.7804328799	1.37e-07
0.9	2.2136428000	2.2136425812	2.19e-07
1.0	2.7182818284	2.7182820090	1.81e-07
L.S.E	2.822575483e - 13		

Example 3. Consider the MV-FIE

$$u(x) = xe^x - \frac{x^2}{2} + \int_0^x \int_0^1 ru(t)dt; 0 \leq x \leq 1$$

the exact solution is $u(x) = xe^x$.

Performing the prescribed steps in the algorithm with different values of (n, m) , we get the results that are listed in Table (4).

Table 4. L.S.E. of Example (3) with different values of n and m

(n, m)	(4, 6)	(5, 10)	(7, 10)
L. S. E	7.33095339e-05	1.37998995e-05	3.21957269e-10

while, taking $n=8$ and $m=10$, gives the numerical solution

$$u_n(x) = -0.000000111 + 1.000024516x + 0.999613485x^2 + 0.502317746x^3 + 0.159879654x^4 + 0.052130940x^5 + 0.000000087x^6 + 0.004315692x^7$$

Table (5) presents the comparison between the exact and the numerical solutions for $n=8$, and $m=10$.

Table 5. The comparison between the exact and the numerical solutions

x_j	Exact solution $u(x_j)$	Approximate solution $u_n(x_j)$	Absolute Error $ u(x_j) - u_n(x_j) $
0	0	-0.000000175	1.750e-07
0.1(π)	0.309016994	0.309017127	1.326e-07
0.2(π)	0.587785252	0.587785120	1.318e-07
0.3(π)	0.809016994	0.809016968	2.589e-08
0.4(π)	0.951056516	0.951056312	2.039e-07
0.5(π)	1.000000000	0.999999670	3.303e-07
L.S.E	2.169220666e-13		

Example 4. Consider the MV-FIE

$$u(x) = f(x) + \int_0^x \int_0^{\frac{\pi}{2}} (r-t)u(t)dt dr, 0 \leq x \leq \frac{\pi}{2}$$

Where $f(x) = \cos(x) + \sin(x) - x^2 + \frac{\pi}{2}x$ and the exact solution $u(x) = \cos(x) + \sin(x)$.

Applying the algorithm of the LPP with $n=6$ and $m=9$ we get a LPP with 6 unknowns and 18 constraints which can be solved by the Simplex method, and then we get the results that shown in Table (6).

Table 6. The comparison between the exact and the numerical solutions

x_j	Exact soluton $u(x_j)$	Approx. solution $u_n(x_j)$	Absolute Error $ u(x_j) - u_n(x_j) $
0.0	1	1.000009658	9.658000e-06
0.1(π)	1.260073511	1.260080774	7.263590e-06
0.2(π)	1.396802247	1.396794509	7.737585e-06
0.3(π)	1.396802247	1.396794513	7.733623e-06
0.4(π)	1.260073511	1.260080794	7.283443e-06
0.5(π)	1	1.000009718	9.717654e-06
L.S.E		4.1319717736e-10	

6. Conclusions

In this paper, the linear programming method is introduced to solve the second kind mixed Volterra–Fredholm integral equations. Several examples are applied for illustration and good approximate results are found. Moreover, the results of (LPP) are compared with the exact solutions to demonstrate the implementation of the method. Also, it is claimed that better results can be obtained by increasing both the number of basis functions (n) and the number of constraints ($m > n$). The given numerical examples and the outcomes in Tables (1-6) support these claims.

REFERENCES

- [1] A. Aghajani, Y. Jalilian, *Existence and global attractivity of Solution of a nonlinear functional Integral Equations*, Communications in Nonlinear science and Numerical Simulation 15 (2010), 3306-3312.
- [2] H. Taha, *Operations Research*, United States of America. (1976), pp. 28-32.
- [3] H. Tidke, *Iexistence and Uniqueness of Continuous Solution of Mixed Type Integral Equations in Cone Metric Space*, Kathmandu University Journal of Science, Engineering and Technology, 7(1), 48-55, (2011).
- [4] H. Ibrahim, F. Attah, and G.T. Gyegwe, *On the solution of Volterra-Fredholm and mixed Volterra-Fredholm integral equations using the new iterative method*. Applied Mathematics, 6 (1): 1-5 (2016).
- [5] K. Maleknejad, M. Hadizadeh, *A New Computational Method for Volterra-Fredholm Integral Equation*, J. Comput. Math. Appl. 37 (1999), 37-48.
- [6] K. Wang, Q. Wang, *Taylor Ollocation Method and Convergnce Analysis for the Volterra-Fredholm Integral Equations*, Journal of Computational and Applied Mathematics, 260 (2014) 294-300.
- [7] L.M. Delves, J. Walsh, *Numerical Solution of Integral Equations*. Oxford. (1974), pp. 97-106.
- [8] M.A. Abdou, G.M. Abd Al-Kader, *Mixed Type of Fredholm-Volterra Integral Equation*, Le Matematiche, LX (2005), pp. 41-58.
- [9] M.A. Abdou, *Integral Equation of Mixed Type and Integrals of Orthogonal Polynomials*, J.Comp. Appl. Math. 138 (2002), 273-285.
- [10] M.A. Abdou, K.I. Mohamed, A.S. Ismail, *On the Numerical Solution of Fredholm-VolterraIntegral Equation*, Appl. Math. Comput. 146 (2003) 713-728.
- [11] M.A. Abdou, *On Asymptotic Method for Fredholm-Volterra Integral Equation of the Second Kind in Contact problem*, J. Com. Appl. Math. 154 (2003), 431-446.
- [12] M. Shahooth, *Numerical Solution for Mixed Volterra-Fredhlm Integral Equations of the Second Kind By Bernstein Polyno- mials Method*, Mathematical Theory and Modeling, 5 (10) 154-162 (2015).
- [13] Q. Wang, K. Wang, and S. Chen, *Least Square Approximation Method for the Solution of Volterra-Fredholm Integral Equations*, Journal of Computational and Applied Mathematics, 272 (2014), 141-147.
- [14] R. Ezzati, S. Najafalizadeh, *Numerical Methods for Solving Linear and Nonlinear Volterra-Fredholm Integral Equations by Using Cas Wavelets*, World Applied Sciences Journal 18 (12): 1847-1854, (2012).
- [15] S. D. Sharma, *Operations Research*, Meerut, U.P. India. (1989), pp. 3-7.
- [16] S.S. Ahmed, *Numerical Solution for Volterra-Fredholm Integral Equation of the Second Kind by Using Least squares technique*, Iraqi Journal of Science, 52 (2011), pp.504-512.
- [17] Z. Chen, W. Jiang, *An Approximate Solution for a Mixed Linear Volterra-Fredholm Integral Equations*, Applied Mathematics Letter, 25 (2012) 1131-1134.