

## Research Article

# Numerical Solution of Second-Order Fredholm Integro-differential Equations with Boundary Conditions by Quadrature-Difference Method

Chriscella Jalius and Zanariah Abdul Majid

*Institute for Mathematical Research, Universiti Putra Malaysia (UPM), 43400 Serdang, Selangor, Malaysia*

Correspondence should be addressed to Zanariah Abdul Majid; zana\_majid99@yahoo.com

Received 15 September 2016; Accepted 8 December 2016; Published 11 January 2017

Academic Editor: Mehmet Sezer

Copyright © 2017 C. Jalius and Z. Abdul Majid. 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.

In this research, the quadrature-difference method with Gauss Elimination (GE) method is applied for solving the second-order of linear Fredholm integrodifferential equations (LFIDEs). In order to derive an approximation equation, the combinations of Composite Simpson's 1/3 rule and second-order finite-difference method are used to discretize the second-order of LFIDEs. This approximation equation will be used to generate a system of linear algebraic equations and will be solved by using Gauss Elimination. In addition, the formulation and the implementation of the quadrature-difference method are explained in detail. Finally, some numerical experiments were carried out to examine the accuracy of the proposed method.

## 1. Introduction

Integrodifferential equation is an equation where the unknown function appears under the sign of integration and it contains the derivatives of the unknown function. In this study, we will focus on second-order linear Fredholm integrodifferential equations (LFIDEs) and they can be defined as follows:

$$y''(x) = q(x)y'(x) + p(x)y(x) + g(x) + \lambda \int_a^b k(x,t)y(t)dt \quad (1)$$

with Dirichlet boundary conditions

$$\begin{aligned} y(a) &= A_1, \\ y(b) &= B_1, \end{aligned} \quad (2)$$

where the functions  $q(x)$ ,  $p(x)$ , and  $g(x)$  and kernel  $k(x,t)$  are known,  $\lambda$  is a real parameter, and  $a, b$  are constant, while  $y(x)$  is an unknown function to be determined.

In this study, we have discovered four types of problems in second-order LFIDEs such as the following.

Type 1. When  $p(x) = 0$  and  $q(x) = 0$

$$y''(x) = g(x) + \lambda \int_a^b k(x,t)y(t)dt. \quad (3)$$

Type 2. When  $p(x) \neq 0$  and  $q(x) = 0$

$$y''(x) = p(x)y(x) + g(x) + \lambda \int_a^b k(x,t)y(t)dt. \quad (4)$$

Type 3. When  $p(x) = 0$  and  $q(x) \neq 0$

$$y''(x) = q(x)y'(x) + g(x) + \lambda \int_a^b k(x,t)y(t)dt. \quad (5)$$

Type 4. When  $p(x) \neq 0$  and  $q(x) \neq 0$

$$\begin{aligned} y''(x) &= q(x)y'(x) + p(x)y(x) + g(x) \\ &+ \lambda \int_a^b k(x,t)y(t)dt. \end{aligned} \quad (6)$$

Nowadays, there are many researchers who study in the field of integrodifferential equations since it has emerged in many scientific and physical engineering applications.

Numerical approaches were practiced in engineering field in order to obtain an approximation solution for problem (1). Therefore, many authors have studied the numerical solutions of Fredholm integrodifferential equations. Several numerical methods were used such as Adomian's Decomposition method [1], Generalised minimal residual method (GMRES) [2], CAS Wavelet method [3], compact finite difference [4], Lagrange interpolation method [5], Spline Collocation [6], differential transformation method [7], variational iteration method [8], Tau method [9], Legendre collocation matrix method [10], and quadrature-difference method [11, 12]. LFIDEs are usually difficult to solve analytically. Therefore, in this study numerical approaches are practiced in order to obtain an approximation solution for problem (1).

In this paper, we will focus on Types 1 and 2 only. Therefore, we will introduce a new combination of quadrature-difference method, namely, Composite Simpson's 1/3 rule with second-order finite difference (CSFD2), and it will be implemented to discretize Types 1 and 2 to generate a system of linear equations. In addition, Gauss Elimination (GE) will be used for solving the linear algebraic equations produced by the discretization of the second-order LFIDEs.

## 2. Formulation of Approximation Equations

In this section, quadrature and second-order finite-difference methods are applied to discretize the second-order of LFIDEs to construct approximation equations.

*2.1. Derivation of the Quadrature Method.* In order to form the system of linear equations, quadrature method is applied to approximate the integral parts in second-order of LFIDEs. Therefore, Composite Simpson's 1/3 rule will be used in this study and it can be expressed as follows:

$$\int_a^b y(x) dx = \frac{h}{3} \left[ y(a) + 4 \sum_{j=1}^{n/2} y(x_{2j-1}) + 2 \sum_{j=1}^{(n/2)-1} y(x_{2j}) + y(b) \right] - \frac{b-a}{180} h^4 y^{(4)}(\mu), \quad (7)$$

where  $x_j = a + jh$  ( $j = 0, 1, 2, \dots, n$ ) are the abscissas of the partition points of the integration in interval  $[a, b]$ .  $((b-a)/180)h^4 y^{(4)}(\mu)$  is the error term of (7) where  $a \leq \mu \leq b$ . In the next section, the formulation of the approximation equations for problem LFIDEs will be further discussed.

The constant step size,  $h$  in (7), can be defined as

$$h = \frac{b-a}{n}, \quad (8)$$

where  $n$  is the number of subintervals in the interval  $[a, b]$ .

Therefore, if Composite Simpson's 1/3 rule with  $n$  subintervals is applied in (3) and (4), then the integration parts in second-order of LFIDEs is approximated by

$$\int_a^b k(x, t) y(t) dt \approx \frac{h}{3} [k(x, t_0) y(t_0) + 4k(x, t_1) y(t_1) + 2k(x, t_2) y(t_2)$$

$$+ 4k(x, t_3) y(t_3) + 2k(x, t_4) y(t_4) + \dots + 4k(x, t_{n-3}) y(t_{n-3}) + 2k(x, t_{n-2}) y(t_{n-2}) + 4k(x, t_{n-1}) y(t_{n-1}) + k(x, t_n) y(t_n)]. \quad (9)$$

*2.2. Derivation of the Finite-Difference Method.* In solving second-order LFIDEs, the differential part will be approximated by using second-order derivative of finite-difference method which is fourth-order accuracy. Therefore, finite-difference method can be derived as follows.

Forward difference is

$$y''(x_i) = \frac{1}{12h^2} [35y(x_i) - 104y(x_{i+1}) + 114y(x_{i+2}) - 56y(x_{i+3}) + 11y(x_{i+4})] + O(h^4) \quad (10)$$

for  $i = 1$ .

Central difference is

$$y''(x_i) = \frac{1}{12h^2} [-y(x_{i-2}) + 16y(x_{i-1}) - 30y(x_i) + 16y(x_{i+1}) - y(x_{i+2})] + O(h^4) \quad (11)$$

for  $i = 2, \dots, n-2$ .

Backward difference is

$$y''(x_i) = \frac{1}{12h^2} [35y(x_i) - 104y(x_{i-1}) + 114y(x_{i-2}) - 56y(x_{i-3}) + 11y(x_{i-4})] + O(h^4) \quad (12)$$

for  $i = n-1$ , where  $h$  is the constants step size as defined in (8). Therefore, the differential term in second-order LFIDEs will be approximated by using equations (10), (11), and (12).

## 3. Formulation of Composite Simpson's 1/3 and Second-Order Finite-Difference Method

The formulation of Composite Simpson's 1/3 rule and second-order finite-difference method (CSFD2) for second-order LFIDEs is elaborated in this section. In order to generate the system of linear algebraic equations, equation in (9), (10), (11), and (12) will be substituted into (4).

*3.1. Formulation of CSFD2 for Types 1 and 2.* A system of linear algebraic equations will be generated in this section. The difference between Types 1 and 2 is only in the function of  $p(x)y(x)$  which is  $p(x) \neq 0$  in Type 2, while in Type 1  $p(x) = 0$ . Therefore, substitute (9) into (4) and we will get

$$y''(x_i) = p(x_i) y(x_i) + g(x_i) + \frac{\lambda h}{3} [k(x_i, t_0) y(t_0) + 4k(x_i, t_1) y(t_1) + 2k(x_i, t_2) y(t_2) + 4k(x_i, t_3) y(t_3) + 2k(x_i, t_4) y(t_4) + \dots + 4k(x_i, t_{n-3}) y(t_{n-3}) + 2k(x_i, t_{n-2}) y(t_{n-2}) + 4k(x_i, t_{n-1}) y(t_{n-1}) + k(x_i, t_n) y(t_n)]. \quad (13)$$

Accordingly, for  $y''(x_i) = y''(t_i) = y_i''$  and  $k(x_i, t_j) = k_{i,j}$ , the relation in (13) becomes

$$y_i'' = p_i y_i + g_i + \frac{\lambda h}{3} [k_{i,0} y_0 + 4k_{i,1} y_1 + 2k_{i,2} y_2 + 4k_{i,3} y_3 + \dots + 4k_{i,n-3} y_{n-3} + 2k_{i,n-2} y_{n-2} + 4k_{i,n-1} y_{n-1} + k_{i,n} y_n] \quad (14)$$

where  $i = 1, 2, \dots, n - 1$ .

Next, we substitute (10), (11), and (12) into (14); then we will have

$$\frac{35y_i - 104y_{i+1} + 114y_{i+2} - 56y_{i+3} + 11y_{i+4}}{12h^2} = p_i y_i + g_i + \frac{\lambda h}{3} [k_{i,0} y_0 + 4k_{i,1} y_1 + 2k_{i,2} y_2 + 4k_{i,3} y_3 + \dots + 4k_{i,n-3} y_{n-3} + 2k_{i,n-2} y_{n-2} + 4k_{i,n-1} y_{n-1} + k_{i,n} y_n] \quad (15)$$

for  $i = 1$ .

$$\frac{-y_{i-2} + 16y_{i-1} - 30y_i + 16y_{i+1} - y_{i+2}}{12h^2} = p_i y_i + g_i$$

$$+ \frac{\lambda h}{3} [k_{i,0} y_0 + 4k_{i,1} y_1 + 2k_{i,2} y_2 + 4k_{i,3} y_3 + \dots + 4k_{i,n-3} y_{n-3} + 2k_{i,n-2} y_{n-2} + 4k_{i,n-1} y_{n-1} + k_{i,n} y_n] \quad (16)$$

for  $i = 2, \dots, n - 2$ .

$$\frac{35y_i - 104y_{i-1} + 114y_{i-2} - 56y_{i-3} + 11y_{i-4}}{12h^2} = p_i y_i + g_i + \frac{\lambda h}{3} [k_{i,0} y_0 + 4k_{i,1} y_1 + 2k_{i,2} y_2 + 4k_{i,3} y_3 + \dots + 4k_{i,n-3} y_{n-3} + 2k_{i,n-2} y_{n-2} + 4k_{i,n-1} y_{n-1} + k_{i,n} y_n] \quad (17)$$

for  $i = n - 1$ .

Lastly, from (15), (16), and (17), the linear system equations for  $y_1, y_2, \dots, y_{n-2}, y_{n-1}$  are generated. Therefore, the generated system of linear algebraic equations can be written in matrix form of

$$\mathbf{A}\mathbf{Y} = \mathbf{F}, \quad (18)$$

where

$$\mathbf{A} = \begin{bmatrix} a_{1,1} + \alpha & b_{1,2} - 104 & a_{1,1} + 114 & b_{1,4} - 56 & \dots & b_{1,n-2} & a_{1,n-1} \\ a_{2,1} + 16 & b_{2,2} - \beta & a_{2,3} + 16 & b_{2,4} - 1 & \dots & b_{2,n-2} & a_{2,n-1} \\ a_{3,1} - 1 & b_{3,2} + 16 & a_{3,3} - \beta & b_{3,4} + 16 & \dots & b_{3,n-2} & a_{3,n-1} \\ a_{4,1} & b_{4,2} - 1 & a_{4,3} + 16 & b_{4,4} - \beta & \dots & b_{4,n-2} & a_{3,n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n-4,1} & b_{n-4,2} & \dots & b_{n-4,n-4} - \beta & a_{n-4,n-3} + 16 & b_{n-4,n-2} - 1 & a_{n-4,n-1} \\ a_{n-3,1} & b_{n-3,2} & \dots & b_{n-3,n-4} + 16 & a_{n-3,n-3} - \beta & b_{n-3,n-2} + 16 & a_{n-3,n-1} - 1 \\ a_{n-2,1} & b_{n-2,2} & \dots & b_{n-2,n-4} - 1 & a_{n-2,n-3} + 16 & b_{n-2,n-2} - \beta & a_{n-2,n-1} + 16 \\ a_{n-1,1} & b_{n-1,2} & \dots & b_{n-1,n-4} - 56 & a_{n-1,n-3} + 114 & b_{n-1,n-2} - 104 & a_{n-1,n-1} + \alpha \end{bmatrix}, \quad (19)$$

where

$$a_{i,j} = -16\lambda h^3 k_{i,j},$$

$$b_{i,j} = -8\lambda h^3 k_{i,j},$$

$$\alpha = 35 - 12h^2 p_i,$$

$$\beta = 30 + 12h^2 p_i,$$

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_{n-4} \\ y_{n-3} \\ y_{n-2} \\ y_{n-1} \end{bmatrix},$$

$\mathbf{F}$

$$= \begin{bmatrix} 12h^2 f_1 + (4\lambda h^3 k_{1,0}) y_0 + (4\lambda h^3 k_{1,n}) y_n \\ 12h^2 f_2 + (4\lambda h^3 k_{2,0} + 1) y_0 + (4\lambda h^3 k_{2,n}) y_n \\ 12h^2 f_3 + (4\lambda h^3 k_{3,0}) y_0 + (4\lambda h^3 k_{3,n}) y_n \\ 12h^2 f_4 + (4\lambda h^3 k_{4,0}) y_0 + (4\lambda h^3 k_{4,n}) y_n \\ \vdots \\ 12h^2 f_{n-4} + (4\lambda h^3 k_{n-4,0}) y_0 + (4\lambda h^3 k_{n-4,n}) y_n \\ 12h^2 f_{n-3} + (4\lambda h^3 k_{n-3,0}) y_0 + (4\lambda h^3 k_{n-3,n}) y_n \\ 12h^2 f_{n-2} + (4\lambda h^3 k_{n-2,0}) y_0 + (4\lambda h^3 k_{n-2,n} + 1) y_n \\ 12h^2 f_{n-1} + (4\lambda h^3 k_{n-1,0}) y_0 + (4\lambda h^3 k_{n-1,n}) y_n \end{bmatrix}, \quad (20)$$

where  $\mathbf{A}$  is a coefficient matrix,  $\mathbf{F}$  is a given function, and  $\mathbf{Y}$  is an unknown function which is to be determined. Gauss Elimination method will be used in order to solve the system of linear algebraic equations.

### 4. Results and Discussion

The performance of the method described in the previous section was tested through numerical examples of second-order LFI-DEs. In addition, the examples were performed by using algorithm in C language.

Example 1 (Type 1).

$$y''(x) = x - 2 + 60 \int_0^1 (x-t)y(t) dt, \quad 0 \leq x \leq 1, \quad (21)$$

with the boundary condition

$$\begin{aligned} y(0) &= 0, \\ y(1) &= 0 \end{aligned} \quad (22)$$

and the exact solution

$$y(x) = x(x-1)^2. \quad (23)$$

Example 2 (Type 1).

$$y''(x) = e^x - x + \int_0^1 xty(t) dt, \quad 0 \leq x \leq 1, \quad (24)$$

with the boundary condition

$$\begin{aligned} y(0) &= 0, \\ y(1) &= e^1 \end{aligned} \quad (25)$$

and the exact solution

$$y(x) = e^x. \quad (26)$$

Example 3 (Type 2).

$$y''(x) = 9y(x) + \frac{e^{-15} - 1}{3} + \int_0^5 y(t) dt, \quad 0 \leq x \leq 5, \quad (27)$$

with the boundary condition

$$\begin{aligned} y(0) &= 0, \\ y(5) &= e^{-15} \end{aligned} \quad (28)$$

and the exact solution

$$y(x) = e^{-3x}. \quad (29)$$

Notations used in Tables 1, 2, and 3 are as follows:

HSGS (CD-RT): half-sweep quadrature-difference schemes (central difference, Repeated Trapezoid), [12].

HSGS (CD-RS): half-sweep quadrature-difference schemes (central difference, Repeated Simpson), [12].

B-Spline: Cubic B-Spline collocation method, [6].

DiffTrans: differential transformation method, [7].

TABLE 1: Comparison between CSFD2 (GE), HSGS (CD-RT), and HSGS (CD-RS) for Example 1.

x	CSFD2 (GE)		HSGS (CD-RS)	HSGS (CD-RT)
	N = 24	N = 48	N = 24	N = 24
0.125	$1.51 \times 10^{-6}$	$9.45 \times 10^{-8}$		
0.250	$2.26 \times 10^{-6}$	$1.41 \times 10^{-7}$	$3.25 \times 10^{-6}$	$2.89 \times 10^{-3}$
0.375	$2.41 \times 10^{-6}$	$1.51 \times 10^{-7}$		
0.500	$2.13 \times 10^{-6}$	$1.33 \times 10^{-7}$	N = 48	N = 48
0.625	$1.58 \times 10^{-6}$	$9.87 \times 10^{-8}$		
0.750	$9.31 \times 10^{-7}$	$5.82 \times 10^{-8}$	$2.41 \times 10^{-7}$	$7.91 \times 10^{-4}$
0.875	$3.49 \times 10^{-7}$	$2.18 \times 10^{-8}$		

TABLE 2: Comparison between CSFD2 (GE), DiffTrans, and B-Spline method for Example 2.

x	CSFD2 (GE)		DiffTrans	B-Spline
	N = 10	N = 20	N = 10	N = 10
0.1	$4.31 \times 10^{-5}$	$1.21 \times 10^{-6}$	$0.00 \times 10^{+0}$	$4.39 \times 10^{-6}$
0.2	$3.07 \times 10^{-5}$	$6.82 \times 10^{-7}$	$4.63 \times 10^{-4}$	$1.81 \times 10^{-5}$
0.3	$1.43 \times 10^{-5}$	$1.41 \times 10^{-7}$	$1.43 \times 10^{-3}$	$4.24 \times 10^{-5}$
0.4	$2.37 \times 10^{-6}$	$4.02 \times 10^{-7}$	$2.94 \times 10^{-3}$	$7.83 \times 10^{-5}$
0.5	$1.91 \times 10^{-5}$	$9.45 \times 10^{-7}$	$5.05 \times 10^{-3}$	$1.27 \times 10^{-4}$
0.6	$3.59 \times 10^{-5}$	$1.49 \times 10^{-6}$	$7.81 \times 10^{-3}$	$1.91 \times 10^{-4}$
0.7	$5.28 \times 10^{-5}$	$2.04 \times 10^{-6}$	$1.13 \times 10^{-2}$	$2.70 \times 10^{-4}$
0.8	$6.93 \times 10^{-5}$	$2.59 \times 10^{-6}$	$1.55 \times 10^{-2}$	$3.67 \times 10^{-4}$
0.9	$7.94 \times 10^{-5}$	$3.12 \times 10^{-6}$	$2.05 \times 10^{-2}$	$4.85 \times 10^{-4}$

Tau method: Tau method with piecewise approximate solution of degree 5, [9].

CSFD2 (GE): the proposed method, Composite Simpson's 1/3 with second-order finite-difference method (Gauss Elimination).

Examples 1, 2, and 3 are solved by using the proposed method which is CSFD2 (GE) and the numerical results are shown in Tables 1, 2, and 3. In Table 1, the same methods are used to compare the absolute errors using a different number of subintervals. The maximum absolute errors of the proposed method also compared with the existing methods such as HSGS (CD-RT) and HSGS (CD-RS). As shown in Table 1, when the number of subintervals increases, the absolute errors become smaller. The results for the proposed method are more accurate compared to the HSGS (CD-RT) while the proposed method is slightly better compared to HSGS (CD-RS).

In Table 2, the numerical results of CSFD2 (GE) for Example 2 are compared with the results from differential transformation method and Cubic B-Spline method. CSFD2 (GE) gives more accurate results than the other two existing methods. In Table 3, the maximum absolute errors for CSFD2 (GE) for Example 3 are slightly better compared to Tau method.

### 5. Conclusion

The new combinations of Composite Simpson's 1/3 rule and second-order finite-difference method with Gauss Elimination CSFD2 (GE) were implemented in this study in

TABLE 3: Comparison between CSFD2 (GE) and Tau method for Example 3.

x	CSFD2 (GE)		Tau	
	N = 10	N = 20	N = 10	N = 20
0.5	$1.28 \times 10^{-1}$	$3.88 \times 10^{-2}$		
1.0	$3.47 \times 10^{-2}$	$6.56 \times 10^{-2}$		
1.5	$2.83 \times 10^{-3}$	$9.48 \times 10^{-4}$		
2.0	$5.30 \times 10^{-3}$	$2.64 \times 10^{-3}$		
2.5	$7.20 \times 10^{-3}$	$3.03 \times 10^{-3}$	$8.31 \times 10^{-1}$	$4.71 \times 10^{-2}$
3.0	$7.64 \times 10^{-3}$	$3.11 \times 10^{-3}$		
3.5	$7.73 \times 10^{-3}$	$3.10 \times 10^{-3}$		
4.0	$7.70 \times 10^{-3}$	$3.01 \times 10^{-3}$		
4.5	$7.08 \times 10^{-3}$	$2.57 \times 10^{-3}$		

order to solve second-order LFI-DEs for Types 1 and 2. In conclusion, this study shows that CSFD2 (GE) calculates better approximations to the exact solution of second-order LFI-DEs. Therefore, we introduce CSFD2 (GE) as a simple and more accurate method. In future research, the concept of the proposed method can also be used for Types 3 and 4 of second-order LFI-DEs and also for higher order of LFI-DEs problems. Moreover, the accuracy and efficiency of the proposed method also can be improved.

### Competing Interests

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

### Acknowledgments

The authors gratefully acknowledge the financial support given from Malaysian Ministry of Education for MyBrain15 (My Master) Scholarship and Graduate Research Fund (GRF) by Universiti Putra Malaysia.

### References

- [1] A. R. Vahidi, E. Babolian, G. Asadi Cordshooli, and Z. Azimzadeh, "Numerical solution of Fredholm integro-differential equation by Adomian's decomposition method," *International Journal of Mathematical Analysis*, vol. 3, no. 33–36, pp. 1769–1773, 2009.
- [2] E. Aruchunan and J. Sulaiman, "Numerical solution of second-order linear fredholm integro-differential equation using generalized minimal residual method," *American Journal of Applied Sciences*, vol. 7, no. 6, pp. 780–783, 2010.
- [3] H. Danfu and S. Xufeng, "Numerical solution of integro-differential equations by using CAS wavelet operational matrix of integration," *Applied Mathematics and Computation*, vol. 194, no. 2, pp. 460–466, 2007.
- [4] J. Zhao and R. M. Corless, "Compact finite difference method for integro-differential equations," *Applied Mathematics and Computation*, vol. 177, no. 1, pp. 271–288, 2006.
- [5] M. T. Rashed, "Lagrange interpolation to compute the numerical solutions of differential, integral and integro-differential equations," *Applied Mathematics and Computation*, vol. 151, no. 3, pp. 869–878, 2004.
- [6] N. Ebrahimi and J. Rashidinia, "Spline collocation for Fredholm and Volterra integro-differential equations," *International Journal of Mathematical Modelling & Computations*, vol. 4, no. 3, pp. 289–298, 2014.
- [7] P. Darania and A. Ebadian, "A method for the numerical solution of the integro-differential equations," *Applied Mathematics and Computation*, vol. 188, no. 1, pp. 657–668, 2007.
- [8] R. Saadati, B. Raftari, H. Adibi, S. M. Vaezpour, and S. Shakeri, "A comparison between the Variational Iteration method and Trapezoidal rule for solving linear integro-differential equations," *World Applied Sciences Journal*, vol. 4, no. 3, pp. 321–325, 2008.
- [9] S. M. Hosseini and S. Shahmorad, "Numerical piecewise approximate solution of Fredholm integro-differential equations by the Tau method," *Applied Mathematical Modelling*, vol. 29, no. 11, pp. 1005–1021, 2005.
- [10] S. Yalcinbas, M. S. Sezer, and H. H. Sorkun, "Legendre polynomial solutions of high-order linear Fredholm integro-differential equations," *Applied Mathematical Model*, vol. 27, pp. 145–154, 2003.
- [11] A. I. Fedotov, "Quadrature-difference methods for solving linear and nonlinear singular integro-differential equations," *Nonlinear Analysis*, vol. 71, no. 12, pp. e303–e308, 2009.
- [12] E. Aruchunan and J. Sulaiman, "Half-sweep quadrature-difference schemes with iterative method in solving linear fredholm integro-differential equations," *Progress in Applied Mathematics*, vol. 5, no. 1, pp. 11–21, 2013.





# Hindawi

Submit your manuscripts at  
<https://www.hindawi.com>

