Malaysian Journal of Mathematical Sciences 16(1): 105–117 (2022)
https://doi.org/10.47836/mjms.16.1.09

Malaysian Journal of Mathematical Sciences

Journal homepage: https://einspem.upm.edu.my/journal

Refinement of SOR Iterative Method for the Linear Rational Finite Difference Solution of Second-Order Fredholm Integro-Differential Equations

Xu, M. M.*[1,2], Sulaiman, J.[2], and Ali, L. H.[2]

[1] *School of Mathematics and Information Technology, Xingtai University, China*
[2] *Faculty of Science and Natural Resources, Universiti Malaysia Sabah, Malaysia*

E-mail: xmmzg@sina.com
**Corresponding author*

Received: 13 July 2021
Accepted: 7 October 2021

Abstract

The primary objective of this paper is to develop the Refinement of Successive Over-Relaxation (RSOR) method based on a three-point linear rational finite difference-quadrature discretization scheme for the numerical solution of second-order linear Fredholm integro-differential equation (FIDE). Besides, to illuminate the superior performance of the proposed method, some numerical examples are presented and solved by implementing three approaches which are the Gauss-Seidel (GS), the Successive Over-Relaxation (SOR) and the RSOR methods. Lastly, through the comparison of the results, it is verified that the RSOR method is more effective than the other two methods, especially when considering the aspects of the number of iterations and running time.

Keywords: second-order Fredholm integro-differential equations; linear rational finite difference schemes; composite trapezoidal scheme; refinement successive over-relaxation iterative method.

1 Introduction

Integro-differential equations (IDEs) play critical roles in a great deal fields, for instance neural networks [12], algorithm [15], wireless sensor networks [16], constrained problem [17], and others. IDEs is an equation that contains not only the derivative of the unknown function but also the integral of the unknown function. Generally, IDEs can be classified into the Fredholm integro-differential equation (FIDE) and the Volterra integro-differential equation (VIDE), where the upper bound of the region for the integral term of the former type is a fixed number, while it is a variable for that of the latter type. Apart from the linear integral or integro-differential equations, many studies have also investigated the nonlinear integral or integro-differential equations by several researchers, which proposed several methods that can be used. For instance, the generalized Kudryashov method [11] has been successfully applied to solve and find the solution of nonlinear integro-partial differential equations. Also the rationalized Haar wavelet method [9] has been used to solve the nonlinear integro-differential equation in complex plane.

In practical problems, a large number of linear and nonlinear integro-differential equations problems are related to solving equations, whereas, it is difficult to solve these equations in most cases, especially analytically. For this cause, numerous computational techniques have been devoted to finding the most attractive numerical solutions. In the present paper, we mainly immerse in solving numerical solution of second-order linear FIDE:

$$y''(t) = p(t)y'(t) + q(t)y(t) + r(t) + \int_a^b K(t,u)y(u)du, \quad a \le t \le b, \tag{1}$$

with two-point boundary conditions $y(a) = y_a$, $y(b) = y_b$, where the functions $p(t)$, $q(t)$, $r(t)$ and the kernel $K(t,u)$ are known, a and b are constants, but the $y(t)$ is an unknow function and the solution to be determined.

In recent years, plenteous works have concentrated on the investigation of efficient numerical methods for FIDE, for instance, parameterization method [6], exponential spline method [13], multiscale Galerkin method [4] and the fixed point method [7]. Nevertheless, to the best of our knowledge, there are no references to the research of FIDE with a linear rational finite difference (LRFD) discretization scheme deliberated [14], which allows us to conduct pioneering investigations into this mathematical formula. The LRFD formula is a method to approximate the derivative of the given function by using the derivative of the linear barycentric rational interpolation (LBRI) function [10], which has many advantages, such as good convergence and stability, mainly when calculating the one-sided derivative near the endpoint of the interval. It is much more stable than classical finite difference method. Therefore, a great number of studies have been carried out applying the LRFD method to find the numerical solution of VIDEs [2], delay VIDEs [1] and stiff ODEs [3]. In the current paper, we consider three-point newly established linear rational finite difference (3LRFD) formulas are established and combined with the compound trapezoidal (CT) scheme to discretize the differential term and integral term of second-order linear FIDE, respectively. The corresponding three-point newly established linear rational finite difference-quadrature approximation equations, which can be derived, generate the large-scale and dense system of established linear rational finite difference-quadrature approximation equations.

We all know that there are direct methods and iterative methods to solve the linear system. Simultaneously, it is intricate to apply the direct methods to get the exact solution of the linear system due to the complexity of the coefficient matrix for plenty of practical problems. For this reason, we mainly implement iterative methods to achieve the numerical solution of a linear system. Considering that the SOR iterative method has the advantage of a flexible selection of relaxation factor values, we improved the SOR iterative method again and obtained the refined iterative methods which were discussed by [8] and [20]. In the end, we investigate the performance of the RSOR iterative method [20] together with the three-point newly established LRFD (3LRFD) formula to obtain the numerical solution of the linear system, which is generated by the corresponding three-point linear rational finite difference-quadrature approximation equations.

After the present introduction. In Section 2, we present the three-point newly established LRFD (3LRFD) formula, the CT formula and derive the three-point linear rational finite difference-quadrature approximation equations for second-order linear FIDE. In Section 3, we show in detail the process of solving a linear system by applying the RSOR iterative method. A large number of numerical experiments have been carried out in Section 4 to clarify the feasibility of the constructed methods in this paper. The conclusions and future work are summarized in Section 5.

2 Derivation of Linear Rational Finite Difference-Quadrature Approximation Equations

As mentioned in the previous section, our primary purpose is to solve the numerical solution of Equation (1). It is evident that Equation (1) contains the differential and the integral terms. In this section, we mainly construct the three-point linear rational finite difference-quadrature scheme to discretize the differential term and the integral term of Equation (1) to generate the corresponding approximation equations. This scheme is a combination of 3LRFD formulas and CT formula. Hence in the following two subsections, we will introduce the 3LRFD formulas and CT formula, respectively.

2.1 Three-Point Linear Rational Finite Difference Formulas

In this subsection, we attempt to construct the 3LRFD formula, which is mainly used to approximate $y'(t)$ and $y''(t)$ in Equation (1).

We firstly divide the solution domain, $[a, b]$ of Equation (1) into N subintervals of an equal step length $h = (b-a)/N, t_i = u_i = a + ih, i = 0, 1, \cdots, N$. In our study, the value of N is given by $N = 2^p, p \geq 1$. Now, we review the LRFD formulas derived from LBRI function. Let $t_0, t_1, \cdots, t_m$ be $(m+1)$ interpolation nodes and $y(t_0), y(t_1), \cdots, y(t_m)$ corresponding values. The LBRI function to these data will be an expressed

$$Y_m(t) = \sum_{j=0}^{m} \frac{\frac{\xi_j}{t-t_j} y(t_j)}{\sum\limits_{j=0}^{m} \frac{\xi_j}{t-t_j}}. \tag{2}$$

A specific expression of weights was given by [10]. For nodes of equal step size, the weights schemes are as follows:

$$\xi_j = \frac{(-1)^{j-d}}{2^d} \sum_{s \in J_j} \binom{d}{j-k}, J_j := (s \in 0, 1, \cdots, m-d : j-d \leq s \leq j). \tag{3}$$

An exemplary application of a LBRI function is to approximate the derivative of the given function were discussed by [3]. In fact, the first derivative and second derivative of the LBRI function and assessing them at the point $t_0, t_1, \cdots, t_m$, lead to the formula of LRFD formulas to approximate

$y'(t)$ and $y''(t)$ as follow:

$$y'(t_i) \approx Y'_m(t_i) = \frac{1}{h}\sum_{j=0}^{m} D^{(1)}_{m_{i,j}} y(t_j), \quad i = 0, 1, \cdots, m, \tag{4}$$

and

$$y''(t_i) \approx Y''_m(t_i) = \frac{2}{h^2}\sum_{j=0}^{m} D^{(2)}_{m_{i,j}} y(t_j), \quad i = 0, 1, \cdots, m, \tag{5}$$

with

$$D^{(1)}_{m_{i,j}} = \begin{cases} \frac{1}{i-j}\frac{\xi_j}{\xi_i}, & j \neq i, \\ -\sum\limits_{l=0; l\neq i}^{m} D^{(1)}_{m_{i,l}}, & j = i, \end{cases} \tag{6}$$

and

$$D^{(2)}_{m_{i,j}} = \begin{cases} \frac{1}{i-j}\left(\frac{\xi_j}{\xi_i} D^{(1)}_{m_{i,j}} - D^{(1)}_{m_{i,i}}\right), & j \neq i, \\ -\sum\limits_{l=0; l\neq i}^{m} D^{(2)}_{m_{i,l}}, & j = i. \end{cases} \tag{7}$$

Based on the approximation idea of Equations (2)-(7), and reducing the calculation complexity, the local approximation formulas of the first derivative and second derivative of $y(t)$ at any point $t_i, i = 1, 2, \cdots, N-1$ are given, and the following are known as the 3LRFD formulas:

$$y'(t_i) = Y'(t_i) + e^{(1)}(t_i), \tag{8}$$

and

$$y''(t_i) = Y''(t_i) + e^{(2)}(t_i), \tag{9}$$

in which

$$Y'(t_i) = \frac{1}{h}\sum_{j=i-1}^{i+1} D^{(1)}_{i,j} y(t_j), \tag{10}$$

and

$$Y''(t_i) = \frac{2}{h^2}\sum_{j=i-1}^{i+1} D^{(2)}_{i,j} y(t_j), \tag{11}$$

where

$$D^{(1)}_{i,j} = \begin{cases} \frac{1}{i-j}\frac{\xi_{i,j}}{\xi_{i,i}}, & j \neq i, \\ -(D^{(1)}_{i,i-1} + D^{(1)}_{i,i+1}), & j = i, \end{cases} \tag{12}$$

and

$$D^{(2)}_{i,j} = \begin{cases} \frac{1}{i-j}\left(\frac{\xi_{i,j}}{\xi_{i,i}} D^{(1)}_{i,i} - D^{(1)}_{i,j}\right), & j \neq i, \\ -(D^{(2)}_{i,i-1} + D^{(2)}_{i,i+1}), & j = i. \end{cases} \tag{13}$$

In this work, the 3LRFD formulas will be implemented to discretize the $y'(t_i)$ and $y''(t_i)$ of Equation (1) to derive the three-point linear rational finite difference-quadrature approximation equations for Equation (1). Here, we primarily concentrated on $d = 1$, and the values of $\xi_{i,j}$, $D^{(1)}_{i,j}$ and $D^{(2)}_{i,j}$ can be shown in Table 1, then the error accuracy can be acquired from [14] as $|e^{(1)}(t_i)| = O(h), |e^{(2)}(t_i)| = C$. ($C$ is a constant)

2.2 Composite Trapezoidal Quadrature Formula

Consistent with the previous subsection of discretizing differential terms of Equation (1), in this subsection, we mainly introduce the CT scheme from the family of compound quadrature methods to discretize the integral term and then construct three-point linear rational finite difference-quadrature discretization scheme approximation equations. In general, the quadrature scheme can be expressed as:

$$\int_a^b K(t,u)y(u)du = \sum_{j=0}^{N} B_j K(t,u_j)y(u_j) + \delta_N(y), \tag{14}$$

where

$$\begin{aligned} N &: \text{the number of subintervals;} \\ u_j &: \text{the u-coordinate of the point of interval;} \\ B_j &: \text{the numerical coefficients;} \\ \delta_N(y) &: \text{the truncation errors.} \end{aligned}$$

In order to establish the approximation equation of Equation (1), we mainly consider the CT formula. Hence, the B_j of the CT formula are as follows:

$$B_j = \begin{cases} \frac{1}{2}h, & j = 0, N, \\ h, & \text{others.} \end{cases} \tag{15}$$

Combined with the 3LRFD formula constructed in subsection 2.1, the Equations (8)-(15) are Substituted into Equation (1), and the general formula of the three-point linear rational finite difference-compound trapezoidal (3LRFD-CT) approximation equations can be constructed as follows:

$$\frac{2}{h^2}\sum_{j=i-1}^{i+1} D_{i,j}^{(2)} y_j = \frac{1}{h} p_i \sum_{j=i-1}^{i+1} D_{i,j}^{(1)} y_j + q_i y_i + r_i + \sum_{j=0}^{N} B_j K_{i,j} y_j, i = 1, 2, \cdots, N-1, \tag{16}$$

where

$$p_i = p(t_i), q_i = q(t_i), r_i = r(t_i), K_{i,j} = K(t_i, u_j), y_i = y(t_i).$$

By approximating Equation (16), we can quickly establish the corresponding linear system as follows:

$$Ay = b, \tag{17}$$

where $A = \widetilde{A}^T \widetilde{A}, b = A^T \widetilde{b}$,

$$y = \left[y_1, y_2, \cdots, y_{N-2}, y_{N-1}\right]^T,$$

$$b = \left[b_1, b_2, \cdots, b_{N-2}, b_{N-1}\right]^T,$$

$$\widetilde{b} = \begin{bmatrix} r_1 + \frac{1}{2}hK_{1,0}y_0 + \frac{1}{2}hK_{1,0}y_N - \frac{1}{2h}p_1y_0 - \frac{1}{h^2}y_0 \\ r_2 + \frac{1}{2}hK_{2,0}y_0 + \frac{1}{2}hK_{2,0}y_N \\ \vdots \\ r_{N-2} + \frac{1}{2}hK_{N-2,0}y_0 + \frac{1}{2}hK_{N-2,0}y_N \\ r_{N-1} + \frac{1}{2}hK_{N-1,0}y_0 + \frac{1}{2}hK_{N-1,0}y_N - \frac{1}{2h}p_{N-1}y_{N-1} - \frac{1}{h^2}y_{N-1} \end{bmatrix}_{(N-1)\times 1}.$$

Consequently, the first step of the discrete process based on the 3LRFD-CT discretization scheme has been completed, and the linear system (17) corresponding to the approximation equation (16) is generated. In the next section, we continue with the second step, which is to solve obtain the numerical solution of the linear system (17) that was generated in this section.

3 The Formula of Refinement of Successive Over-Relaxation Iterative Method

We now turn our attention to finding numerical solutions for the linear system (17). Usually, when having the coefficient matrix of the low-order dense matrix, the direct method should be used to solve the linear system. However, it can be observed that the main characteristic of the coefficient matrix for the linear system (17) is a large-scale and dense matrix, so it is very challenging to use direct methods to calculate the exact solution. This study mainly implements the RSOR iterative method to obtain the numerical solution of the linear system (17). The RSOR method is chose because the RSOR iterative method has the advantage of choosing the value of the relaxation factor flexibly compared with the GS method. However, the RSOR method has a faster convergence rate than the SOR iterative method.

Before constructing the formula of the RSOR method, let us decompose the coefficient matrix A as the summation of three matrices, which is expressed as follows

$$A = E - F - G, \tag{18}$$

where

E : the diagonal matrix;
F : the strictly lower triangular matrix;
G : the strictly upper triangular matrices.

Therefore, the general formulate for the SOR method can be shown as follows

$$y^{(k+1)} = (E - \omega F)^{-1}((1-\omega)E + \omega G)y^{(k)} + \omega(E - \omega F)^{-1}b, \tag{19}$$

where ω is the relaxation factor. k is the number of iterations. As taking $\omega = 1$, Equation (19) can be reduced as the standard GS iteration method. In this work, the GS and SOR iterative method are assigned as the control method.

Then, we will continue to derive how to obtain the RSOR iterative method and substitute Equation (18) into Equation (17) to get

$$(E - F - G)y = b.$$

Thus,

$$y = y + \omega(E - \omega F)^{-1}(b - Dy). \tag{20}$$

Referring to Equation (20). , the general formula for the RSOR method can be stated as

$$y^{(k+1)} = y^{(k+1)} + \omega(E - \omega F)^{-1}(b - Dy^{(k+1)}). \tag{21}$$

From the above equation, it can be clearly observed that $y^{(k+1)}$ appears on both sides of Equation (21), so let us take the place of the $y^{(k+1)}$ on the right side of Equation (21) with Equation (19) and lead to the RSOR iterative method

$$y^{(k+1)} = ((E-\omega F)^{-1}((1-\omega)E+\omega G))^2 y^{(k)} + \omega(I+(E-\omega F)^{-1}((1-\omega)E+\omega G))b(E-\omega F)^{-1}, \tag{22}$$

where the values of matrices and are determined as stated in Equation (18). Based on formulation of the RSOR method in (22), the convergence analysis of this method has been discussed by [20]. In getting the numerical solution, the RSOR iterative method is employed to solve the system of linear equations iteratively with the approximate solution to the vector $y^{(k)} = [y_1^{(k)}, y_2^{(k)}, \cdots, y_{N-1}^{(k)}]^T$. The iterative process of the RSOR iterative method can be described as in Algorithm 1 which is achieved by applying MATLAB software.

Algorithm 3.1. *RSOR iterative method*

1. *Initializing all the terms. Set* $k = 0$, *and* $y_i^{(0)} = 0, i = 1, 2, \cdots, N-1$.
2. *For* $k = 1, 2, 3, \cdots$, *perform*
   - *i) Calculate*
     $y^{(k+1)} = ((E - \omega F)^{-1}((1-\omega)E + \omega G))^2 y^{(k)} + \omega(I + (E - \omega F)^{-1}((1-\omega)E + \omega G))b(E - \omega F)^{-1}$.
   - *ii) Check the convergence test. If the error of tolerance* $\|y^{(k+1)} - y^{(k)}\| \leq \theta = 10^{-10}$ *is satisfied, then go step* c).
3. *Display the numerical solution.*
4. *Stop.*

4 Numerical Experiments

In this second and third sections, we theoretically introduce the methods of solving the numerical solution of second-order linear FIDE and the solution process. In this section, three examples of second-order linear FIDE are inspected to more clearly elucidate the efficiency of the RSOR iterative method based on the 3LRFD-CT discretization scheme for solving numerical solution. All numerical experiments were conducted by using the CPU processor Intel (R) i5 2.11GHz and run via MATLAB software.

Example 1: [19] Consider the linear FIDE of second-order

$$y''(t) = 32t + \int_{-1}^{1} (1 - tu)y(u)du, \quad -1 \leq t \leq 1,$$

with two-point boundary condition $y(-1) = -\frac{5}{2}, y(1) = \frac{15}{2}$, and exact solution is $y(t) = 5t^3 + \frac{3}{2}t^2 + 1$.

Example 2: [21] Consider the linear FIDE of second-order

$$y''(t) = 2 - \frac{16}{15}t - \frac{16}{15}t^2 + \int_{-1}^{1} (tu^2 - t^2u^2)y(u)du, \quad -1 \leq t \leq 1,$$

with two-point boundary condition $y(0) = 1, y(1) = 3$, and exact solution is $y(t) = t^2 + t + 1$.

Example 3: [21] Consider the linear FIDE of second-order

$$y''(t) = e^t - t + \int_{0}^{1} tuy(u)du, \quad 0 \leq t \leq 1,$$

with two-point boundary condition $y(0) = 1, y(1) = e$, and exact solution is $y(t) = e^t$.

For the sake of comparison, the classical GS and SOR methods also are presented, which act as the control method of numerical experiments. At the same time, the three parameters of the number of iterations (**Iterations**), the running time (**Time**) in seconds and the maximum values of absolute errors (**Error**) obtained from the application of GS, SOR and RSOR iterative methods based on the 3LRFD-CT scheme are taken into account. For Examples 1 to 3, various numerical experiments have been carried out and the number of subintervals to be considered are 32, 64, 128, 256 and 512, and the results are displayed in Tables 2-4.

Referring to Tables 2-4, it can be seen that compared with the GS iterative method, the other two methods, such as SOR and RSOR, have fewer Iterations, faster Time and higher accuracy. Especially with the increase of the number of subintervals, the gap becomes more and more remarkable, and the latter two are better than GS iteration. We specifically calculated the percentage reductions of the first two parameters obtained by using the SOR and RSOR iterative methods compared with the values obtained by using the GS iterative method, which is as high as about 99%, shown in Table 5. Furthermore, comparing between SOR and RSOR iterative methods, they both have high accuracy. However, for the first two parameters, namely Iteration and Time, the RSOR iterative method is meaningfully better than the SOR iterative method. In conclusion, there is no doubt that the RSOR iterative method based on the 3LRFD-CT discretization scheme is the most effective among the three methods.

5 Conclusion

In the current paper, the perfomance of the RSOR iterative method based on the three-point newly established LRFD-CT discretization scheme for the numerical solution of Eq. 1 has been successfully investigated to solve the generated linear system based on the numerical results and compared with the GS and SOR iterative methods, implementations of the proposed RSOR method together with the three-point newly established LRFD (3LRFD) formula have provided two main advantages: it has the smallest number of iterations, another considerable advantage is that the running time of the method is very short. Finally, illustrative examples are presented to manifest the effectiveness of the proposed method for dealing with second-order linear FIDE. Due to the benefit of the RSOR iteration family, which is classed as one of the efficient point iteration families, this analysis can be generalised to interact with the implementation of block point iteration approaches by [18], and two-parameter iteration approaches. Other than these iteration families, the concept of the RSOR method can be applied to two-step iteration families, such as AGE by [5] , AM. Apart from iteration families, this study should be extended to solve nonlinear Fredholm integro-differential equations [22] numerically by using the proposed iterative method.

Acknowledgement We would like to extend heartfelt gratitude to the anonymous reviewer of this paper. We also would like to express our deep appreciation for the financial support by Universiti Malaysia Sabah for funding this research under the UMSGreat research grant for postgraduate students: GUG0513-2/2020.

Conflicts of Interest The authors declare no conflict of interest.

References

[1] A. Abdi, J. P. Berrut & S. A. Hosseini (2018). The linear barycentric rational method for a class of delay Volterra integro-differential equations. *Journal of Scientific Computing*, *75*(3), 1755–1775.

[2] A. Abdi & S. A. Hosseini (2018). The barycentric rational difference-quadrature scheme for systems of Volterra integro-differential equations. *SIAM Journal on Scientific Computing*, *40*(3), A1936–A1960. https://doi.org/10.1137/17M114371X.

[3] A. Abdi, S. A. Hosseini & H. Podhaisky (2019). Adaptive linear barycentric rational finite differences method for stiff ODEs. *Journal of Computational and Applied Mathematics*, *357*, 204–214.

[4] J. Chen, M. He & Y. Huang (2020). A fast multiscale Galerkin method for solving second order linear Fredholm integro-differential equation with Dirichlet boundary conditions. *Journal of Computational and Applied Mathematics*, *364*, 112352. https://doi.org/10.1016/j.cam.2019.112352.

[5] A. A. Dahalan, J. Sulaiman & M. S. Muthuvalu (2014). Performance of HSAGE method with Seikkala derivative for 2-D fuzzy Poisson equation. *Applied Mathematical Sciences*, *8*(17-20), 885–899.

[6] D. S. Dzhumabaev (2016). On one approach to solve the linear boundary value problems for Fredholm integro-differential equations. *Journal of Computational and Applied Mathematics*, *294*, 342–357.

[7] M. Efendiev & V. Vougalter (2021). Existence of solutions for some non-Fredholm integro-differential equations with mixed diffusion. *Journal of Differential Equations*, *284*, 83–101.

[8] T. K. Enyew, G. Awgichew, E. Haile & G. D. Abie (2020). Second-refinement of Gauss-Seidel iterative method for solving linear system of equations. *Ethiopian Journal of Science and Technology*, *13*(1), 1–15.

[9] M. Erfanian & A. Mansoori (2019). Solving the nonlinear integro-differential equation in complex plane with rationalized Haar wavelet. *Mathematics and Computers in Simulation*, *165*, 223–237.

[10] M. S. Floater & K. Hormann (2007). Barycentric rational interpolation with no poles and high rates of approximation. *Numerische Mathematik*, *107*(2), 315–331.

[11] K. A. Gepreel, T. A. Nofal & A. A. Alasmari (2017). Exact solutions for nonlinear integro-partial differential equations using the generalized kudryashov method. *Journal of the Egyptian Mathematical Society*, *25*(4), 438–444.

[12] F. Heydarpour, E. Abbasi, M. J. Ebadi & S. M. Karbassi (2020). Solving an optimal control problem of cancer treatment by artificial neural networks. *International Journal of Interactive Multimedia and Artificial Intelligence*, *6*(4), 18–25.

[13] R. Jalilian & T. Tahernezhad (2020). Exponential spline method for approximation solution of Fredholm integro-differential equation. *International Journal of Computer Mathematics*, *97*(4), 791–801.

[14] G. Klein & J. P. Berrut (2012). Linear rational finite differences from derivatives of barycentric rational interpolants. *SIAM Journal on Numerical Analysis*, *50*(2), 643–656.

[15] H. Liu, J. Huang & W. Zhang (2021). Numerical algorithm based on extended barycentric Lagrange interpolant for two dimensional integro-differential equations. *Applied Mathematics and Computation*, *396*, 125931. https://doi.org/10.1016/j.amc.2020.125931.

[16] A. Rahiminasab, P. Tirandazi, M. J. E. A. Ahmadian & M. Salimi (2020). An energy-aware method for selecting cluster heads in wireless sensor networks. *Applied Sciences*, *10*(21), 7886. https://doi.org/10.3390/app10217886.

[17] M. Rostami, K. Berahmand & S. Forouzande (2020). A novel method of constrained feature selection by the measurement of pairwise constraints uncertainty. *Journal of Big Date*, *7*(83), 1–21.

[18] A. Saudi & J. Sulaiman (2012). Robot path planning using four point-explicit group via nine-point laplacian (4EG9L) iterative method. *Procedia Engineering*, *41*, 182–188.

[19] J. Sunday (2019). On exact finite difference scheme for the computation of second-order Fredholm integro-differential equations. *Fulafia Journal of Science and Technology*, *5*(113-119), 243–247.

[20] V. B. K. Vatti & S. Dominic (2015). A refinement of successive over relaxation (RSOR) method for solving of linear system of equations. *International Journal of Advanced Information Science and Technology*, *40*(40), 1–4.

[21] A. M. Wazwaz (2015). *A first course in integral equations*. World Scientific, USA.

[22] D. Xu (2020). Analytical and numerical solutions of a class of nonlinear integro-differential equations with L1 kernels. *Nonlinear Analysis: Real World Applications*, *51*, 103002. https://doi.org/10.1016/j.nonrwa.2019.103002.

Appendix

Table 1: The values of $\xi_{i,j}$, $D^{(1)}_{i,j}$, $D^{(2)}_{i,j}$,

$i = 1, 2, \cdots, N-1$	$\xi_{i,i-1}$	$\xi_{i,i}$	$\xi_{i,i+1}$
	$-\frac{1}{2}$	1	$-\frac{1}{2}$
	$D^{(1)}_{i,i-1}$	$D^{(1)}_{i,i}$	$D^{(1)}_{i,i+1}$
	$-\frac{1}{2}$	0	$\frac{1}{2}$
	$D^{(2)}_{i,i-1}$	$D^{(2)}_{i,i}$	$D^{(2)}_{i,i+1}$
	$\frac{1}{2}$	-1	$\frac{1}{2}$

Table 2: Comparison of results for two different iteration methods at Example 1.

N	Methods	Iterations	Time	Error
32	GS-3LRFD	185224	0.4624	1.2910E-03
	SOR-3LRFD (ω)	8216 (1.894700000)	0.0262	1.2908E-03
	RSOR-3LRFD (ω)	4274 (1.894120000)	0.0105	1.2908E-03
64	GS-3LRFD	2492458	7.6251	3.2510E-04
	SOR-3LRFD (ω)	58845 (1.943049000)	0.2959	3.2270E-04
	RSOR-3LRFD (ω)	31012 (1.946836000)	0.1209	3.2270E-04
128	GS-3LRFD	32429703	232.2066	1.0914E-04
	SOR-3LRFD (ω)	435012 (1.973894100)	4.2856	8.1130E-05
	RSOR-3LRFD (ω)	226096 (1.973643000)	1.5437	8.0914E-05
256	GS-3LRFD	400325235	6634.2979	4.5008E-04
	SOR-3LRFD (ω)	3139706 (1.987250000)	59.6045	2.2816E-05
	RSOR-3LRFD (ω)	1637449 (1.987084256)	26.3720	2.1434E-05
512	GS-3LRFD	4513359199	190132.3872	4.9842E-03
	SOR-3LRFD (ω)	21559438 (1.992936250)	903.6976	2.2708E-05
	RSOR-3LRFD (ω)	11345038 (1.992892590)	551.8140	1.1088E-05

Table 3: Comparison of results for two different iteration methods at Example 2.

N	Methods	Iterations	Time	Error
32	GS-3LRFD	448234	1.0256	5.3204E-04
	SOR-3LRFD (ω)	9670 (1.9443380000)	0.0327	5.3154E-04
	RSOR-3LRFD (ω)	4996 (1.9440800000)	0.0122	5.3154E-04
64	GS-3LRFD	5959234	27.3561	1.3904E-04
	SOR-3LRFD (ω)	70503 (1.9723800000)	0.8217	1.3323E-04
	RSOR-3LRFD (ω)	36507 (1.9722130000)	0.1201	1.3323E-04
128	GS-3LRFD	76098613	566.8762	1.2114E-04
	SOR-3LRFD (ω)	486033 (1.9853554600)	11.9369	3.2263E-05
	RSOR-3LRFD (ω)	253489 (1.9853204000)	4.8374	3.2759E-05
256	GS-3LRFD	910442625	14873.0139	1.1261E-03
	SOR-3LRFD (ω)	3470593 (1.9927109326)	58.8191	5.6794E-06
	RSOR-3LRFD (ω)	1936709 (1.9927109326)	28.7981	1.1373E-05
512	GS-3LRFD	9658341997	414576.0779	1.2686E-02
	SOR-3LRFD (ω)	26735660 (1.9967859460)	1364.6092	4.7144E-05
	RSOR-3LRFD (ω)	13965124 (1.9967335111)	594.5617	2.3281E-05

Table 4: Comparison of results for two different iteration methods at Example 3.

N	Methods	Iterations	Time	Error
32	GS-3LRFD	461118	1.0987	6.7632E-06
	SOR-3LRFD (ω)	9239 (1.942293800)	0.0305	6.0313E-06
	RSOR-3LRFD (ω)	4785 (1.942223000)	0.0144	6.0523E-06
64	GS-3LRFD	6163295	23.7031	1.0129E-05
	SOR-3LRFD (ω)	66863 (1.971061900)	0.3315	1.2856E-06
	RSOR-3LRFD (ω)	34738 (1.971013800)	0.1444	1.4003E-06
128	GS-3LRFD	79290805	622.6473	9.9643E-05
	SOR-3LRFD (ω)	482119 (1.985530300)	3.6553	1.2249E-06
	RSOR-3LRFD (ω)	264332 (1.986414500)	1.9583	1.1031E-06
256	GS-3LRFD	960096418	15178.6218	1.1235E-03
	SOR-3LRFD (ω)	3685270 (1.993410000)	60.0946	8.4396E-06
	RSOR-3LRFD (ω)	1919339 (1.993336423)	30.8216	4.1840E-06
512	GS-3LRFD	10428247925	441013.7995	1.1207E-02
	SOR-3LRFD (ω)	24259791 (1.996432210)	992.9554	5.3243E-05
	RSOR-3LRFD (ω)	12784226 (1.996414240)	588.9079	2.8628E-05

Table 5: Percentage reductions of the SOR and RSOR iterative methods relative to the GS iterative method in solving Examples 1-3 via applying 3LRFD-CT scheme.

Example	Methods	Iterations	Time
1	SOR-3LRFD	95.57%-99.52%	94.33%-99.52%
	RSOR-3LRFD	97.71%-99.75%	97.73%-99.71%
2	SOR-3LRFD	97.84%-99.72%	96.81%-99.67%
	RSOR-3LRFD	98.89%-99.86%	98.81%-99.86%
3	SOR-3LRFD	97.99%-99.77%	97.22%-99.77%
	RSOR-3LRFD	98.96%-99.88%	98.69%-99.87%