International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V69I6P221
Volume 69 Issue 6, 136-143, June 2021

Linear Rational Finite Difference Approximation for Second-Order Linear Fredholm Integro-Differential Equations Using the Half-Sweep SOR Iterative Method

Ming-Ming Xu[#1], Jumat Sulaiman[2], Labiyana Hanif Ali[3]

[1]*School of Mathematics and Information Technology, Xingtai University, 054000 Xingtai, Hebei, China.*
[1,2,3] *Faculty of Science and Natural Resources, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia.*

[1]xmmzg@sina.com, [2]jumat@ums.edu.com , [3]labiyana15@gmail.com

Abstract — *This paper proposes the hybridization of the three-point half-sweep linear rational finite difference (3HSLRFD) schemes with the half-sweep composite trapezoidal (HSCT) approach to derive the 3HSLRFD-HSCT discretization schemes, in which these discretization schemes are used to derive the corresponding approximation equation for second-order linear Fredholm integro-differential equation. Based on the approximation equation, the related linear system can be generated, in which its coefficient matrix is dense. Furthermore, the half-sweep Successive Over-Relaxation (HSSOR) technique is implemented to find the numerical solution of the linear system. To make a comparison, the full-sweep Gauss-Seidel (FSGS) and the full-sweep Successive Over-Relaxation (FSSOR) techniques are also presented as the control method. In numerical experiments, three parameters like the quantity of iterations, elapsed time and the maximum absolute errors have been recorded via three methods. Lastly, it can be pointed out that the HSSOR technique is more superior to the other two techniques, especially in terms of the quantity of iterations and elapsed time.*

Keywords — *Second-order Integro-differential equations, Half-sweep SOR iterative method, Three-point half-sweep linear rational finite difference scheme, Half-sweep composite trapezoidal scheme.*

I. INTRODUCTION

Integro-differential equations (IDEs) contribute a powerful tool in many branches of natural science and engineering. Many problems in fluid dynamics, finance, physics, astronomy, biology, and so on lead to these equations [1]-[3]. On the other hand, it is pretty troublesome to find the exact solution of the IDEs in plenty of practical problems. That is a reason why considerable works have been focusing on the development of efficient numerical methods for the approximate solutions of IDEs, for instance, Legendre-spectral method [4], Bernstein polynomials method [5], spline collocation methods [6] and modified Adomian decomposition method [7]. In this paper, we fix attention on finding numerical solutions to second-order linear Fredholm integro-differential equation (SOLFIDE):

$$y''(t) = \alpha(t)y'(t) + \beta(t)y(t) + \gamma(t) + \int_a^b K(t,u)y(u)du, \quad (1)$$

$a \le t \le b$, with two-point boundary conditions $y(a) = y_a$, $y(b) = y_b$, where the functions $\alpha(t)$, $\beta(t)$, $\gamma(t)$ and the kernel $K(t,u)$ are known, a and b are constant, but the $y(t)$ is an unknown function that needs to be determined.

Clearly, it can be observed that problem (1) contains the differential term and the integral term. The most classical method to discretize the differential terms is the finite difference (FD) method, which approximates the derivative of the interpolated function by the derivative of the polynomial interpolation function. Apart from the FD method, the linear rational finite difference (LRFD) [8], [9] has aroused much interest among researchers. Compared with the FD method, the LRFD method has better approximation and stability, mainly when calculating the one-sided derivative near the endpoint of the interval. Therefore, in recent years, many researchers apply the LRFD method to solve VIDEs [10], delay VIDEs [11], stiff ODEs [12]. These studies motivate us to implement the extended LRFD method to discretize the differential terms of the SOLFIDE (1). This paper constructs three-point half-sweep LRFD (3HSLRFD) schemes combined with a half-sweep composite trapezoidal (HSCT) approach to discretize the differential term and integral term of the SOLFIDE to generate a linear system. Here, this combination of the 3HSLRFD scheme and the HSCT approach is referred to as the 3HSLRFD-HSCT discretization scheme. The half-sweep (HS) iteration concept will be explained in the next paragraph.

In 1991, Abdullah [13] first proposed the HS iteration concept via the four-point EDG iterative technique for solving the Poisson equation of two-dimensional. Clearly, the author has shown that the main idea of HS iteration concept is to take only half of the quantity of entire points in the solution domain of the proposed problem. Consequently, the HS iteration concept has the potential to reduce computational complexity in the solution procedure, which naturally leads to fewer the quantity of iterations and faster elapsed time. Due to its advantage of economic computation in implementing this concept, several applications of the HS iteration concept have been extensively carried out to investigate its performance of finding the numerical solution for fuzzy boundary value problem [14]-[17], robotic path planning [18], [19], and two-dimensional free space wave propagation problem [20,21]. By considering these advantages into account, further discussion of this paper concentrates on extending the implementation of the HS iteration concept with the Successive Over-Relaxation (SOR) technique, namely HSSOR, to obtain the numerical solution of the linear system, which is generated by applying the corresponded 3HSLRFD-HSCT approximation equation.

The primary intention of the present paper is to seek out the numerical solution to problem (1). The solution process consists of two steps. The primary step is to construct the 3HSLRFD-HSCT discretization scheme approximate equation of the problem (1) in section 2. The second step is to implement the HSSOR iterative method to find the numerical solution of the corresponding approximation equation for the problem (1) in section 3. Section 4 offers many numerical examples to validate the effectiveness of the proposed technique in this paper. Section 5 contains a brief conclusion as well as future work.

II. DERIVATION OF APPROXIMATION EQUATION

This section constructs and applies 3HSLRFD schemes and the HSCT approach to discretize the differential term and integral term of the problem (1) and get the 3HSLRFD-HSCT discretization schemes to derive an approximation equation. Before getting the formulation of the 3HSLRFD-HSCT discretization scheme to derive the approximation equation, the following discussion attempts to explain these described schemes.

Now, we introduce the concept of HS iteration. For the solution domain $[a,b]$ of problem (1), it is divided to into N subintervals of equal step length $h=\frac{b-a}{N}$, $t_i=u_i=a+ih,\ i=0,1,\cdots,N$. In our paper, the value of N is given by $N=2^p, p\geq 1$. Based on the full-sweep (FS) iteration concept, the general solution process takes out all the partition points. Whereas the HS iteration method is that we only take out the even partition points. The distribution of grid points is shown in Fig. 1 a) and b) are FS iteration method and HS iteration method, respectively. Referring to Fig. 1, the FS and HS iteration concepts will only calculate approximate values onto node points (type ●) until the convergence condition is met. After that, the direct method [13] can then be used to calculate other approximate solutions at the other points (type ○). Compared with h for each grid size of the FS iteration, each grid size of the HS iteration is $2h$, so it is logical that the latter is much faster and more computationally economical than the former.

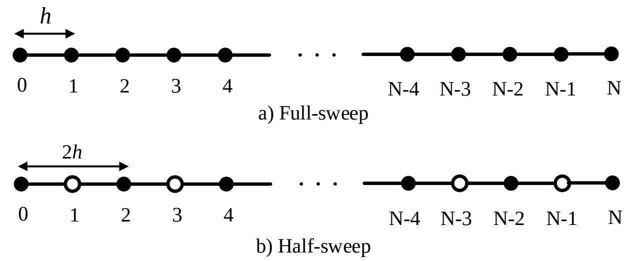

Fig. 1 Distribution of the uniform mesh size for full-sweep and half-sweep cases.

Taking into account the distribution of uniformly node points in the HS case, we begin to combine the HS iteration method alone with LFRD and CT schemes, respectively, to construct both new schemes, namely HSLRFD and HSCT. Further, we attempt to hybridize 3HSLRFD and HSCT discretization schemes to develop a fast and reliable algorithm, finding the numerical solution of problem (1). To do this matter, we need to discuss how to obtain the 3HSLRFD-HSCT approximation equation via both newly established schemes.

A. The Three-Point Half-Sweep Linear Rational Finite Difference Schemes

In this subsection, we try to establish the 3HSLRFD schemes which mainly applied to discretize the $y'(t)$ and $y''(t)$ of problem (1). Firstly, we review the LRFD method, as mentioned in introduction, LRFD method is derived from LBRI.

Let $t_0,t_1,\cdots,t_m$ be $m+1$ genuine abscissas and corresponding values $y(t_0),y(t_1),\cdots y(t_m)$. The LBRI to these data will be expressed as follows:

$$Y_{F_m}(t)=\sum_{j=0}^{m}\left(\frac{\left(\frac{\xi_{F_j}}{t-t_j}\right)y(t_j)}{\left(\sum_{j=0}^{m}\frac{\xi_{F_j}}{t-t_j}\right)}\right) \tag{2}$$

where the weights $\xi_{F_j}, j=0,1,\cdots m.\ (0\leq d\leq m)$ were proposed by Floater and Hormann [22] in 2007. For nodes of equal step size, the weights formulas are

$$\xi_{F_j}=\frac{(-1)^{j-d}}{2^d}\sum_{s\in J_{F_j}}\binom{d}{j-s}, \tag{3}$$

where

$$J_{F_j}=\{s\in\{0,1,\cdots,m-d\}: j-d\leq s\leq j\}.$$

In 2011, Berrut *et al.* [8] introduced the derivative of LBRI, the formulation of LRFD to approximate the $y'(t)$ and $y''(t)$ on $t_0,t_1,\cdots,t_m$ is written as

$$y'(t_i)\approx Y'_{F_m}(t_i)=\frac{1}{h}\sum_{j=0}^{m}\Delta_{F_{i,j}}^{(1)}y(t_j), \tag{4}$$

and

$$y''(t_i)\approx Y''_{F_m}(t_i)=\frac{2}{h^2}\sum_{j=0}^{m}\Delta_{F_{i,j}}^{(2)}y(t_j), \tag{5}$$

where

$$\Delta^{(1)}_{F_{i,j}} = \begin{cases} \frac{\xi_{F_j}}{(i-j)\xi_{F_i}}, & j \neq i, \\ -\sum_{\substack{q=0 \\ q\neq i}}^{m} \Delta^{(1)}_{F_{i,q}}, & j = i. \end{cases} \quad (6)$$

and

$$\Delta^{(2)}_{F_{i,j}} = \begin{cases} \frac{\Delta^{(1)}_{F_{i,i}}\xi_{F_j}}{(i-j)\xi_{F_i}} - \frac{\Delta^{(1)}_{F_{i,j}}}{i-j}, & j \neq i, \\ -\sum_{\substack{q=0 \\ q\neq i}}^{m} \Delta^{(2)}_{F_{i,q}}, & j = i. \end{cases} \quad (7)$$

Based on the idea of the HS iteration concept introduced and the Equations (2)-(3) in this section. The HSLBRI on $t_0, t_2, \cdots, t_m$ (m here and below is even) will be constructed as

$$Y_{H_m}(t) = \sum_{j=0,2}^{m} \left(\frac{\left(\frac{\xi_{H_j}}{t-t_j}\right) y(t_j)}{\left(\sum_{j=0,2}^{m} \frac{\xi_{H_j}}{t-t_j}\right)} \right), \quad (8)$$

where

$$\xi_{H_j} = \frac{(-1)^{\frac{j}{2}-d}}{2^d} \sum_{s\in J_{H_j}} \binom{d}{\frac{j}{2}-s}, \quad (9)$$

which

$$J_{H_j} := \left\{ s \in \left\{0,1,2,\cdots,\frac{m}{2}-d\right\} : \frac{j}{2} - d \leq s \leq \frac{j}{2} \right\}.$$

Similarly, based on the HS iteration concept and Equation (4)-(7), the the schemes of HSLRFD to approximate the $y'(t)$ and $y''(t)$ on $t_0, t_2, \cdots, t_m$ is written as

$$y'(t_i) \approx Y'_{H_m}(t_i) = \sum_{j=0,2}^{m} \frac{1}{h} \Delta^{(1)}_{H_{i,j}}(t_j), \quad (10)$$

and

$$y''(t_i) \approx Y''_{H_m}(t_i) = \sum_{j=0,2}^{m} \frac{2}{h^2} \Delta^{(2)}_{H_{i,j}}(t_j), \quad (11)$$

where

$$\Delta^{(1)}_{H_{i,j}} = \begin{cases} \frac{\xi_{H_j}}{(i-j)\xi_{H_i}}, & j \neq i, \\ -\sum_{\substack{q=0,2 \\ q\neq i}}^{m} \Delta_{H_{i,q}}, & j = i, \end{cases} \quad (12)$$

and

$$\Delta^{(2)}_{H_{i,j}} = \begin{cases} \frac{\Delta^{(1)}_{H_{i,i}}\xi_{H_j}}{(i-j)\xi_{H_i}} - \frac{\Delta^{(1)}_{H_{i,j}}}{i-j}, & j \neq i, \\ -\sum_{\substack{q=0,2 \\ q\neq i}}^{m} \Delta^{(2)}_{H_{i,j}}, & j = i. \end{cases} \quad (13)$$

Compared with HSLBRI Equation (8) and HSLRFD Equations (10)-(11) schemes constructed in this paper, the previous LBRI Equation (2) and LRFD Equation (4)-(5) schemes are also called FSLBRI and FSLRFD, respectively. As taking $m = 2$ for Equation (2), so we need to consider three-nodes to construct the derivation of 3HSLRFD formula.

By observing Equation (8), we can see its interpolation function about $(\frac{m}{2}+1)$ nodes. Now let us consider the interpolation function at the these nodes of $t_{i-2}, t_i, t_{i+2}, i = 2,4,\cdots, N-2$. Meanwhile, combining with Equation (9)-(13), we can quickly derive the 3HSLRFD schemes, whose expression are

$$y'(t_i) = Y'(t_i) + \tau^{(1)}(t_i), \quad i = 2,4,\cdots,N, \quad (14)$$

and

$$y''(t_i) = Y''(t_i) + \tau^{(2)}(t_i), \quad i = 2,4,\cdots,N, \quad (15)$$

where $\tau^{(1)}(t_i)$ and $\tau^{(2)}(t_i)$ are truncation errors.

$$Y'(t_i) = \frac{1}{h} \sum_{j=i-2}^{i+2} \Delta^{(1)}_{i,j} y(t_j), \quad (16)$$

and

$$Y''(t_i) = \frac{2}{h^2} \sum_{j=i-2}^{i+2} \Delta^{(2)}_{i,j} y(t_j), \quad (17)$$

where

$$\Delta^{(1)}_{i,j} = \begin{cases} \frac{\xi_{i,j}}{(i-j)\xi_{i,i}}, & j \neq i, \\ -\left(\Delta^{(1)}_{i,i-2} + \Delta^{(1)}_{i,i+2}\right), & j = i. \end{cases} \quad (18)$$

and

$$\Delta^{(2)}_{i,j} = \begin{cases} \left(\frac{\Delta^{(1)}_{i,i}\xi_{i,j}}{(i-j)\xi_{i,i}} - \frac{\Delta^{(1)}_{i,j}}{(i-j)}\right), & j \neq i, \\ -\left(\Delta^{(2)}_{i,i-2} + \Delta^{(2)}_{i,i+2}\right), & j = i. \end{cases} \quad (19)$$

In this study, we apply the 3HSLRFD schemes in Equations (14)-(19) to discretize $y'(t)$ and $y''(t)$ of problem (1) in order to derive the three-point linear rational finite difference-quadrature approximation equation of Equation (1). we concentrated primarily on the 3HSLRFD at $d = 1$, and the values of $\xi_{i,j}, D^{(1)}_{i,j}$ and $D^{(2)}_{i,j}$ ($i = 2,4,\cdots, N-2$) are shown in Tables 1 and 2. Then the order of error accuracy can be acquired by Berrut *et al.* [8] as $|\tau^{(1)}(t_i)| = O(h)$, $|\tau^{(2)}(t_i)| = C$ where C is a constant.

TABLE 1. The values of $\xi_{i,j}$.

$\xi_{i,i-1}$	$\xi_{i,i}$	$\xi_{i,i+1}$
$-1/2$	1	$-1/2$

TABLE 2. The values of $\Delta^{(1)}_{i,j}$ and $\Delta^{(2)}_{i,j}$.

$\Delta^{(1)}_{i,i-2}$	$\Delta^{(1)}_{i,i}$	$\Delta^{(1)}_{i,i+2}$	$\Delta^{(2)}_{i,i-2}$	$\Delta^{(2)}_{i,i}$	$\Delta^{(2)}_{i,i+2}$
$-1/4$	0	$1/4$	$1/8$	$-1/4$	$1/8$

B. The Half-Sweep Composite Trapezoidal Scheme

In this subsection, we attempt to present the HSCT scheme from family of quadrature methods which is applied to discretize the integral term of problem (1) to construct an approximation equation coincide with differential term in section 2.1. In general, the quadrature scheme can be expressed as

$$\int_a^b y(u)du = \sum_{j=0}^{N} C_{F_j} y(u_j) + \tilde{\delta}_N(y), \quad (20)$$

where C_{F_j} denotes the independent numerical coefficients and $\tilde{\delta}_N(y)$ denotes the truncation error. we consider the composite trapezoidal (CT) scheme to construct the quadrature scheme to derive an approximation equation of Equation (1). As a result, the C_{F_j} based on the CT scheme is as follows

$$C_{F_j} = \begin{cases} \frac{1}{2}h, & j = 0, N, \\ h, & j = 1,2,\cdots,N-1. \end{cases} \quad (21)$$

In our paper, we also call Equation (20) as a full-sweep composite trapezoidal (FSCT) scheme. In contrast with the FSCT scheme of Equation (21), The HSCT approach is obtained by combining the HS iteration method with the CT method as follows [23]-[26]

$$\int_a^b y(u)du = \sum_{j=0,2}^{N} C_j y(u_j) + \delta_N(y), \quad (22)$$

where

$$C_j = \begin{cases} h, & j = 0, N, \\ 2h, & j = 2,4,\cdots,N-2. \end{cases} \quad (23)$$

By substituting Equations (14), (15) and (22) into Equation (1), the general form of the 3HSLRFD-HSCT approximate equation can be constructed as

$$\frac{2}{h^2}\sum_{j=i-2}^{i+2}\Delta_{i,j}^{(2)} y_j = \frac{1}{h}\alpha_i \sum_{j=i-2}^{i+2}\Delta_{i,j}^{(1)} y_j + \beta_i y_i + \gamma_i + \sum_{j=0,2}^{N} C_j K_{i,j} y(u_j), \quad (i = 2,4,\cdots,N-2.) \quad (24)$$

where $\alpha_i = \alpha(t_i), \beta_i = \beta(t_i), \gamma_i = \gamma(t_i)$, $K_{i,j} = K(t_i, t_j)$ and $y_i = y(t_i)$.

The related linear systems can be easily shown as a result of the approximation equation (24).

$$My = F, \quad (25)$$

where $M = \widetilde{M}^T \widetilde{M}$, F=$\widetilde{M}^T \tilde{F}$,

$$y^T = [y_2, y_4, \cdots, y_{N-4}, y_{N-2}],$$

$$\tilde{F}_j = [\tilde{F}_2, \tilde{F}_4, \cdots, \tilde{F}_{N-4}, \tilde{F}_{N-2}]^T = \begin{bmatrix} \gamma_2 + hK_{2,0}y_0 + hK_{2,0}y_N - \frac{1}{4h}\alpha_2 y_0 - \frac{1}{4h^2}y_0 \\ \gamma_4 + hK_{4,0}y_0 + hK_{4,0}y_N \\ \vdots \\ \gamma_{N-4} + hK_{N-4,0}y_0 + hK_{N-4,0}y_N \\ \gamma_{N-2} + hK_{N-2,0}y_0 + hK_{N-2,0}y_N + \frac{1}{4h}\alpha_{N-2}y_N - \frac{1}{4h^2}y_N \end{bmatrix}.$$

So far, the first step of the solution process has been completed, and we have obtained the 3HSLRFD-HSCT discretization schemes for the derivation of approximation equation (24) and use it to generate the corresponding linear system (25). Due to the HSCT discretization scheme for the integral term, it can be observed that the main characteristic of the coefficient matrix M of the linear system (25) is dense matrix.

III. DERIVATION OF THE HALF-SWEEP SUCCESSIVE OVER-RELAXATION TECHNIQUE

In this part, we turn our attention to the second step, which is to find the numerical solution to the linear system (25). There are two approaches for solving the linear system: direct and iterative. The former is suitable for solving the exact solution of a linear system with a low-scale coefficient matrix. Nonetheless, it is known from Section 2 that the coefficient matrix of the linear system (25) is a large-scale and dense matrix. Therefore, the iterative techniques are widely regarded as an appropriate method for such a linear system. At the same time, we already know that the HS iteration method can reduce the iteration complexity and thus accelerate the convergence rate. As a result, we combine the HS iteration method with the SOR iterative technique to produce the HSSOR technique, which we then implement to acquire the numerical solution of the linear system (25).

To start the discussion of constructing the formula of the HSSOR technique, let us decompose the coefficient matrix, M as the summation of three matrices which is expressed as follows

$$M = D - L - U, \quad (26)$$

where D, L and U are matrices that are the diagonal, the strictly lower triangular, and the strictly upper triangular. Therefore, the formulation for the HSSOR technique can be demonstrated [18], [20], [27], [28]

$$(D - \omega L)y^{(k+1)} = \big((1-\omega)D + \omega U\big)y^{(k)} + \omega F, \quad (27)$$

where ω is relaxation factor, k is the number of iterations and $y^{(k)} = \left[y_2^{(k)}, y_4^{(k)}, \cdots, y_{N-2}^{(k)}\right]^T$. As taking $\omega = 1$, Equation (27) can be reduced as the HSGS iterative method.

According to Equation (27), we can get

$$y_i^{(k+1)} = y_i^{(k)} + \frac{\omega\left(F_i - \sum_{j=2}^{i-2} M_{i,j} y_j^{(k+1)} - \sum_{j=i}^{N-2} M_{i,j} y_j^{(k)}\right)}{M_{i,i}}, \quad (i = 2,4,\cdots,N-2.)$$

The HSSOR technique is used to get the numerical answer of the linear system (25). The process of iteration is repeated until the solution falls within a predefined acceptable error bound. By calculating the values of the matrices D, L and U as shown in Equation (26). Algorithm 1 describes the general algorithm for the HSSOR technique for solving the linear system (25) with an approximate solution to the vector $y^{(k)}$. In this case, we use the MATLAB software to run Algorithm 1.

Algorithm 1 HSSOR technique

(a) All of the paramaters are initialised. Set $k = 0$, and $y_i^{(0)} = 0, i = 2,4,\cdots,N-2$.

(b) For $k = 1,2,3,\cdots$, perform

(i). Evaluate

$$y_i^{(k+1)} = y_i^{(k)} + \frac{\omega\left(F_i - \sum_{j=2}^{i-2} M_{i,j} y_j^{(k+1)} - \sum_{j=i}^{N-2} M_{i,j} y_j^{(k)}\right)}{M_{i,i}}, \quad (i = 2,4,\cdots,N-2.)$$

(ii). A convergence test is performed. If the error $\left\| y^{(k+1)} - y^{(k)} \right\| \leq \sigma = 10^{-10}$ is satisfied, proceed to step (c). Otherwise continue to repeat (b).

(c) Display the numerical solution.

(d) Stop.

Based on Algorithm 1, we can calculate that each iteration of the HSSOR method requires $(\frac{1}{4}N^2 + \frac{1}{2}N)$ additions/subtractions and $(\frac{1}{4}N^2 + N)$ multiplications/divisions. However, for FSSOR iterative method $i = 1,2,\cdots N-1$, which requires $(N^2 + N)$ additions/subtractions and $(N^2 + 2N)$ multiplications/divisions, as shown in Table 3. Compared with the FSSOR method, the HSSOR method significantly reduces its computational complexity.

TABLE 3. Arithmetic operations per iteration for FSSOR and HSSOR in solving SOLFIDE.

	$+/-$	$\times/\div$
FSSOR	$N^2 + N$	$N^2 + 2N$
HSSOR	$\frac{1}{4}N^2 + \frac{1}{2}N$	$\frac{1}{4}N^2 + N$

IV. NUMERICAL EXPERIMENTS AND RESULTS ANALYSIS

In this part, we carry out the numerical experiments over three examples to manifest the efficiency of the HSSOR methods, which derived by considering the 3HSLRFD-HSCT approximation equation for solving problem (1).

Example 1 [29] Consider the SOLFIDE

$$y''(t) = 32t + \int_{-1}^{1}(1 - tu)y(u)du, \quad (28)$$

$-1 \leq t \leq 1$, with two-point boundary values $y(-1) = -\frac{5}{2}, y(1) = \frac{15}{2}$, and the true solution of Equation (28) is $y(t) = 5t^3 + \frac{3}{2}t^2 + 1$.

Example 2 [30] Consider the SOLFIDE

$$y''(t) = 2 - \frac{16}{15}t - \frac{16}{15}t^2 + \int_{-1}^{1}(tu^2 - t^2u^2)y(u)du, \quad (29)$$

$-1 \leq t \leq 1$, with two-point boundary values $y(-1) = 1, y(1) = 3$, and the true solution of Equation (29) is $y(t) = t^2 + t + 1$.

Example 3 [30] Consider the SOLFIDE

$$y''(t) = e^t - t + \int_{0}^{1} tuy(u)du, \quad (30)$$

$-1 \leq t \leq 1$, with two-point boundary values $y(0) = 1, y(1) = e$, and the true solution of Equation (30) is $y(t) = e^t$.

In getting the numerical solution, the SOR iteration family such as FSSOR and HSSOR iterative methods is inspected to solve the linear system in which the controlling roles of the FSGS and FSSOR iterative methods are played. We carry out MATLAB software to do a lot of numerical experiments and compare the results in aspects of the quantity of iterations (*Iterations*), the elapsed time (*Time*) in seconds and the maximum values of absolute errors (*Error*) at five different numbers of subintervals $N = 32, 64, 128, 256, 512$. All the corresponding results are displayed in Table 4 to Table 6.

The results listed in Table 4 to Table 6 indicate that the HSSOR method based on the 3HSLRFD-HSCT scheme has the smallest amount *Iterations* and faster *Time* of the three strategies. Meanwhile, the results are depicted in Figs. 2-4. For the last parameter *Error*, it can be pointed out that the accuracy of all proposed techniques is in smart agreement. Moreover, we can also find the relationship between the FSSOR method and the HSSOR method. When the value of N of the latter method is two times that of the former method, the corresponding value of the three parameters is half of that of the former method. Finally, we calculate the percentage reduction of the first two parameters obtained by the FSSOR method and the HSSOR method compared with the values obtained by the FSGS method, which are as high as about 99%, as shown in Table 7. The incontestable results imply that HSSOR iterative method based on the 3HSLRFD-HSCT discretization scheme achieves the numerical solution with the highest efficiency.

V. CONCLUSIONS AND FUTURE WORK

This paper has successfully constructed the 3HSLRFD-HSCT discretization scheme, which is utilized to discretize the differential term and integral term of problem (1) to obtain the approximation equation and then generate the corresponding linear system (25). Next, we implement the HSSOR technique to find the numerical solution of system linear (25). The numerical experiments showed that the proposed method could achieve the desired accuracy fast with low Iterations and fast Time. The comparison with the other two approaches also elucidates that the HSSOR technique performs better in aspects of the *Iterations* and *Time*. This significant performance is not only thanks to the good approximation of the 3HSLRFD approach but also because the HSSOR and FSSOR iterative methods have the advantage of accelerating convergence. This conclusion is because both FSSOR and HSSOR iterative methods have involved one weighted parameter as an accelerating parameter. Apart from the SOR family, which is categorised as one of the point iteration families, this analysis can be continued to deal with the application of Alternating Group Explicit [14]-[17] and weighted mean [23]-[26] iteration families can be used as a linear solver.

TABLE 4. Comparison of results for three different techniques for Example 1.

Parameters	Methods	*N* = 32	64	128	256	512
Iterations	FSGS-3LRFD	185224	2492458	32429703	400325235	4513359199
	FSSOR-3LRFD (ω)	8216 (1.89470000)	58845 (1.94304900)	435012 (1.97389410)	3139706 (1.98725000)	21559438 (1.99293625)
	HSSOR-3LRFD (ω)	1141 (1.79251000)	8216 (1.89470000)	58845 (1.94304900)	435012 (1.97389410)	3139706 (1.98725000)
***Time* (seconds)**	FSGS-3LRFD	0.4624	7.6251	232.2066	6634.2979	190132.3872
	FSSOR-3LRFD	0.0262	0.2959	4.2856	59.6045	903.6976
	HSSOR-3LRFD	0.0047	0.0299	0.2737	4.1652	54.2396
Error	FSGS-3LRFD	1.2910E-03	3.2510E-04	1.0914E-04	4.5008E-04	4.9842E-03
	FSSOR-3LRFD	1.2908E-03	3.2270E-04	8.1130E-05	2.2816E-05	2.2708E-05
	HSSOR-3LRFD	5.1588E-03	1.2908E-03	3.2267E-04	8.1130E-05	2.2816E-05

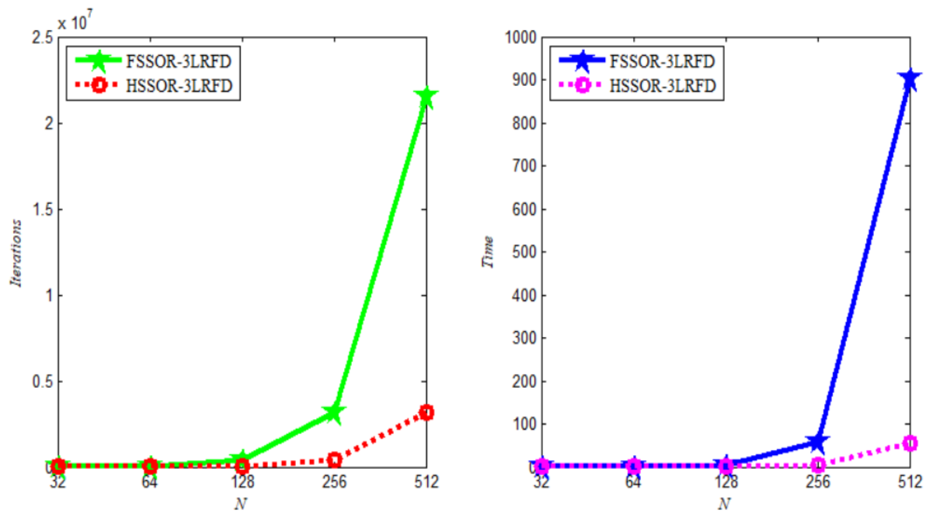

Fig. 2 *Iterations* and *Time* (seconds) versus *N* of two different techniques for Example 1.

TABLE 5. Comparison of results for three different techniques for Example 2.

Parameters	Methods	*N* = 32	64	128	256	512
Iterations	FSGS-3LRFD	448234	5959234	76098613	910442625	9658341997
	FSSOR-3LRFD (ω)	9670 (1.9443380000)	70503 (1.9723800000)	486033 (1.9853554600)	3470593 (1.9927109326)	26735660 (1.9967859460)
	HSSOR-3LRFD (ω)	1349 (1.8873600000)	9670 (1.9443380000)	70503 (1.9723800000)	486033 (1.9853554600)	3470593 (1.9927109326)
***Time* (seconds)**	FSGS-3LRFD	1.0256	27.3561	566.8762	14873.0139	414576.0779
	FSSOR-3LRFD	0.0327	0.8217	11.9369	58.8191	1364.6092
	HSSOR-3LRFD	0.0040	0.0294	0.3074	4.1724	58.2971
Error	FSGS-3LRFD	5.3204E-04	1.3904E-04	1.2114E-04	1.1261E-03	1.2686E-02
	FSSOR-3LRFD	5.3154E-04	1.3323E-04	3.2263E-05	5.6794E-06	4.7144E-02
	HSSOR-3LRFD	2.1151E-03	5.3154E-04	1.3323E-04	3.2263E-05	5.6794E-06

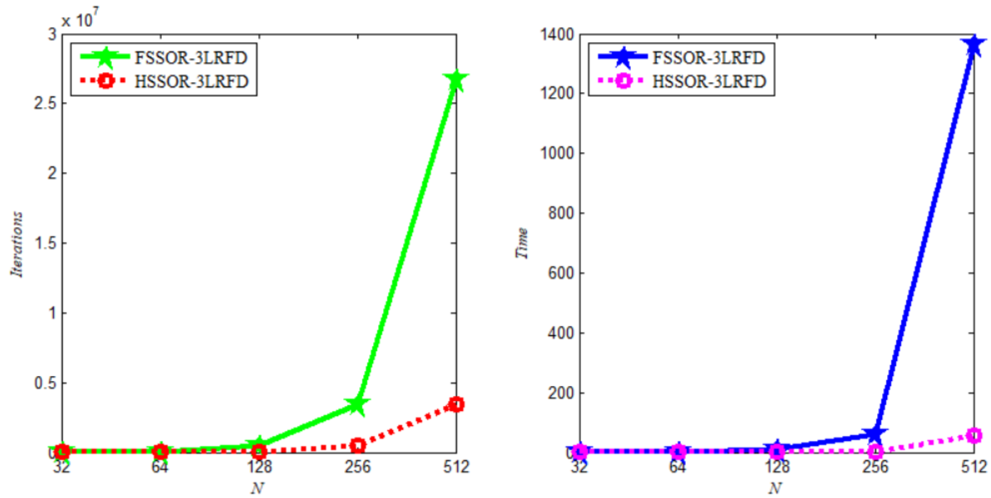

Fig. 3 *Iterations* and *Time* (seconds) versus *N* of two different techniques for Example 2.

TABLE 6. Comparison of results for three different techniques for Example 3.

Parameters	Methods	*N*				
		32	**64**	**128**	**256**	**512**
	FSGS-3LRFD	461118	6163295	79290805	960096418	10428247925
	FSSOR-3LRFD	9239	66863	482119	3685270	24259791
Iterations	(ω)	(1.94229380)	(1.97106190)	(1.98553030)	(1.99341000)	(1.99643221)
	HSSOR-3LRFD	1295	9239	66863	482119	3685270
	(ω)	(1.88417900)	(1.94229380)	(1.97106190)	(1.98553030)	(1.99341000)
***Time* (seconds)**	FSGS-3LRFD	1.0987	23.7031	622.6473	15178.6218	441013.7995
	FSSOR-3LRFD	0.0305	0.3315	3.6553	60.0946	992.9554
	HSSOR-3LRFD	0.0043	0.0348	0.3415	3.9125	61.6715
Error	FSGS-3LRFD	6.7632E-06	1.0129E-05	9.9643E-05	1.1235E-03	1.1207E-02
	FSSOR-3LRFD	6.0313E-06	1.2856E-06	1.2249E-06	8.4393E-06	5.3243E-05
	HSSOR-3LRFD	2.4195E-05	6.0313E-06	1.2856E-06	1.2249E-06	8.4394E-06

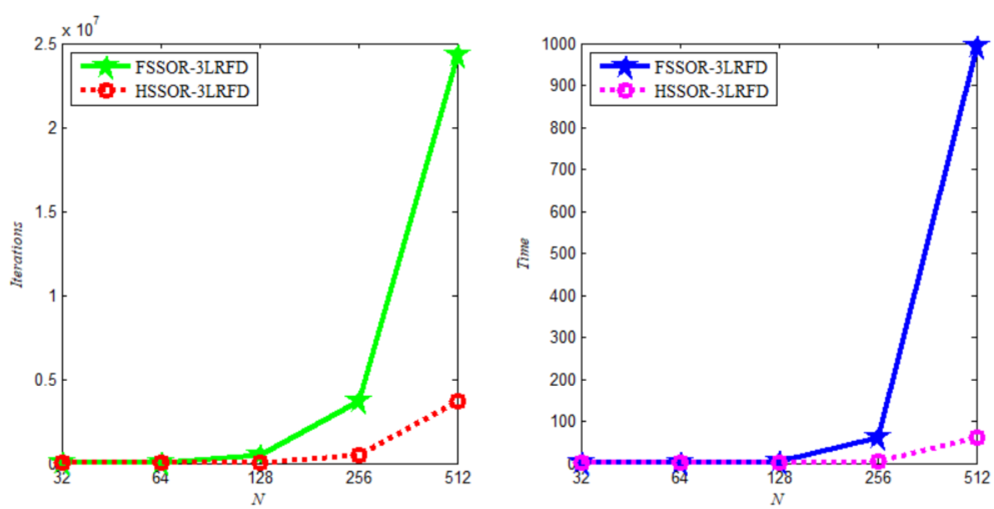

Fig. 4 *Iterations* and *Time* (seconds) versus *N* of two different techniques for Example 3.

TABLE 7. The percentage reductions in *Iterations* and *Time* of the HSSOR and FSSOR techniques in comparison to the FSGS technique.

Example	Methods	*Iterations*	*Time*
1	FSSOR-3LRFD	95.57%-99.52%	94.33%-99.52%
	HSSOR-3LRFD	99.38%-99.93%	98.98%-99.97%
2	FSSOR-3LRFD	97.84%-99.72%	96.81%-99.67%
	HSSOR-3LRFD	99.70%-99.96%	99.61%-99.99%
3	FSSOR-3LRFD	97.99%-99.77%	97.22%-99.77%
	HSSOR-3LRFD	99.73%-99.96%	99.61%-99.99%

ACKNOWLEDGMENT

The authors would really like to precise their heartfelt appreciation to the paper's anonymous reviewer as well as like to thank Universiti Malaysia Sabah for funding this analysis through the UMSGreat research grant for postgraduate students: GUG0513-2/2020.

REFERENCES

[1] V. Lakshmikantham and M. R. M. Rao, Theory of integro-differential equations, 1nd ed., USA : Gordon and Breach Science. (1995).

[2] S. M. Zemyan, The classical theory of integral equations. USA : Birkhauser. (2012).

[3] M. E. Durmaz, and G. M. Amiraliyev, A robust numerical method for a singularly perturbed fredholm integro-differential equation, Mediterranean Journal of Mathematics. 18(1) (2021) 1-17.

[4] N. Rohaninasab, K. Maleknejad and R. Ezzati, Numerical solution of high-order Volterra–Fredholm integro-differential equations by using Legendre collocation method, Applied Mathematics and Computation. 328 (2018) 171-188.

[5] E. Hesameddini, M. Shahbazi, Solving multipoint problems with linear Volterra–Fredholm integro-differential equations of the neutral type using Bernstein polynomials method, Applied Numerical Mathematics. 136 (2019) 122-138.

[6] E. D. Khiabani, B. Shiri, H. Ghaffarzadeh and D. Baleanu, Visco-elastic dampers in structural buildings and numerical solution with spline collocation methods, Journal of Applied Mathematics and Computing. 63(1) (2019) 1-29.

[7] N. Ahmad, A. Ullah, A. Ullah, S. Ahmad and K. Pakhtunkhwa, On analysis of the fuzzy fractional order volterra-fredholm integro-differential equation, Alexandria Engineering Journal. 60(1) (2021) 1827-1838.

[8] J. P. Berrut, M. S. Floater and G. Klein, Convergence rates of derivatives of a family of barycentric rational interpolants, Applied Numerical Mathematics. 61(9) (2011) 989–1000.

[9] G. Klein and J. P. Berrut, Linear rational finite differences from derivatives of barycentric rational interpolants, SIAM J. Numer. Anal. 50(2) (2012) 643-656.

[10] A. Abdi, S. A. Hosseini, The barycentric rational difference-quadrature scheme for systems of Volterra integro-differential equations, SIAM J. Sci. Comput.. 40(3) (2018)) A1936–A1960.

[11] A. Abdi, J. P. Berrut and S. A. Hosseini, The linear barycentric rational method for a class of delay Volterra integro-differential equations, J. Sci. Comput.. 75(3) (2018) 1757–1775.

[12] A. Abdi, S. A. Hosseini and H. Podhaisky, Adaptive linear barycentric rational finite differences method for stiff ODEs, Journal of Computational and Applied Mathematics. 357(0377-0427) (2019) 204–214.

[13] A. R. Abdullah, The Four Point Explicit Decoupled Group (EDG) Method: A Fast Poisson Solver, International Journal of Computer Mathematics. 38(1-2) (1991) 61–70.

[14] A. A. Dahalan, M. S. Muthuvalu and J. Sulaiman, Numerical Solutions of Two-Point Fuzzy Boundary Value Problem Using Half-Sweep Alternating Group Explicit Method, AIP Conference Proceedings, 1557 (2013) 103-107.

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

[16] A. A. Dahalan, M. S. Muthuvalu and J. Sulaiman, Numerical Solutions of Two-Point Fuzzy Boundary Value Problem Using Half-Sweep Alternating Group Explicit Method, AIP Conference Proceedings, 1557 (2013) 103-107.

[17] A. A. Dahalan and J. Sulaiman, Approximate solution for 2 dimensional Fuzzy Parabolic equations in QSAGE iterative method. International Journal of Mathematical Analysis, 9(35) (2015) 1733-1746.

[18] A. Saudi, J. Sulaiman, Path planning simulation using harmonic potential fields through four point-EDGSOR method via 9-point Laplacian, Jurnal Teknologi. 78(8-2) (2016) 12–24.

[19] A. Saudi, J. Sulaiman, Red-Black Strategy for Mobile Robot Path Planning, Lecture Notes in Engineering and Computer Science. 2182(1) (2010) 2215-2220.

[20] J. Sulaiman, M. K. Hasan, M. Othman and S. A. A. Karimd, MEGSOR iterative method for the triangle element solution of 2D Poisson equations, Procedia Computer Science. 1(1) (2010) 377-385.

[21] M. K. Hasan, M. Othman, Z. Abbas, J. Sulaiman, and R. Johari, The HSLO(3)-FDTD With Direct-Domain and Temporary-Domain Approaches On Infinite Space Wave Propagation, Proceedings of the 7th IEEE Malaysia International Conference on Communications and the 13th IEEE International Conference on Networks (MICC-ICON 2005), (2005) 1002-1007.

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

[23] M. S. Muthuvalu and J. Sulaiman, Quarter-Sweep Arithmetic Mean (QSAM) iterative method for second kind linear Fredholm integral equations, Appl. Math. Sci. 4 (2010) 2943-2953.

[24] M. S. Muthuvalu and J. Sulaiman, The Arithmetic Mean iterative methods for solving dense linear systems arise from first kind linear Fredholm integral equations, Proceedings of the Romanian Academy, Series A, of the Romanian Academy. (2012) 1-3.

[25] M. S. Muthuvalu and J. Sulaiman, The Quarter-Sweep Geometric Mean method for solving second kind linear Fredholm integral equations, Bull. Malays. Math. Sci. Soc., 36 (2013) 1009-1026.

[26] M. S. Muthuvalu and J. Sulaiman, Half-sweep geometric mean iterative method for the repeated Simpson solution of second kind linear Fredholm integral equations, Proyecciones (Antofagasta). 31(1) (2012) 65-79.

[27] L. H. Ali, J. Sulaiman, A. Saudi and M. M. Xu, Newton-SOR with Quadrature Scheme for Solving Nonlinear Fredholm Integral Equations, Lecture Notes in Electrical Engineering. 724 (2021) : 325-337.

[28] M. M. Xu, J. Sulaiman and L. H. Ali, Rational Finite Difference Solution of First-Order Fredholm Integro-differential Equations via SOR Iteration, Lecture Notes in Electrical Engineering. 724 (2021) 463-474.

[29] O. A. Gegele, O. P. Evans and D. Akoh, Numerical solution of higher order linear Fredholm Integro-differential equations, American Journal of Engineering Research. 3(8) (2014) 243-247.

[30] A. M. Wazwaz, A first course in integral equations, 2ed., USA : World Scientific. (2015).