International Journal of Mathematics and
Computer Science, **16**(2021), no. 4, 1555–1570

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

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

[1]School of Mathematics and Information Technology
Xingtai University
Xingtai - 054000, Hebei, China

[1,2]Faculty of Science and Natural Resources
Universiti Malaysia Sabah
Kota Kinabalu - 88400, Sabah, Malaysia

email: xmmzg@sina.com, jumat@ums.edu.my, labiyana15@gmail.com

(Received March 7, 2021, Revised April 21, 2021,
Accepted May 5, 2021)

Abstract

In order to highlight the advantages of linear rational finite difference (LRFD) and half-sweep iteration methods, we investigate the numerical solution of the first-order linear Fredholm integro-differential equation (FIDE) by implementing the Half-Sweep Successive Over-Relaxation (HSSOR) method based on the 3-point LRFD (3LRFD) and composite trapezoidal (CT) quadrature schemes. Meanwhile, for the sake of comparison, the classical or the standard Full-Sweep Gauss-Seidel (FSGS) and Full-Sweep Successive Over-Relaxation (FSSOR) methods are also presented as the control method. Finally, we introduce three numerical examples to verify that the HSSOR method proposed in this paper is superior to the existing FSGS and FSSOR methods, especially that the number of iterations and execution time are significantly reduced.

Key words and phrases: First-order Integro-differential equations, Half-sweep SOR iteration, Linear rational finite difference scheme, Half-sweep composite quadrature.
AMS (MOS) Subject Classifications: 45B05, 65R20.
ISSN 1814-0432, 2021, http://ijmcs.future-in-tech.net

1 Introduction

Mathematics, physics, chemistry, biology and many other disciplines involve mathematical equation models, which include differential equations, integral equations, integro-differential equations (IDEs), etc. [9, 15, 16, 17, 30]. In general, analytical solutions of equations are difficult to obtain. Therefore, many researchers focus on numerical solutions of IDEs problem, for instance, evolutionary computational intelligence [14], the priori Nystrom-method [8], and spline collocation methods [18]. In this paper, we concentrate on finding numerical solutions to the first-order Fredholm integro-differential equation (FIDE), which is defined as follows

$$y'(t) = p(t)y(t) + f(t) + \int_a^b K(t,u)y(u)du, \quad a \le t \le b, \tag{1.1}$$

$y(a) = y_a$, where the functions $f(t)$, $p(t)$ and the kernel $K(t,u)$ are known and $y(t)$ is the solution to be determined.

In 1988, Schneider and Werner [23] proposed the barycentric rational interpolants, which has the advantages of avoiding ill-conditioning and good stability compared with polynomial interpolants. Therefore, in recent years, more and more researchers [20, 29] are interested in it. Its significant application is that the linear rational finite difference (LRFD) method [19, 22, 28], which comes from by using the derivative of linear barycentric rational interpolants (LBRI) function to approximate the derivative of the interpolated function. In 2018, Abdi et al. introduced the LFRD method for a class of delay Volterra integro-differential equations [1] and systems of Volterra integro-differential equations [2]. In 2019, Abdi and Hosseini, in collaboration with other researchers, have further studied the LRFD for stiff ODEs [3] and stiff VIEs [4]. These applications of the LRFD method inspired us to apply it to solve the FIDE problem. In this paper, we use the 3-point LRFD (3LRFD) method combined with the composite trapezoidal (CT) quadrature formula respectively to discretize the differential part and integral part of the FIDE to generate linear systems. This combination of the linear rational finite difference discretization scheme and the composite trapezoidal quadrature scheme is referred to as the linear rational finite difference-quadrature discretization scheme.

Solving any linear system which is generated from the discretization process of any observed problem has attracted many studies to produce the fast numerical solution of the linear system. For instance, in 1991, Abdullah [10] first introduced the half-sweep iteration concept via the four-point explicit decoupled group iterative method for solving the two-dimensional Poisson equation. Clearly the author has shown that the main idea of half-sweep iteration is to take only half of the number of entire points in the solution domain of the observed problem. As a result, the half-sweep iteration concept can reduce the complexity of calculation in the iterative process, which naturally led to the reduction of the number of iterations and execution time. Due to its advantage in terms of the low computational complexity in implementing this concept, several applications of the half-sweep iteration concept have been extensively carried out to investigate its performance of finding the numerical solution for fuzzy boundary value problem [6, 7], robotic path planning [11, 12], and two-dimensional free space wave propagation problem [21, 25]. By taking these advantages into account, we extend the application of the half-sweep iteration concept with the Successive Over-Relaxation (SOR) method, namely HSSOR, to obtain the numerical solution of linear systems, which are generated by using the corresponding 3-point linear rational finite difference-quadrature approximate equation. In the next section, the formulations of the full- and half-sweep 3LRFD, namely 3FSLRFD and 3HSLRFD, CT and the SOR iteration family are elaborated. Some numerical experiments are demonstrated in Section 3 to validate the efficiency and accuracy of the constructed methods in this paper. In the end, the conclusion is given.

2 Methodology

As mentioned in the introduction, our main purpose is to find the numerical solution of the FIDE. In this study, the solution process can be mainly divided into two steps. The first step is to apply 3HSLRFD formula and half-sweep composite trapezoidal (HSCT) quadrature scheme respectively to discretize the differential and integral terms of the FIDE to get the approximate equation of the problem (1.1). The second step is to implement 3-point half-sweep SOR (3HSSOR) method to accelerate the solution process and find the numerical solution of the approximate equation for the problem (1.1). In this section, we will introduce 3HSLRFD, HSCT and HSSOR methods in detail. Before we introduce the rational finite difference-quadrature discretization scheme, the following discussion attempts to give the explanation of these described formulas.

2.1 Derivation of Half-Sweep Rational Finite Difference-Quadrature Approximate Equation

In order to find the numerical solution to problem (1.1), we firstly divide the solution domain, $[a, b]$ of problem (1.1) into N subintervals of equal step length $h = \frac{(b-a)}{N}$, $t_i = u_i = a + ih$, $i = 0, 1, \cdots, N$. In this study, the value of N is given by $N = 2^p, p \geq 1$. Based on the full-sweep iteration concept, the general solution process takes out all the partition points. However, the half-sweep iteration method is that we only take out the even partition points. The distribution of grid points is shown in Figure 1 a) and b) are standard or full-sweep iteration method and half-sweep iteration method respectively. Compared with h for each grid size of the full sweep iteration, each grid size of the half sweep iteration is $2h$, so it is reasonable that the latter has a fast convergence rate.

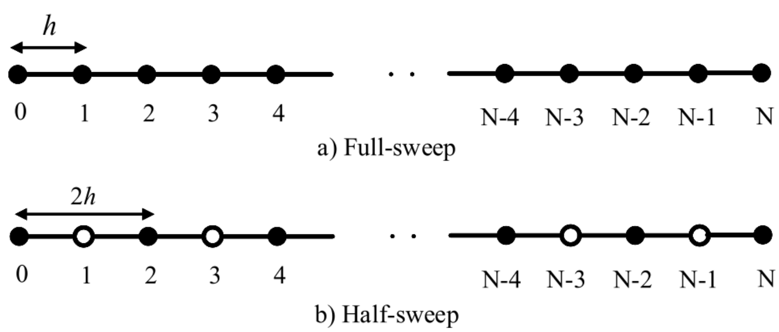

Figure 1: Distribution of the uniform mesh size for full-and half-sweep cases.

Referring to Figure 1 and considering the distribution of uniformly node points in the half-sweep case, we start to make the combination between the half-sweep iteration method alone with LFRD and CT schemes respectively to construct both new schemes, namely HSLRFD, and HSCT. Further, we attempt to apply both newly established formulas to quickly and accurately obtain the numerical solution of the problem (1.1). To do this matter, we need to discuss how to get the 3-point half-sweep linear rational finite difference-quadrature approximate equation via both newly established schemes.

2.1.1 3-Point Linear Half-Sweep Rational Finite Difference Method

In this subsection, we need to construct the 3HSLRFD formula. As mentioned in Section 1, LRFD method [19, 22] are based on LBRI. Compared with the half-sweep LBRI (HSLBRI) and HSLRFD formulas proposed in this paper, the previous LBRI and LRFD formulas are also called FSLBRI and FSLRFD respectively.

Let $t_0, t_1, \cdots, t_m$ be $(m+1)$ real abscissas and $y(t_0), y(t_1), \cdots, y(t_m)$ corresponding values. The FSLBRI [20, 23] to these data will here be an expression of the form

$$Y_{F_m}(t) = \sum_{j=0}^{m} \frac{\frac{\xi_{F_j}}{t-t_j} y(t_i)}{\sum\limits_{j=0}^{m} \frac{\xi_{F_j}}{t-t_j}}, \tag{2.2}$$

In 2007, Floater et al. [29] gave a specific expression of weights $\xi_j, j = 0, 1, \cdots, m$ $(0 \leq d \leq m)$. For equispaced nodes, the weights formulas are

$$\xi_{F_j} = \frac{(-1)^{j-d}}{2^d} \sum_{k \in J_{F_j}} \binom{d}{j-k}, J_{F_j} := (k \in 0, 1, \cdots, m-d : j-d \leq k \leq j). \tag{2.3}$$

In 2011, Berrut et al. [22] introduced the differentiation of FSLBRI, the formula of FSLRFD to approximate the first derivative of $y(t)$ on $t_0, t_1, \cdots, t_m$ is written as

$$y'(t_i) \approx Y'_{F_m}(t_i) = \frac{1}{h} \sum_{j=0}^{m} D_{F_{i,j}} y(t_j), \quad i = 0, 1, \cdots, m, \tag{2.4}$$

where

$$D_{F_{i,j}} = \begin{cases} \frac{\xi_{F_j}}{\xi_{F_i}} \left(\frac{1}{i-j}\right), & j \neq i, \\ -\sum\limits_{l=0; l \neq i}^{m} D_{F_{i,l}}, & j = i. \end{cases} \tag{2.5}$$

Based on the idea of the half-sweep iteration method introduced in section 2.1 and the Equations (2.2)-(2.5) in this section. The HSLBRI to $t_0, t_2, \cdots, t_m$ (m here and below is even) will be constructed as

$$Y_{H_m}(t) = \sum_{j=0,2}^{m} \frac{\frac{\xi_{H_j}}{t-t_j} y(t_i)}{\sum\limits_{j=0,2}^{m} \frac{\xi_{H_j}}{t-t_j}}, \tag{2.6}$$

where

$$\xi_{H_j} = \frac{(-1)^{\frac{j}{2}-d}}{2^d} \sum_{k \in J_{H_j}} \binom{d}{\frac{j}{2}-k}, J_{H_j} := (k \in 0, 1, \cdots, \frac{m}{2} - d : \frac{j}{2} - d \leq k \leq \frac{j}{2}). \tag{2.7}$$

So, the the formula of HSLRFD to approximate the first derivative of $y(t)$ on $t_0, t_2, \cdots, t_m$ is written as

$$y'(t_i) \approx Y'_{H_m}(t_i) = \frac{1}{h} \sum_{j=0,2}^{m} D_{H_{i,j}} y(t_j), \quad i = 0, 2, \cdots, m, \tag{2.8}$$

where

$$D_{H_{i,j}} = \begin{cases} \frac{\xi_{H_j}}{\xi_{H_i}} (\frac{1}{i-j}), & j \neq i, \\ -\sum\limits_{l=0,2;l \neq i}^{m} D_{H_{i,l}}, & j = i. \end{cases} \tag{2.9}$$

Considering the Equations (2.6)-(2.9), we quickly get the 3-point HSLRFD (3HSLRFD) formula, whose expression is

$$y'(t_i) = Y'(t_i) + e(t_i) \quad i = 2, 4, \cdots, N, \tag{2.10}$$

where $e(t_i)$ is the truncation error and

$$Y'_{t_i} = \begin{cases} \frac{1}{2h} \sum\limits_{j=i-2,i}^{i+2} D_{i,j} y(t_j), & i = 2, 4, \cdots, N-2, \\ \frac{1}{2h} \sum\limits_{j=i-4,i-2}^{i} D_{i,j} y(t_j), & i = N, \end{cases} \tag{2.11}$$

where

$$D_{i,j} = \begin{cases} \frac{\xi_{i,j}}{\xi_{i,i}} (\frac{1}{i-j}), & j \neq i, \\ -(D_{N,N-4} + D_{N,N-2}), & j = i = N. \\ -(D_{i,i-2} + D_{i,i+2}), & others. \end{cases} \tag{2.12}$$

In this study, we apply the 3HSLRFD formula to discretize the differential part of problem (1.1) in order to derive the corresponding rational finite difference-quadrature approximate equation for problem (1.1). we mainly focused on the 3HSLRFD at $d = 1$, and the corresponding value of $\xi_{i,j}$ and $D_{i,j}$ can be shown in Tables 1 and 2. Then the corresponding order of error accuracy can be obtained from Klein and Berrut [19] as $|e(t_i)| = O(h)$.

Table 1: The values of $\xi_{i,j}$

	$\xi_{i,i-2}/\xi_{N,N-4}$	$\xi_{i,i}/\xi_{N,N-2}$	$\xi_{i,i+2}/\xi_{N,N}$
$i = 2, 4, \cdots, N-2$	$-\frac{1}{2}$	1	$-\frac{1}{2}$

Table 2: The values of $D_{i.j}$

$i = 2, 4, \cdots, N-2$	$D_{i,i-2}$	$D_{i,i}$	$D_{i,i+2}$
	$-\frac{1}{4}$	0	$\frac{1}{4}$
$i = N$	$D_{i,i-4}$	$D_{i,i-2}$	$D_{i,i}$
	$\frac{1}{4}$	-1	$\frac{3}{4}$

2.1.2 Half-Sweep Quadrature Method

In this subsection, we attempt to present the HSCT scheme. The integral term in Equation (1.1) need to be discretized by applying the HSCT scheme from the family of quadrature methods to construct approximate equations coincide with the differential term. Generally, the quadrature formula can be expressed as

$$\int_a^b y(u)du = \sum_{j=0}^{N} C_{F_j} y(u_j) + \delta_n(y), \tag{2.13}$$

where C_{F_j} are independent numerical coefficients and $\delta_n(y)$ is the truncation error. To construct the formula of the approximate equations for problem (1.1), we consider the CT method. Thus, the C_{F_j} based on CT method can be shown as follows:

$$C_{F_j} = \begin{cases} \frac{1}{2}h, & j = 0, N, \\ h, & j = 1, 2, \cdots, N-1. \end{cases} \tag{2.14}$$

In our paper, we also call Equation (2.13) the FSCT. The HSCT formula is obtained by combining the half-sweep iteration method with the CT method as follows:

$$\int_a^b y(u)du = \sum_{j=0,2}^{N} C_j y(u_j) + \widetilde{\delta}_n(y), \tag{2.15}$$

where

$$C_j = \begin{cases} h, & j = 0, N, \\ 2h, & j = 2, 4, \cdots, N-2. \end{cases} \tag{2.16}$$

By substituting Equations (2.11), (2.12), (2.15) and (2.16) into Equation (1.1), a general form of the half-sweep rational finite difference-quadrature approximate equation can be constructed as

$$\begin{cases} \frac{1}{h}\sum_{j=i-2,i}^{i+2} D_{i,j}y_j = p_i y_i + f_i + \sum_{j=0,2}^{N} C_j K_{i,j} y_j, & i = 2,4,\cdots,N-2, \\ \frac{1}{h}\sum_{j=i-4,i-2}^{i} D_{i,j}y_j = p_i y_i + f_i + \sum_{j=0,2}^{N} C_j K_{i,j} y_j, & i = N, \end{cases} \tag{2.17}$$

where $y_i = y(t_i), p_i = p(t_i), f_i = f(t_i), K_{i,j} = K(t_i, u_j)$.

Based on the approximation in Equation (2.17), the corresponding linear systems can be constructed which can be easily shown as

$$\widetilde{M}\widetilde{y} = \widetilde{F}, \tag{2.18}$$

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

$$\widetilde{y} = \left[y_2, y_4, \cdots, y_{N-2}, y_N\right]^T,$$

$$F = \left[f_2 + \tfrac{1}{4h}y_0 + hK_{2,0}y_0, f_4 + hK_{4,0}y_0, \cdots, f_{N-2} + hK_{N-2,0}y_0, f_N + hK_{N,0}y_0\right]^T,$$

Clearly it can be observed that the main characteristic of the coefficient matric, $\widetilde{M}$ for the linear system (2.18) is large-scale and dense matric. In the next subsection, we proceed to the second step, namely finding the numerical solution of the linear system (2.18).

2.2 Formulation of Half-Sweep Successive Over-Relaxation Method

We have already known that the half-sweep iteration method can reduce the complexity of iteration and thus accelerate the convergence rate. Here, we combine the half-sweep iteration method with the SOR method [11, 21, 24, 26, 27, 28] to obtain the HSSOR method, and then apply the HSSOR method to get the numerical solution of the linear system (2.18).

To start the discussion of constructing the formulation of the HSSOR iterative method, let us decompose the coefficient matrix $\widetilde{M}$ as the summation of three matrices which is expressed as follows

$$\widetilde{M} = \widetilde{D} - \widetilde{L} - \widetilde{U}, \tag{2.19}$$

where $\widetilde{D}, \widetilde{L}$ and $\widetilde{U}$ are diagonal, strictly lower triangular and strictly upper triangular matrices respectively. Thus, the general formula for HSSOR iterative method can be written as

$$\widetilde{y}^{(k+1)} = (\widetilde{D} - \omega\widetilde{L})^{-1}((1-\omega)\widetilde{D} + \omega\widetilde{U})\widetilde{y}^{(k)} + \omega(\widetilde{D} - \omega\widetilde{L})^{-1}\widetilde{F}, \qquad (2.20)$$

where ω is a weighted parameter.

In getting the numerical solution, the SOR iteration family such as FSSOR and HSSOR, which is stated in the previous section is used to solve the system of linear equations iteratively with the approximate solution to the vector $\widetilde{y}$. The iteration process of these three iterative methods is continued until the solution is within a predetermined acceptable bound on the error. By determining the values of matrices $\widetilde{D}, \widetilde{L}$ and $\widetilde{U}$ as per stated in Equation (2.19), the general algorithm for HSSOR method to solve the linear system (2.18) can be described as in Algorithm 2.1. Here we apply MATLAB software, using the matrix method to achieve the Algorithm 2.1.

Algorithm 2.1. *HSSOR method*
a) Initializing all the parameters. Set $k = 0$, *and* $\widetilde{y}^{(0)} = [0, 0, \cdots, 0]^T$.
b) For $k = 1, 2, 3, \cdots$, *perform*
  i) Calculate

$$\widetilde{y}^{(k+1)} = (\widetilde{D} - \omega\widetilde{L})^{-1}(((1-\omega)\widetilde{D} + \omega\widetilde{U})\widetilde{y}^{(k)} + \omega\widetilde{F}).$$

  ii) Convergence test. If the error of tolerance $\widetilde{y}^{(k+1)} - \widetilde{y}^{(k)} \leq \sigma = 10^{-10}$ *is satisfied, then go step c).*
c) Display the numerical solution.

Now, we analyse the computational complexity of FSSOR method and HSSOR method from the perspective of arithmetic operation of each iteration. Based on algorithm 2.1, we can calculate that each iteration of the HSSOR method requires $\frac{N^2}{4} + \frac{N}{2}$ additions/subtractions and $\frac{N^2}{4} + N$ multiplications/divisions, while that of the FSSOR method requires $N^2 + N$ additions/subtractions and $N^2 + 2N$ multiplications/divisions. Compared with the FSSOR method, the HSSOR method greatly reduces the computational complexity.

3 Results and Discussion

In this section, we give three examples of first-order linear FIDE problem. Through these three examples, we can more convincingly verify the effectiveness of the methods we established in the second section for solving problem (1.1).

Example 3.1. *[30] Consider the linear FIDE of first order*

$$y'(t) = 1 - \frac{1}{3}t + \int_0^1 tuy(u)du, \quad 0 \leq t \leq 1, \tag{3.21}$$

with its initial condition $y(0) = 0$*, and exact solution is* $y(t) = t$*.*

Example 3.2. *[9][Section 4.4.2, Example 2] Consider the linear FIDE of first order*

$$y'(t) = \frac{1}{6} - \frac{1}{18}t + \int_0^1 tuy(u)du, \quad 0 \leq t \leq 1, \tag{3.22}$$

with its initial condition $y(0) = 0$*, and exact solution is* $y(t) = \frac{1}{6}t$*.*

Example 3.3. *[9][Section 4.4.2, Example 1] Consider the linear FIDE of first order*

$$y'(t) = \cos(t) + \frac{1}{4}t - \frac{1}{4}\int_0^{\frac{\pi}{2}} tuy(u)du, \quad 0 \leq t \leq 1, \tag{3.23}$$

with its initial condition $y(0) = 0$*, and exact solution is* $y(t) = \sin(t)$*.*

For these three examples, the numerical solutions are obtained by implementing the HSSOR, FSSOR and FSGS methods respectively, in which the controlling roles of the FSSOR method and FSGS method are played. At the same, we carry out MATLAB software to do a lot of numerical experiments, and compare the results from three aspects of the number of iterations (Iterations), the execution time (Time) in seconds and the maximum values of absolute errors (Error). The corresponding results are displayed in Tables 3 to 5.

From Table 3 to Table 5, it can be intuitively seen that for the values of the first two parameters, that is, the Iterations and Time. The values obtained by the HSSOR method are the lowest, while the values obtained by the opposite FSGS method are the highest, that is, the HSSOR method has the fastest convergence, followed by the FSSOR method, and FSGS method is the slowest. For the last parameter Error, it can be pointed out that the accuracy of all proposed iterative methods is in a good agreement. Finally, we show the percentage reduction of the first two parameters obtained by the FSSOR method and the HSSOR method compared with the values obtained

Table 3: Comparison of three parameters for three different iterative methods at Example 3.1.

Parameters	**Methods**	**N**				
		64	128	256	512	1024
Iterations	FSGS-3LRFD	41584	14046	495795	1789532	6515849
	FSSOR-3LRFD	547	1044	2001	3874	8041
	(ω)	(1.961017)	(1.979581)	(1.989566)	(1.994722)	(1.997494)
	HSSOR-3LRFD	320	547	1044	2001	3874
	(ω)	(1.928139)	(1.961017)	(1.979581)	(1.989566)	(1.994722)
Time	FSGS-3LRFD	0.2088	0.9906	6.9489	60.8101	1895.7701
	FSSOR-3LRFD	0.0028	0.0068	0.0291	0.1743	2.2537
	HSSOR-3LRFD	0.0012	0.0027	0.0100	0.0305	0.1492
Error	FSGS-3LRFD	2.3215E-05	5.7183E-06	1.2026E-06	7.3113E-07	2.0422E-06
	FSSOR-3LRFD	2.3249E-05	5.8112E-06	1.4513E-06	3.6129E-07	9.3526E-08
	HSSOR-3LRFD	9.3018E-05	2.3249E-05	5.8112E-06	1.4513E-06	3.6129E-07

Table 4: Comparison of three parameters for three different iterative methods at Example 3.2.

Parameters	**Methods**	**N**				
		64	128	256	512	1024
Iterations	FSGS-3LRFD	37141	124500	435527	1555552	5594038
	FSSOR-3LRFD	526	1013	1752	3760	7414
	(ω)	(1.961228)	(1.980999)	(1.989222)	(1.995124)	(1.997552)
	HSSOR-3LRFD	296	526	1013	1752	3760
	(ω)	(1.928726)	(1.961228)	(1.980999)	(1.989222)	(1.995124)
Time	FSGS-3LRFD	0.1964	1.0987	7.1231	59.4149	1430.5265
	FSSOR-3LRFD	0.0029	0.0101	0.0315	0.1626	2.0271
	HSSOR-3LRFD	0.0011	0.0024	0.0097	0.0309	0.1388
Error	FSGS-3LRFD	3.8377E-06	8.7422E-07	2.6501E-07	7.3113E-07	2.0421E-06
	FSSOR-3LRFD	3.8742E-06	9.7089E-07	2.4471E-07	6.3816E-08	1.8814E-08
	HSSOR-3LRFD	1.5502E-05	3.8742E-06	9.7089E-07	2.4471E-07	6.3816E-08

Table 5: Comparison of three parameters for three different iterative methods at Example 3.3.

P.	**Methods**	**N**				
		64	128	256	512	1024
I.	FSGS-3LRFD	25786	92868	340451	1255513	4629343
	FSSOR-3LRFD	463	845	1710	3205	6317
	(ω)	(1.949431)	(1.973541)	(1.987117)	(1.993322)	(1.996656)
	HSSOR-3LRFD	246	463	845	1710	3205
	(ω)	(1.902434)	(1.949431)	(1.973541)	(1.987117)	(1.993322)
T.	FSGS-3LRFD	0.1331	0.8219	5.8326	47.7569	2513.6701
	FSSOR-3LRFD	0.0025	0.0086	0.0291	0.1359	1.7727
	HSSOR-3LRFD	0.0009	0.0026	0.0064	0.0280	0.1319
E.	FSGS-3LRFD	7.0823E-05	1.7668E-05	4.3335E-06	8.5428E-07	1.7465E-06
	FSSOR-3LRFD	7.0842E-05	1.7706E-05	4.4253E-06	1.1081E-06	2.7851E-07
	HSSOR-3LRFD	2.8359E-04	7.0842E-05	1.7706E-05	4.4253E-06	1.1081E-06

by the FSGS method, which are as high as about 99%, as shown in Table 6. To sum up, considering these three methods comprehensively, the HSSOR method is the best choice.

Table 6: Percentage reductions of number of iterations and execution time of FSSOR and HSSOR method relative to FSGS method in solving Examples 3.1 to 3.3 by implementing 3LRFD-CT formulas.

Example	**Methods**	**Iterations**	**Time**
1	FSGS-3LRFD	98.68%-99.88%	98.66%-99.88%
	HSSOR-3LRFD	99.61%-99.94%	99.43%-99.95%
2	FSGS-3LRFD	98.58%-99.87%	99.43%-99.95%
	HSSOR-3LRFD	99.20%-99.93%	99.44%-99.99%
3	FSGS-3LRFD	98.20%-99.86%	98.12%-99.93%
	HSSOR-3LRFD	98.39%-99.93%	98.95%-99.99%

4 Conclusion

In this paper, we have succesfully constructed 3HSLRFD, HSCT and HSSOR methods by combining the half-sweep iteration method with LRFD, CT and SOR methods.Then, we apply these three methods to find the numerical solution of problem (1.1). In order to verify the effectiveness of the HSSOR method based on the 3HSLRFD-HSCT methods, we take the FSSOR method and FSGS method as control, and confirmed the value of the method proposed in this paper through three examples. This significant result is not only due to the good approximation of the 3HSLFRD method, but also because the half-sweep iteration has the advantage of accelerating convergence. Due to the advantage of the SOR iteration family, which is categorised as the one weighted parameter iteration family, this study should be extended to deal with the application of the two parameter iteration family mainly on MSOR [26, 27] and AOR [5, 13] iteration families.

Acknowledgment. Authors would like to express sincere gratitude to Universiti Malaysia Sabah for funding this research under UMSGreat research grant for postgraduate student: GUG0513-2/2020.

References

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

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

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

[4] A. Abdi, S. A. Hosseini, H. Podhaisky, Numerical methods based on the Floater-Hormann interpolants for stiff VIEs, Numerical Algorithms, **85**, (2020), 867–886.

[5] A. A. Dahalan, A. A. Saudi, J. Sulaiman, Autonomous Navigation on Modified AOR Iterative Method in Static Indoor Environment, Journal of Physics: Conference Series, **1366**, no. 1, (2019), 012020.

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

[7] A. A. Dahalan, M. S. Muthuvalu, 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.

[8] A. I. Fairbairn, M. K. Kelmanson, A priori Nystrom-method error bounds in approximate solutions of 1-D Fredholm integro-differential equations, International Journal of Mechanical Sciences, **150**, (2019), 755–766.

[9] A. M. Wazwaz, A first course in integral equations, World Scientific, USA, 2015.

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

[11] 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.

[12] A. Saudi, J. Sulaiman, Red-black strategy for mobile robot path planning, Lecture Notes in Engineering and Computer Science, **2182**, no. 1, (2010), 2215–2220.

[13] A. Sunarto, J. Sulaiman, A. Saudi, Implicit finite difference solution for time-fractional diffusion equations using AOR method, Journal of Physics: Conference Series, **495**, (2014), 012032.

[14] B. S. H. Kashkaria, M. I. Syam, Evolutionary computational intelligence in solving a class of nonlinear Volterra-Fredholm integro-differential equations, Journal of Computational and Applied Mathematics, **311**, (2017), 314–323.

[15] C. Tunç, A remark on the qualitative conditions of nonlinear IDEs, Int. J. Math. Comput. Sci., **15**, no. 3, (2020), 905–922.

[16] C. Tunç, O. Tunç, New results on the stability, integrability and boundedness in Volterra integro- differential equations, Bull. Comput. Appl. Math., **6**, no. 1, (2018), 41–58.

[17] C. Tunç, O. Tunç, New qualitative criteria for solutions of Volterra integro-differential equations, Arab Journal of Basic and Applied Sciences, **25**, no. 3, (2018), 158–165.

[18] E. Dadkhah, B. Shiri, H. Ghaffarzadeh, D. Baleanu, Visco-elastic dampers in structural buildings and numerical solution with spline collocation methods, Journal of Applied Mathematics and Computing, **63**, (2020), 29–57.

[19] G. Klein, J. P. Berrut, Linear rational finite differences from derivatives of barycentric rational interpolants, SIAM J. Numer. Anal., **50**, no. 2, (2012), 643-656.

[20] J. P. Berrut, L. N. Trefethen, Barycentric Lagrange Interpolation, Society for Industrial and Applied Mathematics, **46**, no. 3, (2004), 501–517.

[21] J. Sulaiman, M. K. Hasan, M. Othman, S. A. A. Karim, MEGSOR iterative method for the triangle element solution of 2D Poisson equations, International Conference on Computational Science, ICCS 2010, Published by Elsevier Ltd., (2012), 377–385.

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

[23] J. P. Berrut, Rational functions for guaranteed and experimentally well-conditioned global interpolation, Comput. Math. Appl., **15**, no. 1, (1988), 1–16.

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

[25] M. K. Hasan, M. Othman, Z. Abbas, J. Sulaiman, R. Johari, The HSLO(3)-FDTD with direct-domain and temporary-domain approaches on infinite space wave propagation, In: Proceedings of the 7^{th} IEEE Malaysia International Conference on Communications and the 13^{th} IEEE International Conference on Networks, (2005), 1002–1007.

[26] M. K. M. Akhir, M. Othman, J. Sulaiman, Z. A. Majid, M. Suleiman, Numerical solution of Helmholtz equation using a new four point EGM-SOR iterative method, Applied Mathematical Sciences, **5**, no. 80, (2011), 3991–4004.

[27] M. K. M. Akhir, M. Othman, J. Sulaiman, Z. A. Majid, M. Suleiman, The four point-EDGMSOR iterative method for solution of 2D Helmholtz equations, Informatics Engineering and Information Science, (2001), 218–227.

[28] M. M. Xu, J. Sulaiman, 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] M. S. Floater, K. Hormann, Barycentric rational interpolation with no poles and high rates of approximation, Numer. Math., **107**, (2007), 315–331.

[30] P. Darania, A. Ebadian, A method for numerical solution of integro-differential equations, Applied Mathematics and Computation, **188**, (2007), 657–668.