Journal of Engineering Science and Technology
Vol. 15, No. 5 (2020) 3303 - 3313

NUMERICAL SOLUTION OF FUZZY FREDHOLM INTEGRAL EQUATIONS OF SECOND KIND USING HALF-SWEEP GAUSS-SEIDEL ITERATION

L. H. ALI*, J. SULAIMAN, S. R. M. HASHIM

Mathematics with Economics Programme, Faculty of Science and Natural Resources, Universiti Malaysia Sabah, 88400, Kota Kinabalu, Sabah, Malaysia
*Corresponding Author: labiyana15@gmail.com

Abstract

The purpose of this study is to apply the Half-Sweep Gauss-Seidel (HSGS) iteration to find the approximate solution of fuzzy Fredholm integral equations of the second kind (FFIE-2). The approximation equation of FFIE-2 is derived by using trapezoidal quadrature scheme to generate Half-Sweep system of fuzzy linear equations. Then, the generated linear system of FFIE-2 is solved using HSGS. After that, some numerical examples are conducted to test the efficiency of the HSGS compared with Full-Sweep Jacobi (FSJ) and Full-Sweep Gauss-Seidel (FSGS) iterative methods. The comparative analysis is done by using three measuring parameters: number of iterations, iteration time, and Hausdorff distance. Based on the findings, it can be pointed out that HSGS method is more efficient than the FSJ and FSGS iterative methods.

Keywords: Fuzzy integral equations, Gauss-Seidel iteration, Half-sweep Gauss-Seidel, Numerical methods.

1. Introduction

The topic of fuzzy integral equations has been developed in the research area due to its uniqueness in its applications. Fuzzy integral equations have a wide range of applications in engineering, applied mathematics, and medical sciences [1, 2]. Therefore, the researchers all over the world are interested to study and propose a variety of methods to solve the problem of fuzzy integral equations. Some researchers have proposed analytical methods to solve fuzzy Fredholm integral equations such as in [3-6]. However, in some cases, analytical methods cannot be used or has a difficult solution in solving these equations [7]. Thus, many researchers have solved fuzzy Fredholm integral equations using numerical methods such as in [8-11]. Apart from fuzzy Fredholm integral equations, many studies have been conducted in solving fuzzy Volterra integral equations [12, 13], fuzzy Fredholm-Volterra integral equations [7], and fuzzy Volterra-Fredholm integral equations [14-16]. However, since some methods are unable to be applied in all problems; therefore, it is important to propose the other efficient method to solve the problem of fuzzy integral equations.

In this paper, we aim to solve the problem of fuzzy Fredholm integral equations of the second kind (FFIE-2) using a numerical method. In our previous studies, we have proposed iterative methods to solve FFIE-2 based on trapezoidal and Simpson's 1/3 rule [17, 18]. However, the computational complexity of these approaches is still high because it is categorized as a Full-Sweep case. In 1999, the computational complexity technique which known as Half-Sweep approach has been introduced by Abdullah to speed up the computational time in solving the linear problem [19]. Then, this approach has been applied in many numerical research papers including in [20-22] and they found Half-Sweep approach can reduce the iteration time compared with Full-Sweep approach. In this study, we proposed Half-Sweep Gauss-Seidel (HSGS) iteration based on trapezoidal quadrature scheme to solve linear FFIE-2 with r-cuts. In past studies, HSGS has been applied in robot part planning [23], image processing [24], and hydrodynamic lubrication problems [25].

Before we construct the approximation equations of Half-Sweep FFIE-2, let us consider the following general definition of Fredholm integral equations of the second kind as follows [10]

$$u(x) = f(x) + \lambda \int_a^b k(x,t)u(t)dt, \tag{1}$$

where $\lambda > 0, k(x,t)$ is an arbitrary kernel function over the square $a \le x,t \le b$ and f(x) is a continuous function of x on interval $[a,b]$. Equation (1) will obtain crisp solution if $f(x)$ is a crisp function and the solution of Eq. (1) is a fuzzy solution if $f(x)$ is a fuzzy function. Based on Eq. (1), we can get the general form of FFIE-2 which given as follows

$$\tilde{u}(x,r) = \tilde{f}(x,r) + \lambda \int_a^b k(x,t)\tilde{u}(t,r)dt, \tag{2}$$

where $\tilde{u}(x,r) = \left(\underline{u}(x,r), \overline{u}(x,r)\right)$ and $\tilde{f}(x,r) = \left(\underline{f}(x,r), \overline{f}(x,r)\right)$.

The parametric form of FFIE-2 now can be represented as follows [1]

$$\underline{u}(x,r) = \underline{f}(x,r) + \lambda \int_a^b \underline{U}(t,r)dt, \tag{3}$$

$$\overline{u}(x,r) = \overline{f}(x,r) + \lambda \int_a^b \overline{U}(t,r)dt, \tag{4}$$

where

$$\underline{U}(t,r) = \begin{cases} k(x,t)\underline{u}(t,r), & k(x,t) \geq 0 \\ k(x,t)\overline{u}(t,r), & k(x,t) < 0 \end{cases} \quad (5)$$

$$\overline{U}(t,r) = \begin{cases} k(x,t)\overline{u}(t,r), & k(x,t) \geq 0 \\ k(x,t)\underline{u}(t,r), & k(x,t) < 0 \end{cases} \quad (6)$$

for each $0 \leq r \leq 1$ and $t \in [a,b]$.

This paper consists of five sections. In section 2, we discuss the Half-Sweep quadrature discretization scheme to derive the Half-Sweep on trapezoidal approximation equation and then we generate the corresponding linear system of FFIE-2 using the Half-Sweep trapezoidal approximation equation. In section 3, we discuss the formulation and algorithm of HSGS iterative method. In section 4, we present some numerical examples for the case of linear FFIE-2. Finally, we present the numerical results and discuss the performances of the Full-Sweep Jacobi (FSJ) and the family of GS iterative methods which are Full-Sweep Gauss-Seidel (FSGS) and Half-Sweep Gauss-Seidel (HSGS) in Section 5.

2. Half-Sweep Trapezoidal Approximation Equation

Before constructing the discretization process over the problem of FFIE-2 in Eq. (2), let us consider the finite grid networks for the case of Full-Sweep and Half-Sweep on interval $[a,b]$. The node points of Full-Sweep and Half-Sweep cases can be uniformly distributed on the interval $[a,b]$ as shown in Fig. 1 [17, 18]. This means that interval $[a,b]$ can be divided into n subintervals $\{x_0, x_1, x_2, \ldots, x_n\}$ as $a = x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n = b$.

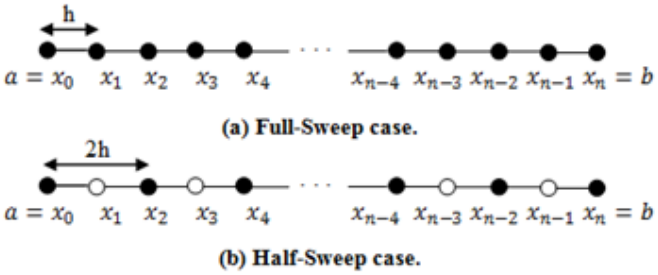

Fig. 1. Finite grid networks and the uniform distribution of node points on interval $[a, b]$.

As illustrated on Fig. 1, the Full-Sweep case consists of node points type ●, meanwhile, the Half-Sweep case consists of node points type ● and ○. When computing the approximate values of n subintervals, the Full-Sweep iteration will consider all node points type ●. Using Half-Sweep iteration, the approximate values will only be computed on node points of type ●, while the computation of the other remaining node points of type ○ will be computed after the convergence criterion is satisfied. Theoretically, the application of this Half-Sweep approach can reduce the iteration time by nearly half if compared with Full-Sweep approach [19].

Referring to Fig. 1, we propose the first-order Half-Sweep quadrature discretization scheme for the derivation of Half-Sweep trapezoidal approximation equation to form a linear system of FFIE-2. To do this, we need to consider the general quadrature scheme based on closed Newton-Cotes scheme over the fuzzy functions, $\tilde{g}(t,r) = \left(\underline{g}(t,r), \overline{g}(t,r)\right)$ as follows

$$\left(\int_a^b \underline{g}(t,r)dt\right) = \sum_{j=0,2,4,\ldots,n}^{n} A_j \underline{g}(t_j,r) + \underline{\varepsilon_n}(t), \quad (7)$$

$$\left(\int_a^b \overline{g}(t,r)dt\right) = \sum_{j=0,2,4,\ldots,n}^{n} A_j \overline{g}(t_j,r) + \overline{\varepsilon_n}(t), \quad (8)$$

where $A_j (j = 0,2,4,\ldots,n)$ is an independent numerical coefficient to the functions $\underline{g}(t,r)$ and $\overline{g}(t,r)$, t_j is the abscissa of the partition points of integrations on the interval $[a,b]$, $\tilde{\varepsilon}_n(t)$ is the truncation error, and n is the number of subintervals [20]. Using Eqs. (7) and (8), we represent those equations into a general form as follows

$$\int_a^b \tilde{g}(t,r)dt = \sum_{j=0,2,4,\ldots,n}^{n} A_j \tilde{g}(t_j,r) + \tilde{\varepsilon}_n(t), \quad (9)$$

Next, we define the general form of Half-Sweep trapezoidal integral equation based on closed Newton-Cotes on interval $[x_i, x_{i+2}]$ being stated as

$$\int_{x_i}^{x_{i+2}} g(t,r)dt = h\,(g_{i,r} + g_{i+2,r}), \quad (10)$$

where the value of constant step (h) is given as

$$h = \frac{b-a}{n}. \quad (11)$$

Then, by considering Eq. (10), the definition of A_j for the Half-Sweep case can be defined as [20]

$$A_j = \begin{cases} h & j = 0,n \\ 2h & otherwise \end{cases} \quad (12)$$

For illustration, let us consider $n = 5$ in Fig. 1. Then, we apply the Eq. (9) into Eq. (2) to get the following approximation equation of Half-Sweep FFIE-2 based on trapezoidal scheme [17, 18]

$$\tilde{u}(x,r) = \tilde{f}(x,r) + \int_{t_0}^{t_2} k(x,t)\tilde{u}(t,r)dt + \int_{t_2}^{t_4} k(x,t)\tilde{u}(t,r)dt + \int_{t_4}^{t_6} k(x,t)\tilde{u}(t,r)dt. \quad (13)$$

Eq. (13) also can be expended by using trapezoidal rule in Eq. (10) as follows

$$\begin{aligned}\tilde{u}(x,r) = \tilde{f}(x,r) &+ h[k(x,t_0)\tilde{u}(t_0,r) + k(x,t_2)\tilde{u}(t_2,r)] \\ &+ h[k(x,t_2)\tilde{u}(t_2,r) + k(x,t_4)\tilde{u}(t_4,r)] \\ &+ h[k(x,t_4)\tilde{u}(t_4,r) + k(x,t_6)\tilde{u}(t_6,r)]. \end{aligned} \quad (14)$$

Then,

$$\begin{aligned}\tilde{u}(x,r) = \tilde{f}(x,r) &+ h[k(x,t_0)\tilde{u}(t_0,r) + 2k(x,t_2)\tilde{u}(t_2,r) + 2k(x,t_4)\tilde{u}(t_4,r) + \\ &+ k(x,t_6)\tilde{u}(t_6,r)]. \end{aligned} \quad (15)$$

Consider $\tilde{u}(x_i,r) = \tilde{u}_{i,r}$; $\tilde{f}(x_i,r) = \tilde{f}_{i,r}$; $k(x_i,t_j) = k_{i,j}$; $\tilde{u}(t_i,r) = \tilde{u}_{i,r}$ we simplify Eq. (15) as

$$\tilde{u}_{i,r} = \tilde{f}_{i,r} + hk_{i,0}\tilde{u}_{0,r} + 2hk_{i,2}\tilde{u}_{2,r} + 2hk_{i,4}\tilde{u}_{4,r} + hk_{i,6}\tilde{u}_{6,r}. \quad (16)$$

Further, from Eqs. (12) and (16) we consider four node points of type ● on Fig. 1 (b) to have the following approximation equations of each node points for $i = 0,2,4,6$.

$$\left(1 - A_0(k_{0,0})\right)\tilde{u}_{0,r} - A_2(k_{0,2})\tilde{u}_{2,r} - A_4(k_{0,4})\tilde{u}_{4,r} - A_6(k_{0,6})\tilde{u}_{6,r} = \tilde{f}_{0,r} \quad (17)$$

$$-A_0(k_{2,0})\tilde{u}_{0,r} + \left(1 - A_2(k_{2,2})\right)\tilde{u}_{2,r} - A_4(k_{2,4})\tilde{u}_{4,r} - A_6(k_{2,6})\tilde{u}_{6,r} = \tilde{f}_{2,r} \quad (18)$$

$$-A_0(k_{4,0})\tilde{u}_{0,r} - A_2(k_{4,2})\tilde{u}_{2,r} + \left(1 - A_4(k_{4,4})\right)\tilde{u}_{4,r} - A_6(k_{4,6})\tilde{u}_{6,r} = \tilde{f}_{4,r} \quad (19)$$

$$-A_0(k_{6,0})\tilde{u}_{0,r} - A_2(k_{6,2})\tilde{u}_{2,r} - A_4(k_{6,4})\tilde{u}_{4,r} - \left(1 - A_6(k_{6,6})\right)\tilde{u}_{6,r} = \tilde{f}_{6,r} \quad (20)$$

Based on Eqs. (17) to (20), we can form the general matrix of all approximation equations based on the uniform node points of type ● for Half-Sweep case as follows

$$A\underline{\tilde{u}} = \underline{\tilde{f}}, \tag{21}$$

where

$$A = \begin{bmatrix} 1 - A_0(k_{0,0}) & -A_2(k_{0,2}) & -A_4(k_{0,4}) & \cdots & -A_n(k_{0,n}) \\ -A_0(k_{2,0}) & 1 - A_2(k_{2,2}) & -A_4(k_{2,4}) & \cdots & -A_n(k_{2,n}) \\ -A_0(k_{4,0}) & -A_2(k_{4,2}) & 1 - A_4(k_{4,4}) & \cdots & -A_n(k_{4,n}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -A_0(k_{n,0}) & -A_2(k_{n,2}) & -A_4(k_{n,4}) & \cdots & 1 - A_n(k_{n,n}) \end{bmatrix},$$

$$\underline{\tilde{u}} = [\tilde{u}_{0,r} \quad \tilde{u}_{2,r} \quad \tilde{u}_{4,r} \quad \cdots \quad \tilde{u}_{n,r}]^T,$$

$$\underline{\tilde{f}} = \left[\tilde{f}_{0,r} \quad \tilde{f}_{2,r} \quad \tilde{f}_{4,r} \quad \cdots \quad \tilde{f}_{n,r}\right]^T.$$

3. Formulation of the Gauss-Seidel Iteration Family

In this section, we discuss the formulation of GS iteration family to solve the linear system FFIE-2 in Eq. (21). First, we need to decompose the coefficient matrix A to construct the formulation of GS iteration family as follows [26]

$$A = L + D + U, \tag{22}$$

where

L = strictly lower triangular matrix,

D = diagonal matrix and

U = strictly upper triangular matrix.

The general formulation of GS iteration family to solve the linear system in Eq. (21) is defined as follows [27]

$$\underline{\tilde{u}}^{(k+1)} = (D - L)^{-1}U\underline{\tilde{u}}^{(k)} + (D - L)^{-1}. \tag{23}$$

Further, we used the formulation in Eq. (23) to write the algorithm of HSGS iterative method to solve linear system of Eq. (21) as in Algorithm 1

Algorithm 1 [17]:

i. Set the initial value $\underline{\tilde{u}}^{(0)} = 0$ and $\varepsilon = 10^{-10}$.

ii. Calculate matrix A and vector $\underline{\tilde{f}}$.

iii. For $i = 0,2,4, \ldots, n-2, n$ and $j = 0,2,4, \ldots, n-2, n$ calculate

$$\tilde{u}_i^{(k+1)} \leftarrow \begin{cases} \frac{1}{A_i K_{i,i}}\left(\tilde{f}_i - \sum_{j=2}^{n} A_j K_{i,j} \tilde{u}_j^{(k)}\right) & i = 0 \\ \frac{1}{A_i K_{i,i}}\left(\tilde{f}_i - \sum_{j=0}^{n-2} A_j K_{i,j} \tilde{u}_j^{(k+1)}\right), & i = n \\ \frac{1}{A_i K_{i,i}}\left(\tilde{f}_i - \sum_{j=0}^{i-2} A_j K_{i,j} \tilde{u}_j^{(k+1)} - \sum_{j=i+2}^{n} A_j K_{i,j} \tilde{u}_j^{(k)}\right), & otherwise \end{cases}$$

iv. Convergence test $\left|u_i^{(k+1)} - u_i^{(k)}\right| \leq \varepsilon = 10^{-10}$. Proceed to step v if the convergence criterion is satisfied. Otherwise, repeat from step iii.

v. Calculate the approximate solutions for node points of type ○ using the direct method.

vi. Display approximate solutions.

Based on Algorithm 1, we can observe the total computing operation of additions/subtractions and multiplications/divisions involved in HSGS as shown in Table 1. In contexts of computing complexity, it can be observed that the HSGS can reduce the total computing operations by half compared with FSJ and FSGS.

Table 1. Total computing operations involved per iteration.

Methods	Arithmetic Operation	
	additions/subtractions	**multiplications/divisions**
FSJ	$2(n+1)(n+1)$	$2(n+2)(n+1)$
FSGS	$2(n+1)(n+1)$	$2(n+2)(n+1)$
HSGS	$2\left(\frac{n}{2}+1\right)\left(\frac{n}{2}+1\right)$	$2\left(\frac{n}{2}+2\right)\left(\frac{n}{2}+1\right)$

4. Numerical Results

In this section, we consider three numerical examples of linear FFIE-2 to illustrate the applicability of HSGS compared with FSJ and FSGS. As a comparison, we set FSJ as a control method to compare the efficiency between FSGS and HSGS over FFIE-2. Secondly, we defined the measurement of parameters such as the number of iterations (K), iteration time (Time) in seconds, and Hausdorff distance (HD) as the parameters for comparative analysis. Next, we carried out the simulations on five selected grid sizes (G) i.e. 256, 512, 1024, 2048, and 4096 on the different values of r =1.0, 0.6, and 0.3 over all examples. The numerical examples of FFIE-2 in this study are presented as follows:

Example 1 [8]

Let us consider the following problem of linear FFIE-2

$$\underline{f}(x,r) = -\frac{1}{52}r(5-52x+2x^2), \tag{24}$$

$$\overline{f}(x,r) = \frac{1}{52}(r-2)(5-52x+2x^2), \tag{25}$$

and kernel

$$k(x,t) = \frac{x^2+t^2+2}{13}, 0 \le x,t \le 1 \text{ and } \lambda = 1, \tag{26}$$

and a = 0, b = 1. Given the exact solution of FFIE-2 as follows

$$\underline{u}(x,r) = rx, \tag{27}$$

$$\overline{u}(x,r) = (2-r)x. \tag{28}$$

Example 2 [8]

Let us consider the following problem of linear FFIE-2

$$\underline{f}(x,r) = \frac{2}{3}(2+r)x, \tag{29}$$

$$\overline{f}(x,r) = -\frac{2}{3}(r-4)x, \tag{30}$$

and kernel

$$k(x,t) = xt, 0 \le x,t \le 1, \lambda = 1, \quad (31)$$

and a = 0, b = 1. Given the exact solution of FFIE-2 as follows

$$\underline{u}(x,r) = (2+r)x, \quad (32)$$

$$\overline{u}(x,r) = (4-r)x. \quad (33)$$

Example 3 [11]

Let us consider the following problem of linear FFIE-2

$$\underline{f}(x,r) = rx(r^4+2)(3cos(1-x) + 5sin(1-x) - 6cos(x) + x^2), \quad (34)$$

$$\overline{f}(x,r) = -3x(r^3-2)(3cos(1-x) + 5sin(1-x) - 6cos(x) + x^2) \quad (35)$$

and kernel

$$k(x,t) = xcos(t-x), 0 \le x,t \le 1, \lambda = 1, \quad (36)$$

and a = 0, b = 1. Given the exact solution of FFIE-2 as follows

$$\underline{u}(x,r) = x^3(r^5+2r), \quad (37)$$

$$\overline{u}(x,r) = x^3(6-3r^3). \quad (38)$$

The numerical results from the implementation of FSJ, FSGS, and HSGS iterative methods for these three examples have been tabulated in Table 2, 3, and 4. In Table 5, we illustrate the performance analysis of these three iterative methods. The numerical results obtained show a reduction in the number of iteration and execution time using FSGS and HSGS compared with FSJ. Both FSGS and HSGS have reduced the number of iterations approximately 29.03-32.26% (Example 1), 39.13% (Example 2), and 40.63-42.19% (Example 3). However, in terms of execution time, HSGS recorded faster iteration time compared with FSGS as shown in Table 5. This happens because of the reduction of total computing operations in HSGS as discussed in the previous sections. The reduction percentages of HSGS recorded as 71.43-82.83% (Example 1), 75.00-84.70% (Example 2) and 84.33-89.39% (Example 3) which are relatively faster than FSJ. Thus, we conclude that HSGS is more efficient than FSGS and HSGS in solving FFIE-2 in terms of the number of iterations and iteration time due to the complexity reduction technique.

Table 2. Comparison between FSJ and GS iteration family for Example 1.

G	Methods	*r* = 1.0			*r* = 0.6			*r* = 0.3		
		K	**Time**	**HD**	**K**	**Time**	**HD**	**K**	**Time**	**HD**
256	FSJ	32	0.07	3.88E-07	31	0.07	5.44E-07	31	0.07	6.60E-07
	FSGS	22	0.06	3.88E-07	22	0.06	5.44E-07	21	0.06	6.60E-07
	HSGS	22	0.02	1.55E-06	22	0.02	2.17E-06	21	0.02	2.64E-06
512	FSJ	32	0.27	9.71E-08	31	0.26	1.36E-07	31	0.25	1.65E-07
	FSGS	22	0.19	9.71E-08	22	0.18	1.36E-07	21	0.19	1.65E-07
	HSGS	22	0.05	3.88E-07	22	0.05	5.44E-07	21	0.05	6.60E-07
1024	FSJ	32	1.00	2.43E-08	31	0.97	3.40E-08	31	0.96	4.12E-08
	FSGS	22	0.70	2.43E-08	22	0.69	3.40E-08	21	0.67	4.13E-08
	HSGS	22	0.18	9.71E-08	22	0.17	1.36E-07	21	0.17	1.65E-07
2048	FSJ	32	3.96	6.06E-09	31	3.80	8.48E-09	31	3.79	1.03E-08
	FSGS	22	2.74	6.07E-09	22	2.72	8.50E-09	21	2.60	1.03E-08
	HSGS	22	0.68	2.43E-08	22	0.70	3.40E-08	21	0.66	4.13E-08
4096	FSJ	32	15.28	1.51E-09	31	14.84	2.11E-09	31	14.73	2.57E-09
	FSGS	22	10.74	1.52E-09	22	10.79	2.12E-09	21	10.17	2.58E-09
	HSGS	22	2.82	6.07E-09	22	2.70	8.50E-09	21	2.56	1.03E-08

Table 3. Comparison between FSJ and GS iteration family for Example 2.

G	Methods	r = 1.0			r = 0.6			r = 0.3		
		K	**Time**	**HD**	**K**	**Time**	**HD**	**K**	**Time**	**HD**
256	FSJ	46	0.08	1.14E-05	46	0.08	1.30E-05	46	0.08	1.41E-05
	FSGS	28	0.05	1.14E-05	28	0.06	1.30E-05	28	0.05	1.41E-05
	HSGS	28	0.02	4.58E-05	28	0.02	5.19E-05	28	0.02	5.65E-05
512	FSJ	46	0.30	2.86E-06	46	0.31	3.24E-06	46	0.30	3.53E-06
	FSGS	28	0.20	2.86E-06	28	0.20	3.24E-06	28	0.20	3.53E-06
	HSGS	28	0.06	1.14E-05	28	0.06	1.30E-05	28	0.06	1.41E-05
1024	FSJ	46	1.18	7.15E-07	46	1.17	8.11E-07	46	1.18	8.82E-07
	FSGS	28	0.73	7.15E-07	28	0.73	8.11E-07	28	0.74	8.82E-07
	HSGS	28	0.19	2.86E-06	28	0.23	3.24E-06	28	0.20	3.53E-06
2048	FSJ	46	4.64	1.79E-07	46	4.61	2.03E-07	46	4.41	2.20E-07
	FSGS	28	2.88	1.79E-07	28	2.91	2.03E-07	28	2.90	2.21E-07
	HSGS	28	0.71	7.15E-07	28	0.73	8.11E-07	28	0.71	8.82E-07
4096	FSJ	46	15.21	4.47E-08	46	16.64	5.06E-08	46	17.21	5.51E-08
	FSGS	28	11.37	4.47E-08	28	11.34	5.07E-08	28	11.46	5.51E-08
	HSGS	28	2.95	1.79E-07	28	2.87	2.03E-07	28	2.84	2.21E-07

Table 4. Comparison between FSJ and GS iteration family for Example 3.

G	Methods	r = 1.0			r = 0.6			r = 0.3		
		K	**Time**	**HD**	**K**	**Time**	**HD**	**K**	**Time**	**HD**
256	FSJ	64	1.12	1.97E-05	64	1.14	3.51E-05	63	1.10	3.88E-05
	FSGS	38	0.66	1.97E-05	37	0.65	3.51E-05	37	0.64	3.88E-05
	HSGS	38	0.17	2.15E-04	37	0.17	3.84E-04	37	0.16	4.25E-04
512	FSJ	64	4.49	4.92E-06	64	4.48	8.78E-06	63	4.40	9.71E-06
	FSGS	38	2.65	4.92E-06	37	2.56	8.78E-06	37	2.56	9.71E-06
	HSGS	38	0.67	5.39E-05	37	0.66	9.62E-05	37	0.65	1.06E-04
1024	FSJ	64	17.39	1.23E-06	64	17.27	2.19E-06	63	17.03	2.43E-06
	FSGS	38	9.78	1.23E-06	37	8.53	2.19E-06	37	8.67	2.43E-06
	HSGS	38	2.64	1.35E-05	37	2.59	2.41E-05	37	2.58	2.66E-05
2048	FSJ	64	68.52	3.07E-07	64	68.96	5.49E-07	63	67.06	6.07E-07
	FSGS	38	31.90	3.08E-07	37	36.40	5.49E-07	37	36.69	6.07E-07
	HSGS	38	10.74	3.37E-06	37	10.09	6.02E-06	37	10.25	6.66E-06
4096	FSJ	64	243.28	7.68E-08	64	259.7	1.37E-07	63	266.39	1.52E-07
	FSGS	38	135.34	7.69E-08	37	158.50	1.37E-07	37	141.87	1.52E-07
	HSGS	38	25.82	8.44E-07	37	32.30	1.51E-06	37	37.81	1.66E-06

Table 5. Reduction percentages for the GS iteration family compared with FSJ in terms of the number of iterations and iteration time.

Methods	**Number of iterations**		
	Example 1	**Example 2**	**Example 3**
FSGS and HSGS	29.03-32.26%	39.13%	40.63-42.19%
Methods	**Iteration time (in seconds)**		
	Example 1	**Example 2**	**Example 3**
FSGS	14.29-31.40%	25.00-38.14%	38.97-53.44%
HSGS	71.43-82.83%	75.00-84.70%	84.33-89.39%

5. Conclusions

In this study, we have solved the FFIE-2 numerically using HSGS iterative method. Three numerical examples have been conducted to illustrate the efficiency of the HSGS iterative method. Based on the performance analysis between FSJ, FSGS, and HSGS method, the results show that HSGS iterative method is superior to FSJ and

FSGS in terms of execution time. Since these three proposed iterative methods can be classified as a family of point iterative methods without using any weighted parameter. Thus, future work could be extended by considering a family of weighted point iterative methods such as in [27-29].

Nomenclatures

a	Upper boundary
A	Matrix coefficient
A_j	An independent numerical coefficient
b	Lower boundary
D	Diagonal matrix
$\underline{\tilde{f}}$	Vector coefficient
$f(x)$	Function of x on interval $[a, b]$
$\tilde{f}(x, r)$	Fuzzy function of x on interval $[a, b]$
h	Constant step
i	Pivot line to-i
j	Pivot line to-j
k	Iteration
$k(x, t)$	Arbitrary kernel function over the square $a \leq x, t \leq b$
L	Strictly lower triangular matrix
n	Number of subintervals
r	Fuzzy parameter
t_j	Abscissa of the partition points of integrations
U	Strictly upper triangular matrix
$\underline{\tilde{u}}$	Vector solution
$\underline{u}$	Lower limit of fuzzy function
$\overline{u}$	Upper limit of fuzzy function
$\tilde{u}(x, r)$	Fuzzy function towards x with unknown r

Greek Symbols

λ	Positive parameter
ε	Absolute error
$\tilde{\varepsilon}_n(t)$	Truncation error

Abbreviations

FFIE-2	Fuzzy Fredholm Integral Equations of the Second Kind
FSGS	Full-Sweep Gauss-Seidel
FSJ	Full-Sweep Jacobi
HD	Hausdorff distance
HSGS	Half-Sweep Gauss-Seidel

References

1. Shamivand, M.M.; Shahsavaran, A.; and Tari, S.M. (2011). Solution of Fredholm fuzzy integral equations with degenerate kernel. *International Journal of Contemporary Mathematics Sciences*, 6(11), 535-543.

2. Amirfakhrian, M.; Shakibi, K.; and Lopez, R.R. (2016). Fuzzy quasi-interpolation solution for Fredholm fuzzy integral equations of second kind. *Soft Computing*, 21(15), 4323-4333.
3. Molabahrami, A.; Shidfar, S.; and Ghyasi, A. (2011). An analytical method for solving linear Fredholm fuzzy integral equations of the second kind. *Computer and Mathematics with Applications*, 61(9), 2754-2761.
4. Vali, M.A.; Agheli, M.J.; and Nezhad, S.G. (2014). Homotopy analysis method to solve two-dimensional fuzzy Fredholm integral equation. *General Mathematics Notes*, 22(1), 31-43.
5. Rivaz, A.; and Yousefi, F. (2012). Modified homotopy perturbation method for solving two-dimensional fuzzy Fredholm integral equation. *International Journal of Applied Mathematics*, 25(4), 591-602.
6. Amawi, M.; and Qatanani, N. (2016). Analytical methods for solving fuzzy Fredholm integral equation of the second kind. *Research Journal of Applied Sciences*, 11(12), 1559-1568.
7. Khezerloo, S.; Allahviranloo, T.; Ghasemi, S.H.; Salahshour, S.; Khezerloo, M.; and Kiasary, M.K. (2010). Expansion method for solving fuzzy Fredholm-Volterra integral equations. *Communications in Computer and Information Science*, 81, 501-511.
8. Gholam, A.M.; and Ezzati, R. (2017). Solving linear fuzzy Fredholm integral equations of the second kind via iterative method and Simpson quadrature rule: A review. *TWMS Journal of Pure and Applied Mathematics*, 8(2), 121-147.
9. Mahaleh, V.S.K.; and Ezzati, R. (2017). Numerical solution of linear fuzzy Fredholm integral equations of second kind using iterative method and midpoint quadrature formula. *Journal of Intelligent and Fuzzy Systems*, 33(3), 1293-1302.
10. Jahantigh, M.; Allahviranloo, T.; and Otadi, M. (2008). Numerical solution of fuzzy integral equations. *Applied Mathematical Sciences*, 2(1), 33-46.
11. Mirzaee, F.; Paripour, M.; and Yari, M.K. (2014). Application of triangular and delta basis functions to solve linear Fredholm fuzzy integral equation of the second kind. *Arabian Journal for Science and Engineering*, 39(5), 3969-3978.
12. Allahviranloo, T.; Khezerloo, M.; Ghanbari, M.; and Khezerloo, S. (2010). The homotopy perturbation method for fuzzy Volterra integral equations. *International Journal of Computational Cognition*. 8(2): 31-37.
13. Ghanbari, M. (2010). Numerical solution of fuzzy linear Volterra integral equations of the second kind by homotopy analysis method. *International Journal of Industrial Mathematics*, 2(2), 73-87.
14. Hamoud, A.A.; Azeez, A.D.; and Ghadle, K.P. (2018). A study of some iterative methods for solving fuzzy Volterra-Fredholm integral equations. *Indonesian Journal of Electrical Engineering and Computer Science*, 11(3), 1228-1235.
15. Hamoud, A.A.; and Ghadle, K.P. (2018). Modified Adomian decomposition method for solving fuzzy Volterra-Fredholm integral equation. *Journal of the Indian Mathematical Society*, 85(1-2), 52-69.
16. Georgeiva, A. (2018). Solving two-dimensional nonlinear Volterra-Fredholm fuzzy integral equations by using Adomian decomposition method. *Dynamic Systems and Applications*, 27(4), 819-835.

17. Ali, L.H.; Sulaiman, J.; and Hashim, S.R.M. (2018). Numerical solution of SOR iterative method for fuzzy Fredholm integral equations of second kind. *Proceeding of the International Conference on Mathematics, Engineering and Industrial Applications*. Kuala Lumpur, Malaysia, 1-8.
18. Ali, L.H.; Sulaiman, J.; and Hashim, S.R.M. (2018). SOR iterative method with Simpson's 1/3 rule for the numerical solution of fuzzy second kind Fredholm integral equations. *The Proceedings of IOP Conference Series: Journal of Physics: Conference Series*. Kuala Lumpur, Malaysia, 1-9.
19. Abdullah, A.R. (1991). The four point explicit decoupled group (EDG) method: a fast Poisson solver. *International Journal Computer Mathematics*, 38(1-2), 61-70.
20. Muthuvalu, M.S.; and Sulaiman, J. (2011). Half-Sweep arithmetic mean method with composite trapezoidal scheme for solving linear Fredholm integral equations. Applied Mathematics and Computation, 217(12), 5442-5448.
21. Dahalan, A.A.; Muthuvalu, M.S.; and Sulaiman, J. (2015). Half-Sweep successive over relaxation iterative method to solve two-point fuzzy boundary value problems. *Proceeding of the 22nd National Symposium on Mathematical Sciences: Strengthening Research and Collaboration of Mathematical Sciences in Malaysia*. Selangor, Malaysia, 1-8.
22. Suardi, M.N.; Radzuan, N.Z.F.M.; and Sulaiman, J. (2017). Cubic B-spline solution of two-point boundary value problem using HSKSOR iteration. *Global Journal of Pure and Applied Mathematics*, 13(11), 7921-7934.
23. Saudi, A.; and Sulaiman, J. (2009). Half-Sweep Gauss-Seidel (HSGS) iterative method for robot path planning. *Proceeding of the 3rd International Conference on Informatics and Technology*. Kuala Lumpur, Malaysia, 34-39.
24. Eng, J.H.; Saudi, A.; and Sulaiman, J. (2018). Implementation of quarter-sweep approach in Poisson image blending problem. *Computational Science and Technology*, 481, 127-136.
25. Yaacob, Z.; and Hasan, M.K. (2014). Family of Gauss-Seidel method for solving 2D Reynolds equation in hydrodynamic lubrication problem. *Proceedings of the 1st International Conference of Recent Trends in Information and Communication Technologies*. Johor, Malaysia, 361-369.
26. Bakari, A.I.; and Dahiru, I.A. (2018). Comparison of Jacobi and Gauss-Seidel iterative methods for the solution of systems of linear equations. *Asian Research Journal of Mathematics*, 8(3), 1-7.
27. Youssef, I.K. (2012). On the successive overrelaxation method. *Journal of Mathematics and Statistics*, 8(2), 176-184.
28. Youssef, I.K.; and Farid, M.M. (2015). On the accelerated overrelaxation method. *Pure and Applied Mathematics Journal*, 4(1), 26-31.
29. Youssef, I.K.; and Taha, A.A. (2013). On the modified successive overrelaxation method. *Applied Mathematics and Computation*, 219(9), 4601-4613.