AGE Iterative Method with First-Order Quadrature Scheme for Fuzzy Fredholm Integral Equations of Second Kind

Labiyana Hanif Ali¹, Jumat Sulaiman² and Siti Rahayu Mohd Hashim³

¹,²,³ Mathematics with Economics Program, Faculty of Science and Natural Resources, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia

E-mail: labiyana15@gmail.com

Abstract. In this study, we propose Alternating Group Explicit (AGE) iterative method to solve the problem of linear fuzzy Fredholm integral equations of the second kind (FFIE-2) based on Trapezoidal quadrature rule. First, we generate a system of linear equations by using approximation equations. Next, we use AGE iterative method to solve the generated linear system of FFIE-2. Then, we illustrate the applicability of AGE iterative method on some numerical examples. Finally, we do the comparison to test the efficiency of AGE with Jacobi and Gauss-Seidel (GS) iterative method based on three parameters: number of iterations, execution time and Hausdorff distance. From our findings, we found that AGE is a better iterative method compared with Jacobi and GS to solve FFIE-2.

1. Introduction

In recent years, the topic of fuzzy integral equations have been widely applied known as one of the most important topics among the researchers due to its wide application in various research fields [1,2]. Fuzzy integral equations consist of fuzzy linear and fuzzy nonlinear integral equations. However, in this study, we only focus on the case of linear fuzzy integral equations. Fuzzy Fredholm integral equations are well-known as an important class of fuzzy integral equations [3]. Therefore, this study aims to solve fuzzy Fredholm integral equations of the second kind (FFIE-2). Basically, the problem of any integral equations can be solved either by analytical or numerical method. However, many studies have been done by using a numerical approach to solve FFIE-2. The problem of FFIE-2 have been solved numerically by using fuzzy transforms [4], Taylor expansion method [5], Bernoulli wavelet method [3], triangular function [6], hat function [7], radial basis function [8], block pulse function [9], triangular and delta basis functions [10] and others [11-15].

In this study, we proposed the Alternating Group Explicit (AGE) iterative method with first-order quadrature rule for solving linear FFIE-2. Consider the general form of Fredholm integral equations of the second kind given by [16]

\[ x(s) = f(s) + \lambda \int_a^b k(s, t)x(t)dt, \]  

where \( \lambda > 0, k(s, t) \) is an arbitrary kernel function over the square \( a \leq s, t \leq b \) and \( f(s) \) is a function of \( s: a \leq s \leq b \). The solution of problem (1) is a crisp if \( f(s) \) is a crisp function and the solution is a fuzzy if \( f(s) \) is a fuzzy function. Therefore, the general form of fuzzy Fredholm integral equation of the second kind also can be given by [1,16]

\[ \tilde{x}(s, r) = \tilde{f}(s, r) + \lambda \int_a^b k(s, t)\tilde{x}(t, r)dt, \]
where $\tilde{x}(s, r) = \left(\tilde{x}(s, r), \tilde{x}(s, r)\right)$ and $\tilde{f}(s, r) = \left(\tilde{f}(s, r), \tilde{f}(s, r)\right)$ with $0 \leq r \leq 1$.

There are several sections in this paper. In the next section, we generate a linear system of FFIE-2 from the discretization process based on Trapezoidal approximation equations. Then, we discuss the formulation of AGE to solve problem (2). Next, we introduce three numerical problem (2) to test the effectiveness of our proposed method. Lastly, we discuss the effectiveness of AGE by comparing it with Jacobi and Gauss-Seidel (GS).

2. Trapezoidal Approximation Equation

In this section, we consider Trapezoidal discretization scheme to get the approximation equations of linear FFIE-2. Then, we use the approximation equations to generate a linear system of FFIE-2. Now, let us consider the following general quadrature scheme based on Newton-Cotes equations for fuzzy functions, $y(t, r)$, and $\tilde{y}(t, r)$ which given as follows [17,18]

$$\int_{a}^{b} y(t, r) dt = \sum_{j=0}^{n} A_j y(t_j, r) + \epsilon_n(y).$$  (3)

$$\int_{a}^{b} \tilde{y}(t, r) dt = \sum_{j=0}^{n} A_j \tilde{y}(t_j, r) + \epsilon_n(y).$$  (4)

where $A_j$ is an independent numerical coefficient as follows [19]

$$A_j = \begin{cases} \frac{1}{2} h, & j = 0, n \\ h, & otherwise \end{cases}$$

while $t_j$ is the quadrature point on the interval $[a, b]$ and $\epsilon_n(y)$ is the truncation error. Both equations (3) and (4) also can be represented as in equation (6)

$$\int_{a}^{b} y(t, r) dt = \sum_{j=0}^{n} A_j y(t_j, r) + \epsilon_n(y).$$  (6)

Next, we consider the Trapezoidal integral equation of function $y(t, r)$ on interval $[x_i, x_{i+1}]$ which is defined as follows

$$\int_{x_i}^{x_{i+1}} y(t, r) dt = \frac{h}{2} (f_i + f_{i+1}),$$

where $h$ is constant step which given by

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

and $n$ is the number of subintervals in the interval $[a, b]$. Interval $[a, b]$ can be divided into $\{x_0, x_1, x_2, \ldots, x_n\}$ to become $a = x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n = b$ as shown in Figure 1.

![Figure 1. Finite grid networks and uniform distribution of node points for solution domain $[a, b]$.](image)

Now, by taking $n = 3$ and $\lambda = 1$, we apply the Trapezoidal integral equation as stated on equation (7) into problem (2) to get the following equation

$$x(s, r) = f(s, r) + \frac{h}{2} [k(s, t_0)x(t_0, r) + 2k(s, t_1)x(t_1, r) + 2k(s, t_2)x(t_2, r) + k(s, t_3)x(t_3, r)].$$  (9)

Next, by simplifying (9), we rewrite the equation as follows

$$x(s) = f(s) + \frac{h}{2} [k(s, t_0)x(t_0) + 2k(s, t_1)x(t_1) + 2k(s, t_2)x(t_2) + k(s, t_3)x(t_3)].$$  (10)

Now by considering four node points, $s = s_0, s = s_1, s = s_2$ and $s = s_3$ and referring to (5), we can have the following approximate equations for each node points in Figure 1 as follows
\[
\begin{align*}
(1 - A_0(k_{0,0}))x_0 - A_1(k_{0,1})x_1 - A_2(k_{0,2})x_2 - A_3(k_{0,3})x_3 &= f_0 \\
-A_0(k_{1,0})x_0 + (1 - A_1(k_{1,1}))x_1 - A_2(k_{1,2})x_2 - A_3(k_{1,3})x_3 &= f_1 \\
-A_0(k_{2,0})x_0 - A_1(k_{2,1})x_1 + (1 - A_2(k_{2,2}))x_2 - A_3(k_{2,3})x_3 &= f_2 \\
-A_0(k_{3,0})x_0 - A_1(k_{3,1})x_1 - A_2(k_{3,2})x_2 + (1 - A_3(k_{3,3}))x_3 &= f_3
\end{align*}
\]

Finally, we can present the generated system of linear equation (11) to (14) for \((n + 1)\) node points in Figure 1 into a general matrix form as follows
\[
A\tilde{x} = \tilde{f},
\]
where
\[
A = \begin{bmatrix}
1 - A_0(k_{0,0}) & -A_1(k_{0,1}) & -A_2(k_{0,2}) & \cdots & -A_n(k_{0,n}) \\
-A_0(k_{1,0}) & 1 - A_1(k_{1,1}) & -A_2(k_{1,2}) & \cdots & -A_n(k_{1,n}) \\
-A_0(k_{2,0}) & -A_1(k_{2,1}) & 1 - A_2(k_{2,2}) & \cdots & -A_n(k_{2,n}) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
-A_0(k_{n,0}) & -A_1(k_{n,1}) & -A_2(k_{n,2}) & \cdots & 1 - A_n(k_{n,n})
\end{bmatrix},
\]
\[
\tilde{x} = [x_0, x_1, x_2, \ldots, x_n]^T,
\]
\[
\tilde{f} = [f_0, f_1, f_2, \ldots, f_n]^T.
\]

3. Formulation of the Alternating Group Explicit

Referring to the system of linear equation (15) in the previous section, now we proceed to the formulation of AGE iterative method to solve equation (15). The efficiency of this proposed method will be tested into three numerical examples of linear FFIE-2 in the next sections. AGE iterative method was introduced by Evans [20]. Evans [20,21] stated that AGE method consists of splitting the coefficient matrix \(A\) into
\[
A = G_1 + G_2,
\]
where matrix \(G_1\) and \(G_2\) satisfy the following condition:

i. \(G_1 + rI\) and \(G_2 + rI\) are nonsingular for any \(r > 0\).

ii. For any vectors \(v_1\) and \(v_2\) and for any constant \(r > 0\), it is convenient to solve the linear systems explicitly; ie.,
\[
x^{(k+1/2)} = G_1^{-1}v_1,
\]
\[
x^{(k+1)} = G_2^{-1}v_2,
\]
for \(x^{(k+1/2)}\) and \(x^{(k+1)}\), respectively.

Since \(n\) is an odd number, the diagonal on matrix \(A\) represented by \(d'_i\) and can be defined as follows
\[
d'_i = \frac{d_i}{2}, \quad i = 1, 2, \ldots, n,
\]
where \(G_1\) and \(G_2\) satisfy the condition that \((G_1 + rI)\) and \((G_2 + rI)\) are nonsingular for any \(r > 0\).

By applying equation (16), equation (15) can be written as follows
\[
(G_1 + G_2)\tilde{x} = \tilde{f},
\]
where
\[
G_i = \begin{bmatrix}
0.5(1 - A_0K_{0,0}) & -A_1K_{0,1} & -A_0K_{0,2} & \cdots & -A_nK_{0,n} \\
-A_0K_{1,0} & 0.5(1 - A_1K_{1,1}) & -A_2K_{1,2} & \cdots & -A_nK_{1,n} \\
-A_0K_{2,0} & -A_1K_{2,1} & 0.5(1 - A_2K_{2,2}) & \cdots & -A_nK_{2,n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
-A_0K_{n,0} & -A_1K_{n,1} & -A_2K_{n,2} & \cdots & 0.5(1 - A_nK_{n,n})
\end{bmatrix},
\]
\[
d_{i} = \frac{d_i}{2}, \quad i = 1, 2, \ldots, n,
\]

Finally, we can present the generated system of linear equation (11) to (14) for \((n + 1)\) node points in Figure 1 into a general matrix form as follows
\[
A\tilde{x} = \tilde{f},
\]
where
\[
A = \begin{bmatrix}
1 - A_0(k_{0,0}) & -A_1(k_{0,1}) & -A_2(k_{0,2}) & \cdots & -A_n(k_{0,n}) \\
-A_0(k_{1,0}) & 1 - A_1(k_{1,1}) & -A_2(k_{1,2}) & \cdots & -A_n(k_{1,n}) \\
-A_0(k_{2,0}) & -A_1(k_{2,1}) & 1 - A_2(k_{2,2}) & \cdots & -A_n(k_{2,n}) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
-A_0(k_{n,0}) & -A_1(k_{n,1}) & -A_2(k_{n,2}) & \cdots & 1 - A_n(k_{n,n})
\end{bmatrix},
\]
\[
\tilde{x} = [x_0, x_1, x_2, \ldots, x_n]^T,
\]
\[
\tilde{f} = [f_0, f_1, f_2, \ldots, f_n]^T.
\]
\[
G_2 = \begin{bmatrix}
0.5(1 - A_0 K_{0,0}) & 0.5(1 - A_1 K_{1,1}) & -A_2 K_{1,2} \\
-A_1 K_{2,1} & 0.5(1 - A_2 K_{2,2}) \\
& & \vdots \\
& & 0.5(1 - A_{j-1} K_{j-1,j-1}) & -A_j K_{j-1,j} \\
& & -A_j K_{j,i-1} & 0.5(1 - A_j K_{j,j})
\end{bmatrix}.
\]

Now, by selecting a positive iteration parameter \( r \) and an initial guess, \( x^{(0)} \) of the solution, and carrying out the iterative method which is defined by

\[
(G_1 + rI)x^{(k+1/2)} = f - (G_2 + rI)x^{(k)},
\]

(18)

\[
(G_2 + rI)x^{(k+1)} = f - (G_1 + rI)x^{(k+1/2)}.
\]

(19)

Next, we used the formulation (18) and (19) to write the general algorithm of AGE to solve equation (17). The algorithm of AGE iteration can be presented as follows

Algorithm 1: AGE iteration
i. Set the initial value \( x^{(0)} = 0 \) and tolerance error \( \varepsilon = 10^{-10} \).
ii. Calculate coefficient matrix \( A \) and vector \( f \).
iii. For first stage calculate,
\[
x^{(k+1/2)} = (G_1 + rI)^{-1}[f - (G_2 + rI)x^{(k)}].
\]
iv. For second stage calculate,
\[
x^{(k+1)} = (G_2 + rI)^{-1}[f - (G_1 + rI)x^{(k+1/2)}].
\]
v. Run the convergence test \( |x_i^{(k+1/2)} - x_i^{(k+1)}| \leq \varepsilon = 10^{-10} \). If yes, proceed to step vi.
Otherwise, go back to step iii.
vi. Display numerical solutions.

4. Numerical Examples
As mention in Section 1, we consider three numerical examples to test the efficiency of our proposed method together with Jacobi and GS. We set Jacobi iterative method as the comparison control in order to investigate the performance between AGE with Jacobi and GS. Then, we do performance analysis by using three criterions which are number of iterations (K), execution time (Time) in seconds and Hausdorff distance (HD). Throughout the simulations, we consider the tolerance error at \( \varepsilon = 10^{-10} \). Then, we carried out the simulations on several selected grid sizes \( n = 256, 512, 1024, 2048 \) and 4096 on the value of parameter fuzzy \( r = 1.0, 0.6 \) and 0.3. All three examples of FFIE-2 are given as follows [15,22]

Example 1
Given FFIE-2

\[
f(s, r) = -\frac{1}{52} r (5 - 52s + 2s^2),
\]

(20)

\[
\bar{f}(s, r) = \frac{1}{52} (r - 2)(5 - 52s + 2s^2),
\]

(21)

and kernel

\[
k(s, t) = \frac{s^2 + t^2 + 2}{13}, 0 \leq s, t \leq 1 \text{ and } \lambda = 1
\]

(22)

and \( \alpha = 0, b = 1 \). The exact solution given by

\[
\chi(s, r) = rs,
\]

(23)

\[
\bar{x}(s, r) = (2 - r)s.
\]

(24)
Example 2
Given FFIE-2

\[ f(s,r) = \frac{2}{3}(2 + r)s, \]
\[ \bar{f}(s,r) = -\frac{2}{3}(r - 4)s, \]

and kernel

\[ k(s,t) = st, 0 \leq s, t \leq 1 \text{ and } \lambda = 1 \]

and \( a = 0, b = 1 \). The exact solutions are given by

\[ \chi(s,r) = (2 + r)s, \]
\[ \bar{\chi}(s,r) = (4 - r)s. \]

Example 3
Given FFIE-2

\[ f(s,r) = rs(r^4 + 2)(3\cos(1 - s) + 5\sin(1 - s) - 6\cos(s) + s^2), \]
\[ \bar{f}(s,r) = -3s(r^3 - 2)(3\cos(1 - s) + 5\sin(1 - s) - 6\cos(s) + s^2), \]

and kernel

\[ k(s,t) = s\cos(t - s), 0 \leq s, t \leq 1 \text{ and } \lambda = 1 \]

and \( a = 0, b = 1 \). The exact solutions are given by

\[ \chi(s,r) = s^3(r^5 + 2r), \]
\[ \bar{\chi}(s,r) = s^3(6 - 3r^3). \]

5. Discussions
In this section, we present the numerical results obtained from the implementation of Jacobi, GS and AGE on Examples 1, 2 and 3 from the previous section. Table 1, 2 and 3 show the comparison between Jacobi, GS and AGE for Examples 1, 2 and 3 respectively. Meanwhile, Table 4 shows the percentages of reduction for the GS and AGE compared with Jacobi based on the number of iterations and execution time. Based on the numerical results recorded in Table 1, 2 and 3, we found that the AGE iterative method recorded the smallest number of iterations and fastest iteration time compared with GS and Jacobi for all selected grid size, \( n \). Besides that, we also found that the HD of all iterative methods increase together with the increasing of grid size and there is no much difference between the value of parameter fuzzy \( r = 1.0, 0.6 \) and 0.3 for all examples. Referring to Table 4, we found that AGE recorded higher reduction percentages followed by GS compared with Jacobi in terms of the number of iteration and iteration time for all examples.
Table 1. Comparison of Jacobi, GS and AGE for Example 1.

| n  | Methods | \( r = 1.0 \) |       | \( r = 0.6 \) |       | \( r = 0.3 \) |       |
|----|---------|-------------|-------|-------------|-------|-------------|-------|
|    |         | K | Time | HD   | K | Time | HD   | K | Time | HD   |
| 256| Jacobi  | 32 | 0.07 | 3.88E-07 | 31 | 0.07 | 5.44E-07 | 31 | 0.07 | 6.60E-07 |
|    | GS      | 22 | 0.06 | 3.88E-07 | 22 | 0.06 | 5.44E-07 | 21 | 0.06 | 6.60E-07 |
|    | AGE     | 10 | 0.05 | 3.88E-07 | 10 | 0.05 | 5.44E-07 | 10 | 0.04 | 6.60E-07 |
| 512| Jacobi  | 32 | 0.27 | 9.71E-08 | 31 | 0.26 | 1.36E-07 | 31 | 0.25 | 1.65E-07 |
|    | GS      | 22 | 0.19 | 9.71E-08 | 22 | 0.18 | 1.36E-07 | 21 | 0.19 | 1.65E-07 |
|    | AGE     | 10 | 0.15 | 9.71E-08 | 10 | 0.17 | 1.36E-07 | 10 | 0.16 | 1.65E-07 |
| 1024| Jacobi | 32 | 1.00 | 2.43E-08 | 31 | 0.97 | 3.40E-08 | 31 | 0.96 | 4.12E-08 |
|    | GS      | 22 | 0.70 | 2.43E-08 | 22 | 0.69 | 3.40E-08 | 21 | 0.67 | 4.13E-08 |
|    | AGE     | 10 | 0.62 | 2.43E-08 | 10 | 0.63 | 3.40E-08 | 10 | 0.61 | 4.13E-08 |
| 2048| Jacobi | 32 | 3.96 | 6.06E-09 | 31 | 3.80 | 8.48E-09 | 31 | 3.79 | 1.03E-08 |
|    | GS      | 22 | 2.74 | 6.07E-09 | 22 | 2.72 | 8.50E-09 | 21 | 2.60 | 1.03E-08 |
|    | AGE     | 10 | 2.40 | 6.07E-09 | 10 | 2.41 | 8.50E-09 | 10 | 2.38 | 1.03E-08 |
| 4096| Jacobi | 32 | 15.28 | 1.51E-09 | 31 | 14.84 | 2.11E-09 | 31 | 14.73 | 2.57E-09 |
|    | GS      | 22 | 10.74 | 1.52E-09 | 22 | 10.79 | 2.12E-09 | 21 | 10.17 | 2.58E-09 |
|    | AGE     | 10 | 8.48 | 1.52E-09 | 10 | 9.51 | 2.12E-09 | 10 | 9.51 | 2.58E-09 |

Table 2. Comparison of Jacobi, GS and AGE for Example 2.

| n  | Methods | \( r = 1.0 \) |       | \( r = 0.6 \) |       | \( r = 0.3 \) |       |
|----|---------|-------------|-------|-------------|-------|-------------|-------|
|    |         | K | Time | HD   | K | Time | HD   | K | Time | HD   |
| 256| Jacobi  | 46 | 0.08 | 1.14E-05 | 46 | 0.08 | 1.30E-05 | 46 | 0.08 | 1.41E-05 |
|    | GS      | 28 | 0.05 | 1.14E-05 | 28 | 0.06 | 1.30E-05 | 28 | 0.05 | 1.41E-05 |
|    | AGE     | 12 | 0.01 | 1.14E-05 | 12 | 0.01 | 1.30E-05 | 12 | 0.01 | 1.41E-05 |
| 512| Jacobi  | 46 | 0.30 | 2.86E-06 | 46 | 0.31 | 3.24E-06 | 46 | 0.30 | 3.53E-06 |
|    | GS      | 28 | 0.20 | 2.86E-06 | 28 | 0.20 | 3.24E-06 | 28 | 0.20 | 3.53E-06 |
|    | AGE     | 12 | 0.05 | 2.86E-06 | 12 | 0.04 | 3.24E-06 | 12 | 0.05 | 3.53E-06 |
| 1024| Jacobi | 46 | 1.18 | 7.15E-07 | 46 | 1.17 | 8.11E-07 | 46 | 1.18 | 8.82E-07 |
|    | GS      | 28 | 0.73 | 7.15E-07 | 28 | 0.73 | 8.11E-07 | 28 | 0.74 | 8.82E-07 |
|    | AGE     | 12 | 0.18 | 7.15E-07 | 12 | 0.19 | 8.11E-07 | 12 | 0.19 | 8.82E-07 |
| 2048| Jacobi | 46 | 4.64 | 1.79E-07 | 46 | 4.61 | 2.03E-07 | 46 | 4.41 | 2.20E-07 |
|    | GS      | 28 | 2.88 | 1.79E-07 | 28 | 2.91 | 2.03E-07 | 28 | 2.90 | 2.21E-07 |
|    | AGE     | 12 | 0.72 | 1.79E-07 | 12 | 0.73 | 2.03E-07 | 12 | 0.73 | 2.21E-07 |
| 4096| Jacobi | 46 | 15.21 | 4.47E-08 | 46 | 16.64 | 5.06E-08 | 46 | 17.21 | 5.51E-08 |
|    | GS      | 28 | 11.37 | 4.47E-08 | 28 | 11.34 | 5.07E-08 | 28 | 11.46 | 5.51E-08 |
|    | AGE     | 12 | 2.82 | 4.47E-08 | 12 | 2.87 | 5.07E-08 | 12 | 2.83 | 5.51E-08 |
Table 3. Comparison of Jacobi, GS and AGE for Example 3.

| n     | Methods | $r = 1.0$ | $r = 0.6$ | $r = 0.3$ |
|-------|---------|-----------|-----------|-----------|
|       |         | K Time    | HD        | K Time    | HD        | K Time    | HD        |
| 256   | Jacobi  | 64 1.12   | 1.97E-05  | 64 1.14   | 3.51E-05  | 64 1.10   | 3.88E-05  |
|       | GS      | 38 0.66   | 1.97E-05  | 37 0.65   | 3.51E-05  | 37 0.64   | 3.88E-05  |
|       | AGE     | 14 0.14   | 1.97E-05  | 14 0.15   | 3.51E-05  | 14 0.14   | 3.88E-05  |
| 512   | Jacobi  | 64 4.49   | 4.92E-06  | 64 4.48   | 8.78E-06  | 64 4.40   | 9.71E-06  |
|       | GS      | 38 2.65   | 4.92E-06  | 37 2.56   | 8.78E-06  | 37 2.56   | 9.71E-06  |
|       | AGE     | 14 0.58   | 4.92E-06  | 14 0.58   | 8.78E-06  | 14 0.58   | 9.71E-06  |
| 1024  | Jacobi  | 64 17.39  | 1.23E-06  | 64 17.27  | 2.19E-06  | 63 17.03  | 2.43E-06  |
|       | GS      | 38 9.78   | 1.23E-06  | 37 8.53   | 2.19E-06  | 37 8.67   | 2.43E-06  |
|       | AGE     | 14 2.29   | 1.23E-06  | 14 2.28   | 2.19E-06  | 14 2.29   | 2.43E-06  |
| 2048  | Jacobi  | 64 68.52  | 3.07E-07  | 64 68.96  | 5.49E-07  | 63 67.06  | 6.07E-07  |
|       | GS      | 38 31.90  | 3.08E-07  | 37 36.40  | 5.49E-07  | 37 36.69  | 6.07E-07  |
|       | AGE     | 14 9.06   | 3.08E-07  | 14 9.09   | 5.49E-07  | 14 9.08   | 6.07E-07  |
| 4096  | Jacobi  | 64 243.28 | 7.68E-08  | 64 259.70 | 1.37E-07  | 63 266.39 | 1.52E-07  |
|       | GS      | 38 135.34 | 7.69E-08  | 37 158.50 | 1.37E-07  | 37 141.87 | 1.52E-07  |
|       | AGE     | 14 36.25  | 7.69E-08  | 14 36.19  | 1.37E-07  | 14 36.63  | 1.52E-07  |

Table 4. Reduction percentages of GS and AGE compared with Jacobi.

| Methods | Number of iterations | Example 1 | Example 2 | Example 3 |
|---------|----------------------|-----------|-----------|-----------|
| GS      | 29.03-32.26%         | 39.13%    | 40.63-42.19% |
| AGE     | 67.74-68.75%         | 73.91%    | 77.78-78.13% |

| Methods | Execution time | Example 1 | Example 2 | Example 3 |
|---------|----------------|-----------|-----------|-----------|
| GS      | 14.29-31.40%    | 25.00-38.14% | 38.97-53.44% |
| AGE     | 28.57-44.50%    | 81.46-87.50% | 85.10-87.50% |

6. Conclusions
Throughout this study, we managed to get the approximation equations of FFIE-2 by using first-order quadrature scheme. Based on the approximation equations obtained, we have formed a linear system of FFIE-2. Then, we have discussed the formulation and algorithm of AGE iterative method to solve the generated linear system of FFIE-2. Besides that, we also have discussed the effectiveness of AGE by comparing its performance with Jacobi and GS on three numerical examples. Through the numerical results obtained, we conclude that the AGE iterative method is a better method compared with Jacobi and GS in terms of iterations number and execution time. For future work, the AGE with higher order discretization scheme can be considered in solving FFIE-2.

7. References
[1] Shamivand M M, Shashaavan A and Tari S M 2011 *Int. J. of Contemporary Mathematics Sciences* 6 pp 535-43
[2] Amirfakhrian M, Shakibi K and Lopez R R 2016 *Soft Comput.* 21 pp 4323-33
[3] Ramadan M A and Ali M R 2018 *J. of Modern Methods in Numerical Mathematics* 9 pp 16-27
[4] Ezzati R and Mokhtari F 2012 *Int. J. of Physical Sciences* 7 pp 1578-83
[5] Jafarian A, Nia S M, Tavan S and Banifazel M 2012 *Applied Mathematical Sciences* 6 pp 4103-17
[6] Mirzaee F, Paripour M and Yari M K 2012 *J. of Hyperstructures* 1 pp 46-60
[7] Mirzaee F, Paripour M and Yari M K 2014 *J. of Intelligent & Fuzzy Systems* 27 pp 211-20
[8] Parand K and Raisi E 2017 *Romai J.* 13 pp 57-69
Acknowledgments
The registration fee for attending Science and Technology Seminar is funded by Centre for Postgraduate Studies, Universiti Malaysia Sabah.