Journal of Engineering and Applied Sciences 11 (7): 1456-1463, 2016
ISSN: 1816-949X

Performance of Quarter-Sweep Successive over Relaxation Iterative Method for Two-Point Fuzzy Boundary Value Problems

[1]A.A. Dahalan, [1]N.S.A. Aziz and [2]J. Sulaiman
[1]Department of Mathematics, Centre for Defence Foundation Studies,
National Defence University of Malaysia, 57000 Kuala Lumpur, Malaysia
[2]Faculty of Science and Natural Resources, Universiti Malaysia Sabah,
88400 Kota Kinabalu, Sabah, Malaysia

Abstract: This study deals with the application of numerical methods in solving the Fuzzy Boundary Value Problems (FBVPs) which is discretized to derive second order fuzzy finite difference approximation equation. Then this fuzzy approximation equation is used to generate the fuzzy linear system. In addition to that, the fuzzy linear system will be solved iteratively by using Gauss-Seidel (GS), Full-Sweep Successive Over-Relaxation (FSSOR), Half-Sweep Successive Over Relaxation (HSSOR) and Quarter-Sweep Successive Over Relaxation (QSSOR) iterative methods. Then several numerical experiments are conducted to illustrate the effectiveness of QSSOR iterative method compared with the GS, FSSOR and QSSOR methods.

Key words: Fuzzy boundary value problem, weighted point iteration, finite difference scheme, linear, Malaysia

INTRODUCTION

Othman and Abdullah (2000) has expanded the half-sweep iteration approach which discovered by Abdullah (1991), to initiate the Modified Explicit Group (MEG) method based on the quarter-sweep approach. It is proved that this method is one of most efficient block iterative methods in solving any linear systems as compared with EG and EDG iterative methods. The fundamental idea of these iteration concept is to reduce the computational complexities of the original linear system methods. Due to the advantages of this iteration, further investigations have been conducted for demonstrating the capability of the concept (Akhir *et al*., 2012; Dahalan *et al*., 2013, 2014; Dahalan and Sulaiman, 2015a, b; Koh *et al*., 2010; Muthuvalu and Sulaiman , 2008, 2004; Sulaiman *et al*., 2004, 2009).

This study consider the Quarter-Sweep Successive Over-Relaxation (QSSOR) is used as linear solver to solve fuzzy linear systems generated from the discretization of the 2 point Fuzzy Boundary Value Problems (FBVPs), whereas Full-Sweep Successive Over Relaxation (FSSOR), Half-Sweep Successive Over Relaxation (HSSOR) and Gauss-Seidel (GS) iterative methods are used as control solvers.

MATERIALS AND METHODS

Half-sweep finite difference approximation equation: Consider 2 point linear FBVPs in general as:

$$x''(t)+j(t)x'(t)+q(t)x(t) = f(t), \qquad t\in[a,b] \quad (1)$$
$$x(a) = \sigma$$
$$x(b)= \varpi$$

Where:

x (t)	= Fuzzy function
f(t)	= Fuzzy continuous function
j (t) and q(t)	= Continuous functions on [a, b]

whereas, σ and ϖ are fuzzy number. Let $\tilde{x}$ be a fuzzy subset of real numbers. It is characterized by a membership function evaluated at t, written $\tilde{x}(t)$ as a number in [0, 1]. Fuzzy numbers can be identified through the membership function. The α-cut of $\tilde{x}$ which α is denoted as a crisp number can be written as $\tilde{x}(a)$ and define $\{t|\tilde{x}(t)\geq a\}$ for $0<\alpha\leq 1$. The α-cut of fuzzy numbers can be written as $\tilde{x}(a)=[\underline{x}(a),\bar{x}(a)]$ for all α since they are always closed and bounded interval (Allahviranloo, 2002). Suppose $(\underline{x},\bar{x})$ be parametric form of fuzzy function x. For arbitrary positive integer n subdivide the interval

Corresponding Author: A.A. Dahalan, Department of Mathematics, Centre for Defence Foundation Studies, National Defence University of Malaysia, 57000 Kuala Lumpur, Malaysia

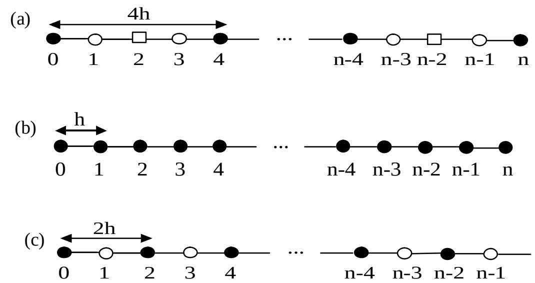

Fig. 1: a-c) Show distribution of uniform solid node points for the full, half and quarter-sweep cases, respectively

$t = [a,b]$ as $a = t_0<t_1<...t_{n-1}<t_n = b$ defined by $t_i = a+ih$ ($i = 0, 1, 2, ..., n$) and $h = (b-a)/n$. To simplify the formulation of the full, half and quarter-sweep second-order finite difference approximation equations, the finite grid network will be used as shown in Fig. 1. Implementations of these point iterative methods are executed onto the interior solid nodal points until convergence test is found. Meanwhile, the approximation solutions for the remaining points can be computed by using direct method (Abdullah, 1991; Ibrahim and Abdullah, 1995). At the representative point t_I denote the value of x and $(\underline{x},\overline{x})$ by x_I and $(\underline{x}_i,\overline{x}_i)$, respectively. By using the second-order central difference scheme, Eq. 1 can be developed as:

$$\underline{x}_i'' \approx \frac{\underline{x}_{i-p}-2\underline{x}_i+\underline{x}_{i+p}}{(ph)^2} \quad (2)$$

$$\overline{x}_i'' \approx \frac{\overline{x}_{i-p}-2\overline{x}_i+\overline{x}_{i+p}}{(ph)^2} \quad (3)$$

And:

$$\underline{x}_i' \approx \frac{\underline{x}_{i+p}-\underline{x}_{i-p}}{2ph} \quad (4)$$

$$\overline{x}_i' \approx \frac{\overline{x}_{i+p}-\overline{x}_{i-p}}{2ph} \quad (5)$$

Will give:

$$x_i''=\left(\underline{x}_i'',\overline{x}_i''\right) \quad (6)$$

And:

$$x_i' = \left(\underline{x}_i',\overline{x}_i'\right) \quad (7)$$

$$\underline{x}_i'' = \underline{f(t_i)-j(t_i)x_i'-q(t_i)x_i} \quad (8)$$

$$\overline{x}_i'' = \overline{f(t_i)-j(t_i)x_i'-q(t_i)x_i} \quad (9)$$

Suppose that $j(t_i)>0$ and $q(t_i)>0$ for $I = 0, 1, 2, ..., n$ then;

$$\underline{x}_i''+j(t_i)\underline{x}_i'+q(t_i)\underline{x}_i=f(t_i) \quad (10)$$

And:

$$\overline{x}_i''+j(t_i)\overline{x}_i'+q(t_i)\overline{x}_i=f(t_i) \quad (11)$$

Based on 2.4 and 10 can be reduced to

$$\frac{\underline{x}_{i-p}-2\underline{x}_i+\underline{x}_{i+p}}{(ph)^2} + j(t_i)\frac{\underline{x}_{i+p}-\underline{x}_{i-p}}{2ph}+q(t_i)\underline{x}_i = f(t_i) \quad (12)$$

for $i = 1, 2, ..., n-1$. Meanwhile by substituting Eq. 3 and 5 in to Eq. 11 we have:

$$\frac{\overline{x}_{i-p}-2\overline{x}_i+\overline{x}_{i+p}}{(ph)^2} + j(t_i)\frac{\overline{x}_{i+p}-\overline{x}_{i-p}}{2ph}+q(t_i)\overline{x}_i = f(t_i) \quad (13)$$

Then, Eq. 12 and 13 can be written as follows:

$$\left(2-phj(t_i)\right)\underline{x}_{i-p}+\left(2p^2h^2q(t_i)-4\right)\underline{x}_i +\left(2+phj(t_i)\right)\underline{x}_{i+p}=2p^2h^2\underline{f(t_i)} \quad (14)$$

And:

$$\left(2-phj(t_i)\right)\overline{x}_{i-p}+\left(2p^2h^2q(t_i)-4\right)\overline{x}_i +\left(2+phj(t_i)\right)\overline{x}_{i+p}=2p^2h^2\overline{f(t_i)} \quad (15)$$

respectively for $i = 1, 2, 3,..., n-1$. As the value of p corresponds to 1, 2 and 4, it represents the full, half and quarter-sweep cases respectively. Since both of Eq. 14 have the same form in terms of the equation, except, based on the interval of the α-cuts, the differences identified only in the upper and lower bound, thus it can be rewritten as:

$$\rho_i x_{i-p}+\beta_i x_i+\varphi_i x_{i+p} = F_i \quad (16)$$

Where:

$$\rho_i=2-phj(t_i),\ \beta_i=2p^2h^2q(t_i)-4$$
$$f_i=2+phj(t_i),\ F_i=2p^2h^2f(t_i)$$

From Eq. 16 this approximation equation leads a linear system in matrix form as follows:

$$Ax = b \quad (17)$$

Famlity of succesive over relaxation iterative methods: The SOR method also called as Liebmann method is actually a simple modification from the GS method. In an effort to discuss the formulation of SOR method for linear system Eq.10 let the coefficient matrix A be defined as:

$$A = D+L+U \tag{18}$$

where, D, L and U are diagonal matrix, strictly lower triangular matrix and strictly upper triangular matrix respectively. From Eq. 18 the formulation of SOR iteration family such as FSSOR, HSSOR and QSSOR can be stated as (Young, 1950; Evans and Yousif, 1988):

$$Dx^{(k+1)} = \omega Lx^{(k+1)} + \omega Ux^{(k)} + \omega b + (1-\omega)Dx^{(k)},\ k^3 \geq 0 \tag{19}$$

where, ω and $x^{(k)}$ indicate the relaxation factor and unknown vector at kth iteration, respectively. It is obvious that by taking $\omega = 1$ will result into the GS method. For a general system of the coefficient matrix A, SOR method converges for any relaxation parameters that holds $1 \leq \omega < 2$. Theoretically, optimal value of ω can be given by the equation:

$$\omega_{opt} = \frac{2}{1+\sqrt{1-\lambda^2}}$$

where, λ is the spectral radius of the Jacobi iteration matrix (Evans and Yousif, 1988; Nutanong *et al.*, 2011). Therefore, in solving the linear system (Eq. 10) implementations of QSSOR iterative method may be also described in Algorithm 1.

Algorithm: 1QSSOR schemes:

i) Initializing all the parameters, ω, k = 0 and for, i = p, 2p, 3p, ..., n-p, set:

$$\rho_i = 2\text{-}phj(t_i),\quad \beta_i = 2p^2h^2q(t_i)\text{-}4,$$
$$f_i = 2+phj(t_i),\quad F_i = 2p^2h^2f(t_i), x_i^{(0)} \leftarrow 0$$

ii) For i = p, 2p, 3p, ..., n-p calculate

$$x_i^{(k+1)} \leftarrow (1\text{-}\omega)x_i^{(k)} + \frac{\omega}{\beta_i}\left(F_i\text{-}\rho_i x_{i-1}^{(k+1)}\text{-}f_i x_{i+1}^{(k)}\right)$$

iii) Check the convergence test,. If yes, go to step (vi). Otherwisego back to step (ii)

iv) Display approximate solutions.

RESULTS AND DISCUSSION

Numerical experiments: By using central difference approximation equations, 3 examples of two-point FBVPs are considered. Three parameters were observed such as number of iterations, execution time (in sec) and Hausdorff distance (as mentioned in Definition 1). In the iterative process, Borland Turbo C++ 5.02 software and Dell computer with the processing unit Inter (R) Core (TM) i3 CPU M370 @ 2.40GHz with 2GB of RAM is used in solving generated fuzzy linear equation systems. During implementation of the proposed iterative methods, the value of the tolerance error considered is $\varepsilon = 10^{-10}$.

Definition 1 (Nutanong *et al.*, 2011): Given two minimum bounding rectangles P and Q, a lower bound of the Hausdorff distance from the elements confined by P to the elements confined by Q is defined as:

$$\text{HausDistLB}(P,Q) = \text{Max}\left\{\text{MinDist}(f_a,Q){:}f_a\,\hat{\text{I}}\text{FacesOf}(P)\right\}$$

Example 1:

$$x''(t) = \tilde{k}(-6t)\quad t \in [0,1] \tag{20}$$

where $\tilde{k}[a]=[\underline{k}(a),\bar{k}(a)]=[0.75+0.25a,1.25\text{-}0.25a]$ with boundry conditions $\tilde{x}(0) = 0$ and $\tilde{x}(1) = 1$. The exact solution for:

$$\underline{x''}(t;a) = \underline{k}(a)(-6t) \tag{21}$$

And:

$$\overline{x''}(t;a) = \bar{k}(a)(-6t) \tag{22}$$

Are:

$$\underline{x}(t;a) = \underline{k}(a)\left[-t^3+2t\right] \tag{23}$$

And:

$$\bar{x}(t;a) = \bar{k}(a)\left[-t^3+2t\right] \tag{24}$$

respectively.

Example 2 (Gasilov *et al.*, 2011):

$$x''(t)\text{-}3x'(t)+2x(t) = \tilde{k}(4t\text{-}6),\qquad t \in [0,1] \tag{25}$$

where, $\tilde{k}[a]=[\underline{k}(a),\bar{k}(a)]=[0.75+0.25a,1.25\text{-}0.25a]$ with the boundary conditions $\tilde{x}(0)=2$ and $\tilde{x}(1)=3$. The exact solution for:

$$\underline{x}''(t;a) = \underline{k}(a)(4t\text{-}6) \quad (26)$$

And:

$$\overline{x''}(t;a) = \overline{k}(a)(4t\text{-}6) \quad (27)$$

Are:

$$\underline{x}(t;a) = \underline{k}(a)\left[c_1e^t + c_2e^{2t} + 2t\right] \quad (28)$$

And:

$$\overline{x}(t;a) = \overline{k}(a)\left[c_1e^t + c_2e^{2t} + 2t\right] \quad (29)$$

respectively.

Example 3:

$$x''(t)\text{-}4(t) = \tilde{k}(4\cosh(1)) \quad t \in [0,1]] \quad (30)$$

where, $\tilde{k}[a] = [\underline{k}(a), \overline{k}(a)] = [0.75+0.25a, 1.25\text{-}0.25a]$ with the boundary conditions $\tilde{x}(0) = 0$ and $\tilde{x}(1) = 0$. The exact solution for

$$\underline{x}''(t;a)\text{-}4(t;a) = \underline{k}(a)(4\cosh(1)) \quad (31)$$

And:

$$\overline{x''}(t;a)\text{-}4(t;a) = \overline{k}(a)(4\cosh(1)) \quad (32)$$

Are:

$$\underline{x}(t;a) = \underline{k}(a)\left[\cosh(2t\text{-}1)\text{-}\cosh(1)\right] \quad (33)$$

And:

$$\overline{x}(t;a) = \overline{k}(a)\left[\cosh(2t\text{-}1)\text{-}\cosh(1)\right] \quad (34)$$

respectively. Based on these three examples, all numerical results from the implementation of GS, FSSOR, HSSOR and QSSOR methods are recorded in Table 1-5.

Table 1: Comparison of number of iterations, execution time and Hausdorff distance between GS, FSSOR, HSSOR and QSSOR methods at α = 0.00

		n				
Variables	Methods	256	512	1024	2048	4096
Problem 1						
Number of iterations	GS	188880	681711	2431928	8548735	29480437
	FSSOR	2050	4098	8194	16386	32618
	HSSOR	1026	2050	4098	8194	16386
	QSSOR	514	1026	2050	4098	8194
Execution time	GS	12.36	48.94	211.19	989.91	5719.2
	FSSOR	0.37	0.54	1.05	2.23	6.83
	HSSOR	0.24	0.39	0.56	1.09	2.41
	QSSOR	0.25	0.25	0.41	0.55	1.10
Hausdorff distance	GS	6.6394e-07	2.6560e-06	1.0624e-05	4.2497e-05	1.6999e-04
	FSSOR	5.9142e-10	2.1106e-10	1.5871e-10	6.2329e-10	8.7074e-08
	HSSOR	9.3689e-10	3.7209e-10	2.1106e-10	1.5871e-10	6.9477e-10
	QSSOR	8.1383e-10	5.4467e-10	3.7209e-10	2.1106e-10	4.8286e-10
Problem 2						
Number of iterations	GS	203680	743097	2685480	9593519	33778950
	FSSOR	2050	4083	8141	16200	32185
	HSSOR	1025	2050	4083	8194	16200
	QSSOR	513	1025	2050	4083	8194
Execution time	GS	13.58	55.15	246.78	1288.71	7512.09
	FSSOR	0.41	0.57	1.07	2.43	7.76
	HSSOR	0.25	0.37	0.49	1.15	2.63
	QSSOR	0.23	0.25	0.35	0.60	1.13
Hausdorff distance	GS	1.3060e-05	9.8893e-07	9.8563e-06	4.1321e-05	1.6576e-04
	FSSOR	1.3696e-05	3.4187e-06	8.4439e-07	1.8859e-07	5.3813e-08
	HSSOR	5.4782e-05	1.3696e-05	3.4183e-06	8.5627e-07	1.8859e-07
	QSSOR	2.1916e-04	5.4782e-05	1.3696e-05	3.4187e-06	8.4439e-07
Problem 3						
Number of iterations	GS	131974	475487	1692329	5930853	20369573
	FSSOR	1452	2727	5246	9881	18489
	HSSOR	767	1452	2727	5246	10437
	QSSOR	386	767	1452	2728	5246
Execution time	GS	8.69	35.27	155.77	764.09	4457.31
	FSSOR	0.31	0.45	0.77	1.71	4.56
	HSSOR	0.21	0.29	0.47	0.75	1.77
	QSSOR	0.22	0.22	0.35	0.45	0.78
Hausdorff distance	GS	2.8934e-06	2.4952e-06	7.7115e-06	3.0279e-05	1.2097e-04
	FSSOR	2.4189e-06	6.0075e-07	1.3918e-07	2.1294e-08	6.4788e-08
	HSSOR	9.6837e-06	2.4189e-06	6.0075e-07	1.3918e-07	5.2556e-08
	QSSOR	3.8733e-05	9.6839e-06	2.4189e-06	6.0083e-07	1.3918e-07

Table 2: Comparison of number of iterations, execution time and Hausdorff distance between GS, FSSOR, HSSOR and QSSOR methods at $\alpha = 0.25$

		n				
Variables	Methods	256	512	1024	2048	4096
Problem 1						
Number of iterations	GS	189072	682475	2434982	8560953	29529307
	FSSOR	2050	4098	8194	16386	32651
	HSSOR	1026	2050	4098	8194	16386
	QSSOR	514	1026	2050	4098	8194
Execution time	GS	12.41	49.07	211.36	991.23	5874.81
	FSSOR	0.36	0.52	1.03	2.20	6.52
	HSSOR	0.27	0.40	0.62	1.01	2.42
	QSSOR	0.22	0.27	0.39	0.53	1.07
Hausdorff distance	GS	6.6389e-07	2.6560e-06	1.0624e-05	4.2497e-05	1.6999e-04
	FSSOR	3.5349e-10	2.0052e-10	1.5078e-10	6.6000e-10	8.7858e-08
	HSSOR	8.9003e-10	3.5349e-10	2.0052e-10	1.5078e-10	9.1266e-10
	QSSOR	7.7313e-10	6.1230e-10	3.5349e-10	2.0052e-10	3.8371e-10
Problem 2						
Number of iterations	GS	203867	743841	2688459	9605435	33826616
	FSSOR	2050	4089	8150	16207	32201
	HSSOR	1026	2050	4089	8150	16207
	QSSOR	513	1026	2050	4089	8150
Execution time	GS	13.56	55.4	247.56	1253.19	7534.92
	FSSOR	0.40	0.53	1.03	2.46	7.97
	HSSOR	0.28	0.38	0.58	1.12	2.64
	QSSOR	0.25	0.26	0.40	0.55	1.13
Hausdorff distance	GS	1.2375e-05	8.3434e-07	9.8142e-06	4.1310e-05	1.6575e-04
	FSSOR	1.3011e-05	3.2471e-06	8.0153e-07	1.7794e-07	5.5686e-08
	HSSOR	5.2044e-05	1.3011e-05	3.2471e-06	1.1200e-07	1.7794e-07
	QSSOR	2.0820e-04	5.2044e-05	1.3011e-05	3.2474e-06	8.0153e-07
Problem 3						
Number of iterations	GS	132109	476030	1694502	5939547	20404350
	FSSOR	1615	2719	5241	9871	18451
	HSSOR	770	1450	2719	5241	10439
	QSSOR	386	767	1450	2720	5241
Execution time	GS	8.73	35.26	155.79	756.06	4465.35
	FSSOR	0.35	0.38	0.81	1.59	4.57
	HSSOR	0.13	0.25	0.47	0.81	1.78
	QSSOR	0.21	0.23	0.34	0.43	0.85
Hausdorff distance	GS	2.7724e-06	2.4650e-06	7.7039e-06	3.0277e-05	1.2097e-04
	FSSOR	2.3048e-06	5.7004e-07	1.3149e-07	2.0797e-08	6.5302e-08
	HSSOR	9.2005e-06	2.2981e-06	5.7004e-07	1.3149e-07	5.0280e-08
	QSSOR	3.6797e-05	9.1995e-06	2.2981e-06	5.7012e-07	1.3149e-07

Table 3: Comparison of number of iterations, execution time and Hausdorff distance between GS, FSSOR, HSSOR and QSSOR methods at $\alpha = 0.50$

		n				
Variables	Methods	256	512	1024	2048	4096
Problem 1						
Number of iterations	GS	189205	683007	2437112	8569470	29563373
	FSSOR	2050	4098	8194	16386	32660
	HSSOR	1026	2050	4098	8194	16386
	QSSOR	514	1026	2050	4098	8194
Execution time	GS	12.41	49.25	210.43	988.93	5784.36
	FSSOR	0.34	0.54	0.99	2.23	7.01
	HSSOR	0.26	0.35	0.56	1.03	2.42
	QSSOR	0.21	0.25	0.36	0.52	1.05
Hausdorff distance	GS	6.6390e-07	2.6560e-06	1.0624e-05	4.2497e-05	1.6999e-04
	FSSOR	5.3228e-10	1.8996e-10	1.4284e-10	5.6095e-10	1.0217e-07
	HSSOR	8.4320e-10	3.3488e-10	1.8996e-10	1.4284e-10	8.6462e-10
	QSSOR	7.3244e-10	5.8008e-10	3.3488e-10	1.8996e-10	3.6351e-10
Problem 2						
Number of iterations	GS	203997	744360	2690535	9613742	33859835
	FSSOR	2050	4094	8156	16211	32212
	HSSOR	1026	2050	4098	8156	16211
	QSSOR	514	1026	2050	4094	8156
Execution time	GS	13.48	55.37	248.32	1225.09	7472.23
	FSSOR	0.40	0.58	1.04	2.46	7.89
	HSSOR	0.24	0.39	0.59	1.09	2.63
	QSSOR	0.21	0.28	0.40	0.55	1.10

Table 3: Continue

Variables	Methods	n 256	512	1024	2048	4096
Hausdorff distance	GS	1.1690e-05	6.8428e-07	9.7722e-06	4.1299e-05	1.6575e-04
	FSSOR	1.2326e-05	3.0757e-06	7.5715e-07	1.6716e-07	5.6802e-08
	HSSOR	4.9305e-05	1.2326e-05	3.0817e-06	7.5827e-07	1.6716e-07
	QSSOR	1.9724e-04	4.9305e-05	1.2326e-05	3.0757e-06	7.5827e-07
Problem 3						
Number of iterations	GS	132204	476410	1696018	5945607	20428592
	FSSOR	1615	2712	5317	9856	18411
	HSSOR	770	1446	2712	5317	10438
	QSSOR	386	770	1446	2712	5317
Execution time	GS	8.73	35.4	155.8	757.38	4585.51
	FSSOR	0.37	0.44	0.79	1.57	4.50
	HSSOR	0.25	0.27	0.46	0.84	1.82
	QSSOR	0.21	0.22	0.31	0.50	0.81
Hausdorff distance	GS	2.6513e-06	2.4346e-06	7.6963e-06	3.0275e-05	1.2097e-04
	FSSOR	2.1837e-06	5.3942e-07	1.2719e-07	2.0332e-08	6.6115e-08
	HSSOR	8.7161e-06	2.1773e-06	5.3942e-07	1.2719e-07	4.8090e-08
	QSSOR	3.4860e-05	8.7161e-06	2.1773e-06	5.3942e-07	1.2719e-07

Table 4: Comparison of number of iterations, execution time and Hausdorff distance between GS, FSSOR, HSSOR and QSSOR methods at $\alpha = 0.75$

Variables	Methods	n 256	512	1024	2048	4096
Problem 1						
Number of iterations	GS	189283	683321	2438369	8574499	29583490
	FSSOR	2050	4098	8194	16386	32690
	HSSOR	1026	2050	4098	8194	16386
	QSSOR	514	1026	2050	4098	8194
Execution time	GS	12.26	49.22	210.33	1026.58	5771.53
	FSSOR	0.38	0.53	1.04	2.22	6.62
	HSSOR	0.25	0.36	0.53	1.07	2.41
	QSSOR	0.21	0.24	0.37	0.58	1.10
Hausdorff distance	GS	6.6392e-07	2.6560e-06	1.0624e-05	4.2497e-05	1.6999e-04
	FSSOR	3.1628e-10	3.8917e-10	1.3490e-10	5.2979e-10	8.7221e-08
	HSSOR	7.9635e-10	3.1628e-10	1.7941e-10	1.3490e-10	8.1659e-10
	QSSOR	6.9175e-10	6.2365e-10	3.1628e-10	3.8917e-10	2.7234e-10
Problem 2						
Number of iterations	GS	204073	744667	2691762	9618646	33879454
	FSSOR	2050	4098	8155	16213	32217
	HSSOR	1026	2050	4098	8155	16213
	QSSOR	514	1026	2050	4098	8155
Execution time	GS	13.70	55.43	247.80	1227.29	7448.72
	FSSOR	0.35	0.54	1.01	2.45	7.71
	HSSOR	0.30	0.30	0.59	1.11	2.61
	QSSOR	0.21	0.27	0.39	0.60	1.17
Hausdorff distance	GS	1.1006e-05	5.4041e-07	9.7301e-06	4.1289e-05	1.6575e-04
	FSSOR	1.1641e-05	2.9105e-06	7.1527e-07	1.5652e-07	5.8554e-08
	HSSOR	4.6566e-05	1.1641e-05	2.9105e-06	7.1527e-07	1.5652e-07
	QSSOR	1.8629e-04	4.6566e-05	1.1641e-05	2.9105e-06	7.1527e-07
Problem 3						
Number of iterations	GS	132260	476633	1696912	5949186	20442908
	FSSOR	1614	2709	5227	9838	18366
	HSSOR	770	1443	2709	5227	10435
	QSSOR	386	770	1443	2709	5227
Execution time	GS	8.72	35.42	155.72	757.27	4364.75
	FSSOR	0.33	0.43	0.74	1.56	4.45
	HSSOR	0.23	0.35	0.46	0.80	1.79
	QSSOR	0.22	0.24	0.31	0.44	0.77
Hausdorff distance	GS	2.5302e-06	2.4044e-06	7.6888e-06	3.0273e-05	1.2097e-04
	FSSOR	2.0628e-06	5.0902e-07	1.1616e-07	1.9888e-08	6.6999e-08
	HSSOR	8.2317e-06	2.0564e-06	5.0902e-07	1.1616e-07	4.6251e-08
	QSSOR	3.2923e-05	8.2319e-06	2.0564e-06	5.0902e-07	1.1616e-07

Table 5: Comparison of number of iterations, execution time and Hausdorff distance between GS, FSSOR, HSSOR and QSSOR methods at $\alpha = 1.00$

		n				
Variables	Methods	256	512	1024	2048	4096
Problem 1						
Number of iterations	GS	189310	683426	2438784	8576162	29590144
	FSSOR	2050	4098	8194	16386	32684
	HSSOR	1026	2050	4098	8194	16386
	QSSOR	514	1026	2050	4098	8194
Execution time	GS	13.84	49.45	210.66	809.53	5758.67
	FSSOR	0.34	0.56	1.05	2.17	6.71
	HSSOR	0.28	0.35	0.59	1.06	2.32
	QSSOR	0.22	0.27	0.41	0.58	1.05
Hausdorff distance	GS	6.6386e-07	2.6559e-06	1.0624e-05	4.2497e-05	1.6999e-04
	FSSOR	4.7314e-10	3.6627e-10	1.2698e-10	4.9867e-10	8.7297e-08
	HSSOR	7.4951e-10	2.9767e-10	1.6885e-10	1.2698e-10	7.6857e-10
	QSSOR	6.5106e-10	5.8697e-10	2.9767e-10	3.6627e-10	2.5631e-10
Problem 2						
Number of iterations	GS	204098	744768	2692166	9620268	33885946
	FSSOR	2050	4098	8154	16212	32218
	HSSOR	1026	2050	4096	8154	16212
	QSSOR	514	1026	2050	4096	8154
Execution time	GS	13.54	55.4	247.5	1230.84	7567.65
	FSSOR	0.36	0.55	1.12	2.45	7.62
	HSSOR	0.27	0.39	0.58	1.08	2.58
	QSSOR	0.25	0.25	0.31	0.63	1.10
Hausdorff distance	GS	1.0321e-05	4.0479e-07	9.6881e-06	4.1279e-05	1.6575e-04
	FSSOR	1.0957e-05	2.7393e-06	6.7247e-07	1.4583e-07	5.9742e-08
	HSSOR	4.3827e-05	1.0957e-05	2.7315e-06	6.7245e-07	1.4583e-07
	QSSOR	1.7533e-04	4.3827e-05	1.0957e-05	2.7314e-06	6.7247e-07
Problem 3						
Number of iterations	GS	132278	476706	1697208	5950370	20447642
	FSSOR	1612	2706	5218	9816	18316
	HSSOR	770	1440	2706	5218	10428
	QSSOR	386	770	1440	2706	5218
Execution time	GS	8.78	35.43	155.72	755.2	4615.31
	FSSOR	0.35	0.43	0.79	1.60	4.51
	HSSOR	0.15	0.34	0.45	0.80	1.82
	QSSOR	0.22	0.23	0.32	0.43	0.81
Hausdorff distance	GS	2.4092e-06	2.3742e-06	7.6812e-06	3.0271e-05	1.2097e-04
	FSSOR	1.9418e-06	4.7870e-07	1.0853e-07	1.9437e-08	6.7972e-08
	HSSOR	7.7475e-06	1.9355e-06	4.7869e-07	1.0852e-07	4.4433e-08
	QSSOR	3.0987e-05	7.7476e-06	1.9355e-06	4.7870e-07	1.0852e-07

CONCLUSION

This study discusses the application of family of SOR iterative methods to solve linear system arised from the discretization of the two-point FBVPs via second-order finite difference scheme. The results show that QSSOR method is more superior in terms of the number of iterations, execution time and Hausdorff distance as compared to the GS, FSSOR and HSSOR methods. This is due to the computational complexity of the high-order discretization schemes. Since the family of SOR method is categorized as the family of one step iterative methods, therefore, further study of this problem will be continued to consider the family of two-step iterative methods in order to accelerate the convergence rate in solving linear systems. It is hoped that the capability of the proposed method will be helpful for the further investigation in solving any multi-dimensional fuzzy partial differential equations (Farajzadeh *et al*., 2010). Also, the family of AGE methods such as Modified Alternating Group Explicit (MAGE) (Evans and Yousif, 1988; Yousif and Evans, 1987) and Two Parameter Alternating Group Explicit (TAGE) (Mohanty *et al*., 2003; Sukon, 1996; Dahalan *et al*., 2015b) methods can be used as linear solvers in solving the same problem.

ACKNOWLEDGEMENT

This research works was funded by research grant, RAGS/2013/UPNM/SG/04/3, National Defence University of Malaysia.

REFERENCES

Abdullah, A.R., 1991. The four point Explicit Decoupled Group (EDG) method: A fast poisson solver. Int. J. Comput. Math., 38: 61-70.

Akhir, M.K.M., M. Othman, J. Sulaiman, Z.A. Majid and M. Suleiman, 2012. Half-sweep iterative method for solving two-dimensional helmholtz equations. Intl. J. Appl. Math. Stat., 29: 101-109.

Allahviranloo, T., 2002. Difference methods for fuzzy partial differential equations. Comput. Methods Applied Math., 2: 233-242.

Dahalan, A.A. and J. Sulaiman, 2015a. Implementation of TAGE method using seikkala derivatives applied to two-point fuzzy boundary value problems. Intl. J. Differ. Equ., 2015: 1-7.

Dahalan A.A. and J. Sulaiman, 2015b. Approximate solution for 2 dimensional fuzzy parabolic equations in QSAGE iterative method. Intl. J. Math. Anal., 9: 1733-1746.

Dahalan, A.A., J. Sulaiman and M.S. Muthuvalu, 2014. Performance of HSAGE method with Seikkala derivative for 2-D fuzzy poisson equation. Applied Math. Sci., 8: 885-899.

Dahalan, A.A., M.S. Muthuvalu and J. Sulaiman, 2013. Numerical solutions of two-point fuzzy boundary value problem using half-sweep alternating group explicit method. AIP Conf. Proc., 1557: 103-107.

Dahalan, A.A., M.S. Muthuvalu and J. Sulaiman, 2015a. Half-sweep successive over relaxation iterative method to solve two-point fuzzy boundary value problems. Proceedings of the 22nd National Symposium On Mathematical Sciences (SKSM22): Strengthening Research and Collaboration of Mathematical Sciences in Malaysia, November 24-26, 2015, AIP Publishing, Selangor, Malaysia, pp: 1-1.

Dahalan, A.A., M.S. Muthuvalu and J. Sulaiman, 2015b. Numerical solution for fuzzy heat equation using HSAGE method. UPB. Sci. Bull. Ser. A., 7: 251-260.

Evans, D.J. and W.S. Yousif, 1988. The Modified Alternating Group Explicit (MAGE) method. Applied Math. Model., 12: 262-267.

Farajzadeh, A., A.H. Pour and M. Amini, 2010. An explicit method for solving fuzzy partial differential equation. Int. Math. Forum, 5: 1025-1036.

Gasilov, N., S.E. Amrahov and A.G. Fatullayev, 2011. Linear differential equations with fuzzy boundary values. Proceedings of the 2011 5th International Conference on Application of Information and Communication Technologies (AICT), October 12-14, 2011, IEEE, Ankara, Turkey, ISBN:978-1-61284-831-0, pp: 1-5.

Ibrahim, A. and A.R. Abdullah, 1995. Solving The two dimensional diffusion equation by the four point Explicit Decoupled Group (EDG) iterative method. Int. J. Comput. Mathe., 58: 253-256.

Koh, W.S., J. Sulaiman and R. Mail, 2010. Quarter-sweep improving modified gauss-seidel method for pricing European option. Math., 26: 179-185.

Mohanty, R.K., P.L. Sachdev and N. Jha, 2003. TAGE method for nonlinear singular two point boundary value problem using a fourth order difference scheme. Neural Parallel Sci. Comput., 11: 281-296.

Muthuvalu, M.S. and J. Sulaiman, 2008. Half-sweep geometric mean method for solution of linear fredholm equations. Matematika, 24: 75-84.

Muthuvalu, M.S. and J. Sulaiman, 2011. Numerical solution of second kind linear fredholm integral equations using QSGS iterative method with high-order newton-cotes quadrature schemes. Malaysian J. Math. Sci., 5: 85-100.

Nutanong, S., E.H. Jacox and H. Samet, 2011. An incremental Hausdorff distance calculation algorithm. Proc. VLDB Endowment, 4: 506-517.

Othman, M. and A.R. Abdullah, 2000. An efficient four points modified explicit group poisson solver. Int. J. Comput. Math., 76: 203-217.

Sukon, K.S., 1996. On Two parameter Alternating Group Explicit (TAGE) method for singular perturbation problems. Parallel Algorith. Applic., 10: 71-77.

Sulaiman, J., M. Othman and M.K. Hasan, 2009. A new quarter sweep arithmetic mean (QSAM) method to solve diffusion equations. A new quarter sweep arithmetic mean (QSAM) method to solve diffusion equations. 1: 93-103.

Sulaiman, J., M.K. Hasan and M. Othman, 2004. The Half-Sweep Iterative Alternating Decomposition Explicit (HSIADE) method for diffusion equation. Proceedings of the 1st International Symposium on Computational and Information Science, December 16-18, 2004, Shanghai, China, pp: 57-63.

Young, D.M., 1950. Iterative methods for solving partial difference equations of elliptic type. Ph.D. Thesis, Harvard University, Cambridge, UK.

Yousif, W.S. and D.J. Evans, 1987. The Modified Alternating Group Explicit (MAGE) algorithm for solving tridiagonal linear equations. Comput. Math. Applic., 14: 169-172