NUMERICAL SOLUTION FOR FUZZY HEAT EQUATION USING HSAGE METHOD

A'qilah Ahmad DAHALAN[1], Mohana Sundaram MUTHUVALU[2], Jumat SULAIMAN[3]

In this paper, application of the Half-Sweep Alternating Group Explicit (HSAGE) method to solve finite difference approximation equations arising from fuzzy heat equation is examined. The formulation and implementation of HSAGE method are also presented. In addition, numerical results by solving two test problems are included and compared with the standard Gauss-Seidel (GS) and Alternating Group Explicit (AGE) methods.

Keywords: Two-stage iteration, Implicit scheme, Fuzzy heat equation

1. Introduction

The Alternating Group Explicit (AGE) method is one of the widely used and successful two-stage iterative methods to solve sparse linear system. The AGE method employs the fractional splitting strategy which is applied alternately at each intermediate step on linear system. In a series of papers, the effectiveness of the AGE and its variants methods were studied and tested by solving a variety of scientific problems, for instance refer [1, 2, 3, 4]. Besides that, the concept of half-sweep iteration has been initiated by Abdullah [5] via the Explicit Decoupled Group (EDG) method for solving two-dimensional Poisson equations. The basic idea of the half-sweep iteration approach is to speed-up the computational time by reducing the computational complexity of the solution method.

Consequently, in this paper, performance of the half-sweep iteration with AGE method i.e. Half-Sweep Alternating Group Explicit (HSAGE) method will be investigated for solving linear systems generated from the fuzzy heat equation. The performance of HSAGE method will be compared with the existing standard Gauss-Seidel (GS) and AGE methods. The standard GS and AGE methods are also known as Full-Sweep Gauss-Seidel (FSGS) and Full-Sweep Alternating Group Explicit (FSAGE) methods respectively.

The remainder of this paper is organized in following way. In Section 2, derivation of the full- and half-sweep finite difference approximation equations will be elaborated. The latter section of this paper will discuss the implementation

[1] School of Science and Technology, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia, e-mail: aqilahAD@yahoo.com
[2] Department of Fundamental and Applied Sciences, Faculty of Science and Information Technology, Universiti Teknologi PETRONAS, Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia, e-mail: msmuthuvalu@gmail.com
[3] School of Science and Technology, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia, e-mail: jumat@ums.edu.my

of FSAGE and HSAGE methods for solving generated linear systems. Some numerical results will be presented in Section 4 to assert performance of the tested methods and concluding remarks are given in Section 5.

2. Full- and Half-Sweep BTCS Approximation Equations

We start this section with the notation that used in the paper. The tilde sign over a letter denote a fuzzy subset of real numbers. For a fuzzy subset of the real numbers, $\tilde{U}$, it is characterized by the membership function evaluated at x, written as $\tilde{U}(x)$ as a number in $[0,1]$. An α-cut of $\tilde{U}$, written $\tilde{U}(\alpha)$ is defined as $\{x \mid \tilde{U}(x) \geq \alpha\}$ for $0 < \alpha \leq 1$. The intervals of α-cut can be written as $\tilde{U}(\alpha) = [\underline{U}(\alpha), \overline{U}(\alpha)]$, for all α because they are always closed and bounded [6].

In this section, we attempt to construct the full- and half-sweep finite difference approximation equations for fuzzy heat equation. For further discussions on formulating the full- and half-sweep finite difference approximation equations, consider the interval that is divided uniformly as shown in Figs. 1 and 2.

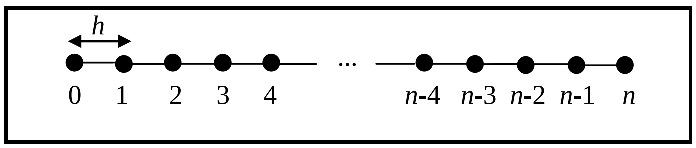

Fig. 1. Distribution of uniform node points for the full-sweep case

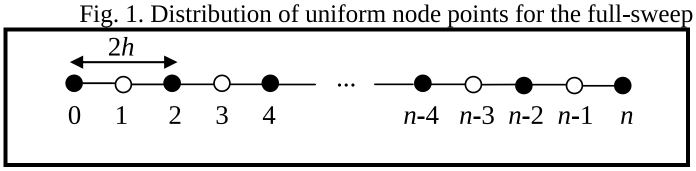

Fig. 2. Distribution of uniform node points for the half-sweep case

Based on Figs. 1 and 2, the full- and half-sweep iterative methods will compute approximate values onto node points of type ● only until the convergence criterion is satisfied. After the convergence criterion is achieved, the approximation solutions for the remaining points are computed directly [5]. From this section onwards, the values of p, which corresponds to one and two represents the full- and half-sweep cases respectively.

Now, let consider the following fuzzy heat equation

$$\frac{\partial \tilde{U}(x,t)}{\partial t} - \lambda \frac{\partial^2 \tilde{U}(x,t)}{\partial x^2} = 0, \quad 0 < x < l, \quad t > 0, \tag{1}$$

where $\tilde{F}\left(x,t,\tilde{K}\right) = 0$ subject to the boundary and initial conditions

$$\tilde{U}(0,t)=\tilde{U}(l,t)=0, \quad t>0$$
$$\tilde{U}(x,0)=\tilde{f}(x), \quad 0\le x\le l \; .$$

Let divide the domain $[0,1]\times[0,T]$ into $n\times m$ mesh with spatial step size $h=\frac{1}{n}$ in x-direction and $k=\frac{T}{m}$ in t-direction. The discrete set of points of x and t are given by $x_i = ih$ $(i=0,1,2,\cdots,n)$ and $t_j = jk$ $(j=0,1,2,\cdots,m)$ respectively. Denote the value of $\tilde{U}$ at the representative discrete points by $\tilde{U}(x_i,t_j)=\tilde{U}_{i,j}$ and parametric form of fuzzy number $\tilde{U}_{i,j}$ as $\tilde{U}_{i,j}=\left(\underline{U_{i,j}},\overline{U_{i,j}}\right)$.

In this paper, we derive the formulation of full- and half-sweep finite difference approximation equations based on the implicit scheme i.e. Backward Time, Centered Space (BTCS). By using BTCS scheme,

$$\left.\begin{aligned}\frac{\partial \underline{U}}{\partial t} &\approx \frac{\underline{U_{i,j+1}}-\underline{U_{i,j}}}{k}\\ \frac{\partial \overline{U}}{\partial t} &\approx \frac{\overline{U_{i,j+1}}-\overline{U_{i,j}}}{k}\end{aligned}\right\} \tag{2}$$

and

$$\left.\begin{aligned}\frac{\partial^2 \underline{U}}{\partial x^2} &\approx \lambda\left[\frac{\underline{U_{i-p,j+1}}-2\underline{U_{i,j+1}}+\underline{U_{i+p,j+1}}}{(ph)^2}\right]\\ \frac{\partial^2 \overline{U}}{\partial x^2} &\approx \lambda\left[\frac{\overline{U_{i-p,j+1}}-2\overline{U_{i,j+1}}+\overline{U_{i+p,j+1}}}{(ph)^2}\right]\end{aligned}\right\}. \tag{3}$$

By subsituting (2) and (3) into problem (1), the generalized full- and half-sweep BTCS approximation equations for problem (1) can be represented as follows

$$\left.\begin{aligned}-\frac{k\lambda}{(ph)^2}\underline{U_{i-p,j+1}}+\left(\frac{2k\lambda}{(ph)^2}+1\right)\underline{U_{i,j+1}}-\frac{k\lambda}{(ph)^2}\underline{U_{i+p,j+1}}=\underline{U_{i,j}}\\ -\frac{k\lambda}{(ph)^2}\overline{U_{i-p,j+1}}+\left(\frac{2k\lambda}{(ph)^2}+1\right)\overline{U_{i,j+1}}-\frac{k\lambda}{(ph)^2}\overline{U_{i+p,j+1}}=\overline{U_{i,j}}\end{aligned}\right\} \tag{4}$$

for $i=p,2p,\ldots,n-2p,n-p$ and $j=1,2,\ldots,,m$.

Based on the Eq. (4), the only difference between these equations are on the interval of the α-cuts which are upper and lower bounds. Therefore, for simplicity, approximation Eq. (4) can be written as

$$-\frac{k\lambda}{(ph)^2}U_{i-p,j+1}+\left(\frac{2k\lambda}{(ph)^2}+1\right)U_{i,j+1}-\frac{k\lambda}{(ph)^2}U_{i+p,j+1}=U_{i,j} \tag{5}$$

for $i=p,2p,\ldots,n-2p,n-p$ and $j=1,2,\ldots,,m$. The full- and half-sweep BTCS approximation equations for $\underline{U}$ and $\overline{U}$ as shown in Eq. (5) can be represented in matrix form as

$$AU=b \tag{6}$$

where A is the tridiagonal matrix

$$A=\begin{bmatrix} 2\beta+1 & -\beta & & & & \\ -\beta & 2\beta+1 & -\beta & & 0 & \\ & -\beta & 2\beta+1 & -\beta & & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & -\beta & 2\beta+1 & -\beta \\ & & & & -\beta & 2\beta+1 \end{bmatrix}_{\left(\frac{n}{p}-1\right)\times\left(\frac{n}{p}-1\right)}$$

with $\beta=\dfrac{k\lambda}{(ph)^2}$. Implementation of the BTCS scheme requires to solve a linear system at each time step and it is unconditional stable.

3. AGE Iterative methods

In the section, an implementation of the FSAGE and HSAGE methods for solving corresponding full- and half-sweep BTCS approximation equations will be discussed. Now, let matrix A be decomposed into sum of two matrices, as follows

$$A=G_1+G_2 \tag{7}$$

where

$$G_1=\left[\begin{array}{cc|cc|c|cc} \gamma & -\beta & & & & & \\ -\beta & \gamma & & & & & \\ \hline & & \gamma & -\beta & & & \\ & & -\beta & \gamma & & & \\ \hline & & & & \ddots & & \\ \hline & & & & & \gamma & -\beta \\ & & & & & -\beta & \gamma \end{array}\right]$$

and

$$G_2=\left[\begin{array}{c|cc|c|cc|c} \gamma & & & & & & \\ \hline & \gamma & -\beta & & & & \\ & -\beta & \gamma & & & & \\ \hline & & & \ddots & & & \\ \hline & & & & \gamma & -\beta & \\ & & & & -\beta & \gamma & \\ \hline & & & & & & \gamma \end{array}\right]$$

if n is odd. Similarly, we define the following matrices

$$G_1 = \begin{bmatrix} \gamma & -\beta & & & & & \\ -\beta & \gamma & & & & & \\ & & \ddots & & & & \\ & & & \gamma & -\beta & & \\ & & & -\beta & \gamma & & \\ & & & & & & \gamma \end{bmatrix} \quad \text{and} \quad G_2 = \begin{bmatrix} \gamma & & & & & \\ & \gamma & -\beta & & & \\ & -\beta & \gamma & & & \\ & & & \ddots & & \\ & & & & \gamma & -\beta \\ & & & & -\beta & \gamma \end{bmatrix}$$

if n is even, with $\gamma = \frac{2\beta + 1}{2}$. Based on the splitting of A, G_1 and G_2 satisfy the following conditions [7]

a) $(rI + G_1)$ and $(rI + G_2)$ are non-singular for any $r > 0$ (r is called the acceleration parameter)

b) it is practical to solve the systems $(rI + G_1)y = c$ and $(rI + G_2)z = d$

for any vectors c and d and, for any $r > 0$ in explicit form since they consist of only the (2×2) subsystems.

By using splitting (7), linear system (6) becomes

$$(G_1 + G_2)U = b \tag{8}$$

and the general formulation of FSAGE and HSAGE methods to compute $U^{(k+1)}$ is as follows

$$\left.\begin{aligned} (rI + G_1)U^{\left(k+\frac{1}{2}\right)} &= b + (rI - G_2)U^{(k)} \\ (rI + G_2)U^{(k+1)} &= b + (rI - G_1)U^{\left(k+\frac{1}{2}\right)} \end{aligned}\right\}. \tag{9}$$

Since $(rI + G_1)$ and $(rI + G_2)$ are non-singular, then their respective inverses exist. Thus, the formulation of FSAGE and HSAGE methods can be rewritten in explicit form as

$$\left.\begin{aligned} U^{\left(k+\frac{1}{2}\right)} &= (rI + G_1)^{-1}\left[b + (rI - G_2)U^{(k)}\right] \\ U^{(k+1)} &= (rI + G_2)^{-1}\left[b + (rI - G_1)U^{\left(k+\frac{1}{2}\right)}\right] \end{aligned}\right\}. \tag{10}$$

The rate of convergence of FSAGE and HSAGE methods is governed by the acceleration parameter, r. From (10), the iteration matrices for FSAGE and HSAGE methods are

$$T_{FSAGE} = T_{HSAGE} = (rI + G_2)^{-1}(rI - G_1)(rI + G_1)^{-1}(rI - G_2) \tag{11}$$

and satisfy the following Theorem 1.

Theorem 1.
If G_1 and G_2 are real positive definite matrices and $r > 0$, then $\rho(T_{FSAGE}) < 1$ and $\rho(T_{HSAGE}) < 1$.

Proof. The proof runs parallel to a standard proof given in [7]. □

Based on (10), the implementation of FSAGE and HSAGE methods to solve corresponding full- and half-sweep BTCS approximation equations is presented in Algorithm 1.

Algorithm 1. FSAGE and HSAGE methods

i. Initialize all the parameters
ii. Iteration cycle
   for $k = 0,1,2,\cdots$
   a. Stage 1
      Compute
      $$U^{\left(k+\frac{1}{2}\right)} \leftarrow (rI + G_1)^{-1}\left[b + (rI - G_2)U^{(k)}\right]$$
   b. Stage 2
      Compute
      $$U^{(k+1)} \leftarrow (rI + G_2)^{-1}\left[b + (rI - G_1)U^{\left(k+\frac{1}{2}\right)}\right]$$
iii. Convergence test. If the convergence criterion i.e. $\left\|U^{(k+1)} - U^{(k)}\right\|_\infty \le \varepsilon$ is satisfied, go to Step (iv). Otherwise go back to Step (ii).
iv. Stop. Display approximate solutions.

4. Numerical results

In order to compare the performance of the FSAGE and HSAGE methods, the following fuzzy heat equations were used as the test problems.

Test Problem 1 [8]

Consider the fuzzy heat equation

$$\frac{\partial \tilde{U}}{\partial t}(x,t) = 4\frac{\partial^2 \tilde{U}}{\partial x^2}(x,t), \quad 0 < x < 1, t > 0 \tag{12}$$

with the boundary conditions $\tilde{U}(0,t) = \tilde{U}(1,t) = 0, \quad t > 0$

and $\tilde{U}(x,0) = \tilde{f}(x) = \frac{2}{\pi}\tilde{k}\sin \pi x$ and, $\tilde{k}[\alpha] = \left[\underline{k}(\alpha), \overline{k}(\alpha)\right] = [0.5\alpha + 0.5,\ 1.5 - 0.5\alpha]$.

The exact solution for $\frac{\partial \underline{U}}{\partial t}(x,t;\alpha) = 4\frac{\partial^2 \underline{U}}{\partial x^2}(x,t;\alpha)$

and $\frac{\partial \overline{U}}{\partial t}(x,t;a) = 4\frac{\partial^2 \overline{U}}{\partial x^2}(x,t;a)$ are $\underline{U}(x,t;\alpha) = \frac{2}{\pi}\underline{k}(\alpha)e^{-4\pi^2 t}\sin \pi x$

and $\overline{U}(x,t;\alpha) = \frac{2}{\pi}\overline{k}(\alpha)e^{-4\pi^2 t}\sin \pi x$ respectively.

Test Problem 2 [6]

Consider the fuzzy heat equation

$$\frac{\partial \tilde{U}}{\partial t}(x,t) = \frac{\partial^2 \tilde{U}}{\partial x^2}(x,t), \quad 0 < x < 1, t > 0 \tag{13}$$

subject to the conditions $\tilde{U}(0,t) = \tilde{U}(l,t) = 0, \quad t > 0$

and $\tilde{U}(x,0) = \tilde{f}(x) = \tilde{k}\sin \pi x, \quad 0 \le x \le 1$

and, $\tilde{k}[\alpha] = [\underline{k}(\alpha), \overline{k}(\alpha)] = [0.5\alpha + 0.5, 1.5 - 0.5\alpha]$.

The exact solution for $\frac{\partial \underline{U}}{\partial t}(x,t;\alpha) = \frac{\partial^2 \underline{U}}{\partial x^2}(x,t;\alpha)$

and $\frac{\partial \overline{U}}{\partial t}(x,t;a) = \frac{\partial^2 \overline{U}}{\partial x^2}(x,t;a)$ are $\underline{U}(x,t;\alpha) = \underline{k}(\alpha)e^{-\pi^2 t}\sin \pi x$

and $\overline{U}(x,t;\alpha) = \overline{k}(\alpha)e^{-\pi^2 t}\sin \pi x$ respectively.

For numerical results, there parameters i.e. number of iterations, execution time (in seconds) and Hausdorff distance [6] were measured and considered for comparative analysis. The value of initial datum, $U^{(0)}$ is set to be zero for all the test problems. The computations are performed on a personal computer with Intel(R) Core(TM) i3 CPU M370 and 2GB RAM and, the programs were compiled by using C++ language. Throughout the numerical experiments, the convergence test considered $\varepsilon = 10^{-10}$ and carried out on several different values of n. All results of numerical simulations obtained from implementation of the FSGS, FSAGE and HSAGE methods for test problems 1 and 2 have been tabulated in Tables 1 to 5. Meanwhile, Table 6 described the percentage gains in terms of number of iterations and execution time for FSAGE and HSAGE methods compared to FSGS method for both test problems.

Table 1: Numerical results of FSGS, FSAGE and HSAGE methods at $\alpha = 0.00$

		Methods	n			
			512	1024	2048	4096
Test Problem 1	Number of iterations	FSGS	36309	128720	448803	1528924
		FSAGE	5199	18926	67947	239878
		HSAGE	1417	5199	18926	67947
	Execution time	FSGS	58.95	418.21	2917.95	19884.31
		FSAGE	9.80	71.25	511.46	3621.10
		HSAGE	1.35	9.73	71.02	516.65

	Hausdorff Distance	FSGS	4.6781e-04	4.5983e-04	4.2804e-04	3.0095e-04
		FSAGE	4.7013e-04	4.6910e-04	4.6513e-04	4.4929e-04
		HSAGE	4.7053e-04	4.7013e-04	4.6910e-04	4.6513e-04
Test Problem 2	Number of iterations	FSGS	33287	45875	166259	596225
		FSAGE	1749	6462	23816	87009
		HSAGE	484	1749	6462	23816
	Execution time	FSGS	54.03	149.03	1081.76	7751.73
		FSAGE	3.33	24.41	180.16	1310.31
		HSAGE	0.49	3.28	24.11	179.95
	Hausdorff Distance	FSGS	1.3474e-03	3.7161e-03	3.6913e-03	3.5930e-03
		FSAGE	3.7252e-03	3.7232e-03	3.7198e-03	3.7071e-03
		HSAGE	3.7301e-03	3.7252e-03	3.7232e-03	3.7198e-03

Table 2: Numerical results of FSGS, FSAGE and HSAGE methods at $\alpha = 0.25$

		Methods	n			
			512	1024	2048	4096
Test Problem 1	Number of iterations	FSGS	36511	129530	452066	1542351
		FSAGE	5224	19026	68348	241494
		HSAGE	1423	5224	19026	68348
	Execution time	FSGS	59.25	420.54	2938.85	20122.23
		FSAGE	9.86	71.60	515.27	3644.49
		HSAGE	1.36	9.78	71.11	516.23
	Hausdorff Distance	FSGS	4.2860e-04	4.2062e-04	3.8884e-04	2.6175e-04
		FSAGE	4.3092e-04	4.2990e-04	4.2593e-04	4.1009e-04
		HSAGE	4.3132e-04	4.3092e-04	4.2990e-04	4.2593e-04
Test Problem 2	Number of iterations	FSGS	12609	46086	167104	599603
		FSAGE	1756	6488	23920	87430
		HSAGE	486	1756	6488	23920
	Execution time	FSGS	20.50	149.72	1086.73	7812.72
		FSAGE	3.33	24.54	179.81	1315.17
		HSAGE	0.48	3.30	24.22	181.11
	Hausdorff Distance	FSGS	3.4130e-03	3.4057e-03	3.3809e-03	3.2827e-03
		FSAGE	3.4147e-03	3.4129e-03	3.4095e-03	3.3967e-03
		HSAGE	3.4193e-03	3.4147e-03	3.4129e-03	3.4095e-03

Table 3: Numerical results of FSGS, FSAGE and HSAGE methods at $\alpha = 0.50$

		Methods	n			
			512	1024	2048	4096
Test Problem 1	Number of iterations	FSGS	36640	130048	454150	1550896
		FSAGE	5240	19090	68605	242527
		HSAGE	1427	5240	19090	68605
	Execution time	FSGS	59.46	422.29	2953.75	20232.25
		FSAGE	9.89	71.87	516.64	3662.89
		HSAGE	1.37	9.83	71.20	517.90
	Hausdorff Distance	FSGS	3.8940e-04	3.8142e-04	3.4964e-04	2.2255e-04
		FSAGE	3.9172e-04	3.9070e-04	3.8673e-04	3.7089e-04
		HSAGE	3.9210e-04	3.9172e-04	3.9070e-04	3.8673e-04
Test Problem 2	Number of iterations	FSGS	12643	46221	167643	601762
		FSAGE	1760	6505	23987	87698
		HSAGE	487	1760	6505	23987

	Execution time	FSGS	20.63	150.29	1089.26	7840.89
		FSAGE	3.33	24.48	180.48	1319.19
		HSAGE	0.47	3.32	24.28	180.99
	Hausdorff Distance	FSGS	3.1025e-03	3.0954e-03	3.0706e-03	2.9724e-03
		FSAGE	3.1043e-03	3.1025e-03	3.0991e-03	3.0864e-03
		HSAGE	3.1084e-03	3.1043e-03	3.1025e-03	3.0991e-03

Table 4: Numerical results of FSGS, FSAGE and HSAGE methods at $\alpha = 0.75$

		Methods	*n*			
			512	1024	2048	4096
Test Problem 1	Number of iterations	FSGS	36713	130338	455318	1555677
		FSAGE	5248	19125	68749	243106
		HSAGE	1429	5248	19125	68749
	Execution time	FSGS	59.59	423.29	3007.30	20236.45
		FSAGE	9.91	71.98	517.61	3673.02
		HSAGE	1.37	9.85	71.36	519.77
	Hausdorff Distance	FSGS	3.5020e-04	3.4222e-04	3.1044e-04	1.8335e-04
		FSAGE	3.5251e-04	3.5150e-04	3.4753e-04	3.3169e-04
		HSAGE	3.5288e-04	3.5251e-04	3.5150e-04	3.4753e-04
Test Problem 2	Number of iterations	FSGS	12662	46296	167946	602972
		FSAGE	1762	6514	24025	87849
		HSAGE	487	1762	6514	24025
	Execution time	FSGS	20.58	150.46	1093.66	7840.69
		FSAGE	3.33	24.62	180.76	1322.73
		HSAGE	0.48	3.31	24.32	181.40
	Hausdorff Distance	FSGS	2.7921e-03	2.7850e-03	2.7603e-03	2.6620e-03
		FSAGE	2.7938e-03	2.7921e-03	2.7888e-03	2.7761e-03
		HSAGE	2.7976e-03	2.7938e-03	2.7921e-03	2.7888e-03

Table 5: Numerical results of FSGS, FSAGE and HSAGE methods at $\alpha = 1.00$

		Methods	*n*			
			512	1024	2048	4096
Test Problem 1	Number of iterations	FSGS	36736	130432	455695	1557218
		FSAGE	5251	19137	68795	243293
		HSAGE	1430	5251	19137	68795
	Execution time	FSGS	59.62	423.54	2964.55	20247.36
		FSAGE	9.91	72.07	518.22	3677.87
		HSAGE	1.38	9.84	71.59	519.8
	Hausdorff Distance	FSGS	3.1099e-04	3.0302e-04	2.7124e-04	1.4415e-04
		FSAGE	3.1331e-04	3.1230e-04	3.0833e-04	2.9249e-04
		HSAGE	3.1366e-04	3.1331e-04	3.1230e-04	3.0833e-04
Test Problem 2	Number of iterations	FSGS	12668	46321	168043	603363
		FSAGE	1763	6517	24037	87897
		HSAGE	487	1763	6517	24037
	Execution time	FSGS	20.60	150.67	1093.01	7845.54
		FSAGE	3.36	24.54	181.32	1322.75
		HSAGE	0.48	3.32	24.32	181.41
	Hausdorff Distance	FSGS	2.4816e-03	2.4747e-03	2.4499e-03	2.3517e-03
		FSAGE	2.4834e-03	2.4818e-03	2.4785e-03	2.4658e-03
		HSAGE	2.4867e-03	2.4834e-03	2.4818e-03	2.4785e-03

Table 6. Percentage gains for HSGS and QSGS methods compared to FSGS method

α	Methods	Test Problem 1		Test Problem 2	
		Execution time (%)	Number of iterations (%)	Execution time (%)	Number of iterations (%)
0.00	FSAGE	81.79 - 83.38	84.31 - 85.68	83.10 - 93.84	85.41 - 94.74
	HSAGE	97.40 - 97.71	95.56 - 96.10	97.68 - 99.09	96.01 - 98.55
0.25	FSAGE	81.89 - 83.36	84.34 - 85.69	83.17 - 83.76	85.42 - 86.08
	HSAGE	97.43 - 97.70	95.57 - 96.10	97.66 - 97.80	96.01 - 96.19
0.50	FSAGE	81.90 - 83.37	84.36 - 85.70	83.18 - 83.86	85.43 - 86.08
	HSAGE	97.44 - 97.70	95.58 - 96.11	97.69 - 97.79	96.01 - 96.19
0.75	FSAGE	81.85 - 83.37	84.37 - 85.70	83.13 - 83.81	85.43 - 86.08
	HSAGE	97.43 - 97.71	95.58 - 96.11	97.67 - 97.80	96.02 - 96.19
1.00	FSAGE	81.84 - 83.38	84.38 - 85.71	83.14 - 83.71	85.43 - 86.08
	HSAGE	97.43 - 97.69	95.58 - 96.11	97.67 - 97.80	96.02 - 96.19

5. Conclusions

In this paper, the performance of HSAGE method for the numerical solution of fuzzy heat equation has been investigated. The results show that HSAGE method is superior to FSGS and FSAGE methods, particularly in the aspect of number of iterations and execution time. Apart from the concept of full- and half-sweep iterations, further investigation based on quarter-sweep [9] iteration can also be considered in order to speed up the convergence rate of the iterative methods.

REFERENCES

[1]. *R. K. Mohanty* and *J. Talwar*, "Compact alternating group explicit method for the cubic spline solution of two point boundary value problems with significant nonlinear first derivative terms", in Mathematical Sciences, **vol. 6**, 2012, Art. 58

[2]. *Q. Feng*, "An alternating group explicit iterative method for solving four-order parabolic equations", in Applied Mathematical Sciences, **vol. 2**, 2008, pp. 2591-2595

[3]. *N. Bildik and S. Özlü*, "On the numerical solution of Helmholtz equation by alternating group explicit (AGE) methods", in Applied Mathematics and Computation, **vol. 163**, 2005, pp. 505-518

[4]. *Q. Feng and B. Zheng*, "High order alternating group explicit finite difference method for parabolic equations", in WSEAS Transactions on Mathematics, **vol. 8**, 2009, pp. 127-137

[5]. *A. R. Abdullah*, "The four point Explicit Decoupled Group (EDG) method: A fast Poisson solver", in International Journal of Computer Mathematics, **vol. 38**, 1991, pp. 61-70

[6]. *T. Allahviranloo*, "Difference methods for fuzzy partial differential equations", in Computational Methods in Applied Mathematics, **vol. 2**, 2002, pp. 233-242

[7]. *D. J. Evans and A. R. Ahmad*, "Comparison of SOR and AGE methods for solution of the two-point boundary value problem", in Advances in Engineering Software, **vol. 26**, 1996, pp. 101-110

[8]. *A. Farajzadeh, A. H. Pour and M. Amini*, "An explicit method for solving fuzzy partial differential equation", in International Mathematical Forum, **vol. 5**, 2010, pp. 1025-1036

[9]. *M. Othman and A. R. Abdullah*, "An efficient four points Modified Explicit Group Poisson solver", in International Journal of Computer Mathematics, **vol. 76**, 2000, pp. 203-217