Jurnal
Teknologi

Full Paper

IMPLEMENTATION OF THE HALF-SWEEP AOR ITERATIVE ALGORITHM FOR SPACE-FRACTIONAL DIFFUSION EQUATIONS

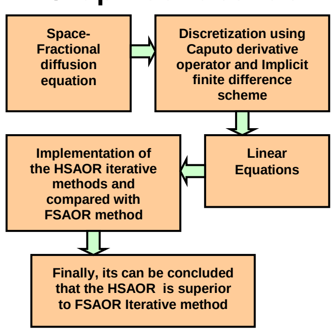
Andang Sunarto[a], Jumat Sulaiman[a*], Azali Saudi[b]

[a]Faculty of Science and Natural Resources, Universiti Malaysia Sabah, Malaysia
[b]Faculty of Computing and Informatics, Universiti Malaysia Sabah, Malaysia

Article history
Received
2 July 2015
Received in revised form
8 December 2015
Accepted
17 January 2016

*Corresponding author
jumat@ums.edu.my

Graphical abstract



Abstract

In this paper, we consider the numerical solution of one dimensional space-fractional diffusion equation. The half-sweep AOR (HSAOR) iterative method is applied to solve linear system generated from discretization of one dimensional space-fractional diffusion equation using Caputo's derivative operator and half-sweep implicit finite difference scheme. Furthermore, the formulation and implementation of HSAOR iterative method to solve the problem are also presented. Two examples and comparisons with FSAOR iterative method are given to show the effectiveness of the proposed method. From numerical results obtained, it has shown that the HSAOR iterative method is superior as compared with the FSAOR methods.

Keywords: HSAOR, space-fractional, caputo, implicit finite difference scheme

1.0 INTRODUCTION

In this paper we focus on numerical solution for one - dimensional space-fractional diffusion equations. Generally, linear space-fractional diffusion equations (SFDE's) given as follows

$$\frac{\partial U(x,t)}{\partial t} = a(x)\frac{\partial^\beta U(x,t)}{\partial x^\beta} + b(x)\frac{\partial U(x,t)}{\partial x} + c(x)U(x,t) + f(x,t) \quad (1)$$

with initial condition

$U(x,0) = f(x), \; 0 \le x \le \ell,$

and boundary conditions

$U(0,t) = g_0(t), \; U(\ell,t) = g_1(t), \; 0 < t \le T.$

We describe some necessary definitions and mathematical preliminaries of the fractional derivative theory which are required for our subsequent development of the approximation equation for the problem in Eq.(1).

Definition 1.[1,2] The Riemann-Liouville fractional integral operator , J^β of order- β is defined as

$$J^\beta f(x) = \frac{1}{\Gamma(\beta)}\int_0^x (x-t)^{\beta-1} f(t)dt, \; \beta > 0 \; x > 0 \quad (2)$$

Definition 2.[2,3] The Caputo's fractional partial derivative operator, D^β of order - β is defined as

$$D^\beta f(x) = \frac{1}{\Gamma(m-\beta)}\int_0^x \frac{f^{(m)}(t)}{(x-t)^{\beta-m+1}}dt, \; \beta > 0 \quad (3)$$

with $m-1 < \beta \le m, \; m \in N, \; x > 0$.

We have the following properties when $m-1 < \beta \le m, \; x > 0$:

$D^\beta k = 0$, (k is constant),

$$D^{\beta}x^{n}=\begin{cases}0, & \text{for } n\in N_0 \text{ and } n<\lceil\beta\rceil\\ \dfrac{\Gamma(n+1)}{\Gamma(n+1-\beta)}x^{n-\beta}, & \text{for } n\in N_0 \text{ and } n\geq\lceil\beta\rceil\end{cases}$$

where function $\lceil\beta\rceil$ denotes the smallest integer greater than or equal to β, $N_0=\{0,1,2,\ldots\}$ and $\Gamma(.)$ is the gamma function.

The linear space-fractional diffusion equations occur in multiple diversified phenomena such as physics, finance and biology problems. Therefore numerical treatment is preferred in order to diagnose and solve the problems. In many application areas, it is necessary to use the numerical approach to obtain an approximation solution for the problem in Eq. (1) such as method of line (MOL) [4], implicit finite element method [5], grid-based schemes and Monte-Carlo method [6]. Based on extension work [7], in this paper, discretization scheme based on Caputo's fractional derivative operator together with implicit finite difference scheme will be implemented to discretize the problem in Eq.(1).Thus, the generated linear system will be solved by using Half-Sweep AOR (HSAOR) iterative method.

Basically, the proposed HSAOR method is inspired by the concept of half-sweep iteration which is introduced by Abdullah [8] via the Explicit Decoupled Group (EDG) iterative method to solve two-dimensional Poisson equations. Actually, The half-sweep iteration concept are essential to reduce the computational complexities during iterative process, because the implementation of half-sweep iteration will only consider nearly half of all node point in a solution domain respectively [9]. In addition to the advantage of this iteration concept, the implementations of this concept in various partial differential equations were further investigated [10, 11, 12]. However, most of problems which have been solved by them are categorized as partial differential equation of integer order. In this work, we descretized space-fractional diffusion equation using implicit finite difference scheme with Caputo's derivative operator in order to examine the implementation of HSAOR iteration method in solving the resultant linear system of equations. The standard AOR iterative method also known as the FSAOR iterative method is implemented as control method in order to investigate the performance of HSAOR iterative method.

2.0 HALF-SWEEP CAPUTO'S IMPLICIT FINITE DIFFERENCE APPROXIMATION EQUATIONS

In this section, the space-fractional diffusion equation (1) is solved. In order to find solution in Eq. (1), let us define $h=\dfrac{\ell}{m+1}$, where, $m=n+1$ is positive even integer.

By implementing definition (2) we obtain

$$\begin{aligned}\frac{\partial^{\beta}U(x_i,t_n)}{\partial x^{\beta}}&=\frac{1}{\Gamma(2-\beta)}\int_0^{t_n}\frac{\partial^2U(x_i,s)}{\partial x^2}(t_n-s)^{1-\beta}\partial s\\&=\frac{1}{\Gamma(2-\beta)}\sum_{j=0,2,4}^{i-2}\int_{jh}^{(j+1)h}\left(\frac{U_{i-j+2,n}-2U_{i-j,n}+U_{i-j-2,n}}{2h^2}\right)(nh-s)^{\beta}\partial s\\&=\frac{(2h)^{-\beta}}{\Gamma(3-\beta)}\sum_{j=0,2,4}^{i-2}\left(U_{i-j+2,n}-2U_{i-j,n}+U_{i-j-2,n}\right)\left(\left(\frac{j}{2}+1\right)^{2-\beta}-\frac{j}{2}^{2-\beta}\right)\end{aligned}\quad(4)$$

Then the discrete approximation equation (4) can be written as

$$\frac{\partial^{\beta}U(x_i,t_n)}{\partial x^{\beta}}=\sigma_{\beta,2h}\sum_{j=0,2,4}^{i-2}g_j^{\beta}\left(U_{i-j+2,n}-2U_{i-j,n}+U_{i-j-2,n}\right)\quad(5)$$

where $\sigma_{\beta,2h}=\dfrac{(2h)^{-\beta}}{\Gamma(3-\beta)}$ and $g_j^{\beta}=\left(\dfrac{j}{2}+1\right)^{2-\beta}-\dfrac{j}{2}^{2-\beta}$.

Then, using implicit finite difference scheme and Caputo's derivative operator in Eq. (4), we obtain half-sweep Caputo's implicit finite difference approximation equation as

$$\begin{aligned}\lambda\left(U_{i,n}-U_{i,n-2}\right)=&\,a_i\sigma_{\beta,2h}\sum_{j=0.2.4}^{i-2}g_j^{\beta}\left(U_{i-j+2,n}-2U_{i-j,n}+U_{i-j-2,n}\right)\\&+b_i\frac{\left(U_{i+2,n}-U_{i-2,n}\right)}{4h}+C_iU_{i,n}+f_{i,n}\end{aligned}\quad(6)$$

for $i = 2,4,\ldots,m-2$. To simplify the above approximation equation, we get

$$\begin{aligned}\lambda U_{i,n-2}=&-a_i\sigma_{\beta,2h}\sum_{j=0,2,4}^{i-2}g_j^{\beta}\left(U_{i-j+2,n}-2U_{i-j,n}+U_{i-j-2,n}\right)\\&-\frac{b_i}{4h}\left(U_{i+2,n}-U_{i-2,n}\right)-C_iU_{i,n}+\lambda U_{i,n}-f_{i,n}\end{aligned}\quad(7)$$

Again, Eq.(7) can be shown

$$\begin{aligned}\therefore\ &b_i^*U_{i-2,n}+\left(\lambda-c_i^*\right)U_{i,n}-b_i^*U_{i+2,n_i}\\&-a_i^*\sum_{j=0,2,4}^{i-2}g_j^{\beta}\left(U_{i-j+2,n}-2U_{i-j,n}+U_{i-j-2,n}\right)=f_i\end{aligned}\quad(8)$$

where

$a_i^*=a_i\sigma_{\beta,2h}$,

$b_i^*=\dfrac{b_i}{4h}$,

$c_i^*=c_i$,

$F_i^*=f_{i,n}$,

$f_i=\lambda\left(U_{i,n-2}\right)+F_i^*$.

Let the series term in Eq. (8) be expanded to get the following approximation equation

$$-R_i+\alpha_iU_{i-6,n}+s_iU_{i-4}+p_iU_{i-2,n}+q_iU_{i,n}+r_iU_{i+2,n}=f_i\quad(9)$$

Where

$$R_i = a_i^* \sum_{j=6}^{i-2} g_j^{\beta} \left(U_{i-j+2,n} - 2U_{i-j,n} + U_{i-j-2,n} \right),$$

$$\alpha_i = \left(-a_i^* g_2^{\beta} \right),$$

$$s_i = \left(-a_i^* g_1^{\beta} + 2a_i^* g_2^{\beta} \right),$$

$$p_i = \left(b_i^* - a_i^* g_2^{\beta} + 2a_i^* g_1^{\beta} - a_i^* \right),$$

$$q_i = \left(-a_i^* g_1^{\beta} + 2a_i^* + \left(\lambda - c_i^* \right) \right),$$

$$r_i = \left(-a_i^* - b_i^* \right).$$

By considering the interior node points of the solution domain in to Eq.(9), the generated linear system by half-sweep Caputo's implicit finite difference approximation equation can be easily shown as

$$A \underset{\sim}{U} = \underset{\sim}{f} \qquad (10)$$

where

$$A = \begin{bmatrix} q_2 & r_2 \\ p_4 & q_4 & r_4 \\ s_6 & p_6 & q_6 & r_6 \\ \alpha_8 & s_8 & p_8 & q_8 & r_8 \\ & \alpha_{10} & s_{10} & p_{10} & q_{10} & r_{10} \\ & & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & & \alpha_{m-4} & s_{m-4} & p_{m-4} & q_{m-4} & r_{m-4} \\ & & & & \alpha_{m-2} & s_{m-2} & p_{m-2} & q_{m-2} \end{bmatrix}_{\left(\frac{m}{2}-1\right) x \left(\frac{m}{2}-1\right)}$$

$$\underset{\sim}{U} = \left[U_{2,1} \quad U_{4,1} \quad U_{6,1} \quad \cdots \quad U_{m-4,1} \quad U_{m-2,1} \right]^T$$

$$\underset{\sim}{f} = \left[f_2 - p_2 U_{2,1} \quad f_4 + s_4 U_{41} \quad f_6 + \alpha_6 U_{6,1} \quad f_8 + R_i \quad \cdots \quad f_{m-4,1} + R_{m-4} \quad f_{m-2,1} - p_{m-2} U_{m,1} + R_{m-2} \right]^T$$

3.0 FORMULATION OF HSAOR ITERATIVE METHOD

In this paper, FSAOR and HSAOR iterative methods will be applied to solve linear system generated from the discretization of the problem in Eq. (1) as shown in Eq. (10). In previous studies, a lot of works have been done to describe the various iterative methods such as Young [13, 14, 15], Hackbush [16], Saad [17], Evans [18] and Yousif and Evans [19]. To derive the formulation of both proposed methods, let the coefficient matrix A in Eq. (10) be expressed as

$$A = D - L - V \qquad (11)$$

where D, L and V are diagonal, strictly lower triangular and strictly upper triangular matrices respectively [13,14,15]. Then, based on Eq. (11)the general scheme for the HSAOR iterative method can be shown as [7,10,20,21]

$$\tilde{U}^{(k+1)} = (D - \omega L)^{-1} \left[\beta V + (\beta - \omega) L + (1 - \beta) D \right] \tilde{U}^{(k)} + \beta (D - \omega L)^{-1} f \qquad (12)$$

where $\tilde{U}^{(k)}$ represents an unknown vector at k^{th} iteration. Since $\beta = \omega$, this iterative method (12) will be named as the HSSOR [12,22,23,24], while $\beta = \omega = 1$ we obtain the HSGS method. Basically, the general algorithm for HSAOR iterative method to solve linear system (10) would be generally described in Algorithm 1.

Algorithm 1: HSAOR method

i. Initialize $\tilde{U} \leftarrow 0$ and $\varepsilon \leftarrow 10^{-10}$.

ii. For $j = 0,1,2,\ldots,n-1$ implement

a. For $i = 1p, 2p, \ldots, m-p$ calculate

$$\tilde{U}^{(k+1)} = (D - \omega L)^{-1} \left[\beta V + (\beta - \omega) L + (1 - \beta) D \right] \tilde{U}^{(k)} + \beta (D - \omega L)^{-1} f$$

b.Convergence test. If the convergence criterion i.e $\left\| \tilde{U}^{(k+1)} - \tilde{U}^{(k)} \right\| \le \varepsilon = 10^{-10}$ is satisfied, go to next time level. Otherwise go back to Step (a).

iii Display approximate solutions.

However If p=1, Algorithm 1 will be named as FSAOR

4.0 CONVERGENCE OF AOR METHOD

In this section we will show that convergence of AOR method .We have AOR method for the solution (1) has the form:

$$\tilde{U}^{(k+1)} = (D - \omega L)^{-1} \left[\beta V + (\beta - \omega) L + (1 - \beta) D \right] \tilde{U}^{(k)} + \beta (D - \omega L)^{-1} f \qquad (13)$$

with *n =0,1,2,…,* where

$$L_{\omega,\beta} = (D - \omega L)^{-1} \left[(1 - \beta) D + (\beta - \omega) L + \beta V \right] = D - \beta (D - \omega L)^{-1} A \qquad (14)$$

Theorem 4.1. [25](a) If the AOR method (13) converges $\left(\rho(L_{\omega,\beta}) < 1 \right)$ for some $\beta, \omega \neq 0$, then exactly one of the following statement hold:

(i). $\omega \in (0,2)$ and $\beta \in (-\infty,0) \cup (0,+\infty)$,

(ii). $\omega \in (-\infty,0) \cup (2,+\infty)$ and $\beta \in \left(\frac{2\omega}{(2-\omega)}, 0 \right) \cup (0,2)$

(b) If the AOR method with $\omega = 0$ converges $\left(\rho(L_{0,\beta}) < 1 \right)$ then $0 < \beta < 2$.

Proof. (a). It is known that the eigenvalues λ_j of $L_{\omega,\beta}$ $(\beta, \omega \neq 0)$ are connected with the eigenvalues ξ_j of $L_{\omega,\omega} \equiv L_\omega$ (L_ω is the SOR iteration matrix) by the relationship [20]

$$\lambda_j = \left(1 - \frac{\beta}{\omega} \right) + \frac{\beta}{\omega} \xi_j, \quad j = 2(2)m-2. \qquad (15)$$

From (15) we get $\xi_j = 1 - \frac{\omega}{\beta} + \frac{\omega}{\beta} \lambda_j \quad j = 2(2)m-2$. We also that $\prod_{j=2,4,..}^{m-2} \xi_j = (1-\omega)^n$. Therefor $\prod_{j=2,4,..}^{m-2} \left(1 - \frac{\omega}{\beta} + \frac{\omega \lambda_j}{\beta} \right) - (1-\omega)^n$ and since $\left| \lambda_j \right| < 1, \quad j = 2(2)m-2$ from hypothesis, we obtain

$$\left|(1-\omega)^n\right| = \prod_{j=2,4,..}^{m-2}\left|1-\frac{\omega}{\beta}+\frac{\omega}{\beta}\lambda_j\right| \le \prod_{j=2,4..}^{m-2}\left(\left|1-\frac{\omega}{\beta}\right|+\left|\frac{\omega}{\beta}\right|\left|\lambda_j\right|\right)$$

$$< \prod_{j=2,4,..}^{m-2}\left(\left|1-\frac{\omega}{\beta}\right|+\left|\frac{\omega}{\beta}\right|\right) = \left(\left|1-\frac{\omega}{\beta}\right|+\left|\frac{\omega}{\beta}\right|\right)^n$$, that is $|1-\omega| < \left|1-\frac{\omega}{\beta}+\left|\frac{\omega}{\beta}\right|\right|$,

or equivalently $|\beta(1-\omega)| < |\beta-\omega|+|\omega|$. (16)

It can be shown (16) hold if and only if exactly one of the following statement hold:

(i). $\omega \in (0,2)$ and $\beta \in (-\infty,0)\cup(0,+\infty)$,

(ii). $\omega \in (-\infty,0)\cup(2,+\infty)$ and $\beta \in \left(\frac{2\omega}{(2-\omega)},0\right)\cup(0,2)$

and proof of part (a) is completed.

(b). If $\omega = 0$, then $L_{0,\beta} = (1-\beta)D+\beta(L+U) = (1-\beta)D+\beta B.$ If $\mu_j,\ \ j=2(2)m-2$ are the eigenvalues of B, then for the eigenvalues λ_j of $L_{0,\beta}$ we get

$$\lambda_j = 1-\beta+\beta\mu_j,\quad j=2(2)m-2, \tag{17}$$

with imply $\mu_j = \frac{1}{\beta}(\beta-1+\lambda_j),\quad j=2(2)m-2$ (18)

But, since tr *B=0* we get

$$\sum_{j=2,4,..}^{m-2}\mu_j = 0 = \sum_{j=2,4,..}^{m-2}\frac{1}{\beta}(\beta-1+\lambda_j) \tag{19}$$

From (19) we have $\sum_{j=2,4,..}^{m-2}\lambda_j = \left(\frac{m}{2}-1\right)(1-\beta)$

and consequently

$\left|\left(\frac{m}{2}-1\right)(1-\beta)\right| = \left|\sum_{j=2,4,..}^{m-2}\lambda_j\right| \le \left|\sum_{j=2,4,..}^{m-2}|\lambda_j| < n,\right|$ since $|\lambda_j|<1,\quad j=2(2)m-2$ from the hypothesis. Therefore $\left|\left(\frac{m}{2}-1\right)(1-\beta)\right| < n,$ or equivalently $0<\beta<2.$

5.0 NUMERICAL EXPERIMENTS

For the numerical experiments, two examples were considered to verify the effectiveness of the implementation of the HSAOR iterative method. To comparison between FSAOR and HSAOR methods, three criteria will be considered such as number of iterations (K), execution time (second) and maximum error at three different values of $\beta=1.2, \beta=1.5$ and $\beta=1.8$ with different mesh sizes as 128, 256, 512, 1024 and 2048. In implementations of two numerical experiments, the convergence test considered the tolerance error $\varepsilon=10^{-10}$. Results of numerical experiments, which were obtained from implementations of the FSAOR and HSAOR iterative method have been recorded in Tables 1 and 2 respectively.

Tables 1 Comparison between number of iterations (K), the execution time (seconds) and maximum errors for the iterative methods using example 1 at $\beta=1.2,1.5,1.8$

M	Method	β = 1.2			β = 1.5			β = 1.8		
		K	Time	Max Error	K	Time	Max Error	K	Time	Max Error
128	FSAOR	65	1.32	2.37e-02	188	3.88	6.21e-04	269	5.35	3.99e-02
	HSAOR	46	0.53	2.24e-02	78	0.83	6.99e-04	225	2.13	4.03e-02
256	FSAOR	128	10.00	2.44e-02	370	28.88	5.69e-04	756	58.90	3.97e-02
	HSAOR	77	2.94	2.37e-02	204	7.70	6.21e-04	732	28.08	3.99e-02
512	FSAOR	270	84.05	2.47e-02	983	104	5.35e-04	2497	703	3.96e-02
	HSAOR	129	19.88	2.44e-02	544	83.61	5.69e-04	2388	368.65	3.97e-02
1024	FSAOR	577	125	2.49e-02	3640	689	5.13e-04	5220	1119	2.36e-02
	HSAOR	278	179.11	2.47e-02	1457	502	5.35e-04	4098	982	3.38e-02
2048	FSAOR	1150	540	2.52e-02	5950	3102	5.09e-04	13203	3920	2.30e-02
	HSAOR	606	424	2.49e-02	3885	2035	5.24e-04	11376	3256	2.35e-02

Table 2 Comparison between number of iterations (K), the execution time (seconds) and maximum errors for the iterative methods using example 2 at $\beta = 1.2, 1.5, 1.8$

M	Method	β = 1.2			β = 1.5			β = 1.8		
		K	Time	Max Error	K	Time	Max Error	K	Time	Max Error
128	**FSAOR**	48	0.93	1.80e-01	133	1.41	5.44e-02	148	1.52	1.25e-04
	HSAOR	34	0.45	1.73e-01	55	0.70	5.16e-02	135	1.24	1.76e-04
256	**FSAOR**	97	3.58	1.84e-01	197	10.93	5.58e-02	457	16.66	1.44e-04
	HSAOR	55	2.67	1.81e-01	145	6.91	5.44e-02	439	11.61	8.88e-04
512	**FSAOR**	106	18.71	5.39e-01	525	83.02	1.28e-02	1357	193.83	1.53e-04
	HSAOR	97	17.52	1.84e-01	386	73.38	5.58e-02	1147	101.20	4.09e-04
1024	**FSAOR**	213	168	5.45e-01	1298	198	1.32e-02	4329	2103	1.25e-04
	HSAOR	209	150.23	1.86e-01	1030	160	5.65e-02	3731	1984.23	1.54e-04
2048	**FSAOR**	815	398	1.92e-01	2506	912	5.73e-02	6520	3834	2.30e-04
	HSAOR	456	273	1.86e-01	2326	878	5.80e-02	6290	3462	2.45e-04

Example 1: [3]

We consider the following space-fractional initial boundary value problem

$$\frac{\partial U(x,t)}{\partial t} = d(x)\frac{\partial^{\beta} U(x,t)}{\partial x^{\beta}} + p(x,t), \tag{20}$$

at finite domain $0 \le x \le 1$, with the diffusion $d(x) = \Gamma(\beta)x^{0.5}$. The source function $p(x,t) = (x^2+1)\cos(t+1) - 2x\sin(t+1)$, with the initial condition $U(x,0) = (x^2+1)\sin(1)$ and the boundary conditions $U(0,t) = \sin(t+1)$, $U(l,t) = 2\sin(t+1)$, for t>0. The exact solution of this problem is $U(x,t) = (x^2+1)\sin(t+1)$.

Examples 2: [3]

We consider the following space-fractional initial boundary value problem

$$\frac{\partial U(x,t)}{\partial t} = \Gamma(1.2)x^{\beta}\frac{\partial^{\beta} U(x,t)}{\partial x^{\beta}} + 3x^2(2x-1)e^{-t}, \tag{21}$$

with the initial condition $U(x,0) = x^2 - x^3$ and zero Dirichlet conditions. The exact solution of this problem is $U(x,t) = x^2(1-x)e^{-t}$.

6.0 CONCLUSION

In this work, we discussed the implementation of the HSAOR iterative algorithm which uses two accelerated parameter. The HSAOR Algorithm has performance good speedup and efficiency for computational time and number of iterations. Again, the HSAOR algorithm has shown their superiority over the FSAOR algorithm. For our future works, this study can be extended to investigate on the use of the AOR to combined with the concept quarter-sweep iterative family [26, 27, 28].

References

[1] Zhang, Y. 2009. A Finite Difference Method for Fractional Partial Differential Equation. *Applied Mathematics and Computation*. 215: 524-529.

[2] Li, C., Qian, D., and Chen, Y. Q. 2011. On Riemann-Liouville and Caputo Derivatives. *Hindawi Publishing Corporation Discrete Dynamics in Nature and Science*. 1: 1-15.

[3] Azizi, H., and Loghmani, G. B. 2013. Numerical Approximation for Space-Fractional Diffusion Equations via Chebyshev Finite Difference Method. *Journal of Fractional and Applications*. 4(2): 303-311.

[4] Liu, F., Anh, V., and Turner, I. 2004. Numerical Solution of The Space Fractional Fokker-Planck Equation. *Journal of Computational And Applied mathematics.* 166: 209-219.

[5] Burrage, K., Hale, N. and Kay, D. 2012. An Efficient Implicit FEM Scheme for Fractional-In-Space Reaction-Diffusion Equations. *Society for Industrial and Applied mathematics*. 34(4): A2145-A2172.

[6] Stern, R., Effenberger, F., Fichtner, H., and Schafer, T. 2013. The Space-Fractional Diffusion–advection Equation: Analytical Solutions And Critical Assessment of Numerical Solutions. *Fractional Calculus and Applied Analysis.* 17(1): 171-190.

[7] Sunarto, A., Sulaiman, J., and Saudi, A. 2014. Implicit Finite Difference Solution for Time-Fractional Diffusion Equations Using AOR Method. *Journal of Physics Conference Series*. 495: 2024-2031.

[8] Abdullah, A. R. The Four Point Explicit Decoupled Group (EDG) Method: A Fast Poisson Solver. *International Journal Computer Mathematics.* 76: 203-217.

[9] Hasan, M. K., Othman, M., Abbas, Z., Sulaiman, J., and Ahmad, F. 2007. Parallel Solution of High Speed Low Order FDTD on 2D Free Space Wave Propagation. *Lecturer Notes in Computer Science LNCS*. 4706: 13-24.

[10] Sunarto, A., Sulaiman, J., and Saudi, A. 2014. Half-Sweep Accelerated Over-Relaxations Iterative Method for The Solution Time-Fractional Diffusion Equations. *Simposium Kebangsaan Sains Mathematiks ke 22*. Shah-Alam, Malaysia. 109-115.

[11] Sunarto, A., Sulaiman J., and Saudi, A. 2014. Half-Sweep Gauss-Seidel Iteration Applied to Time-Fractional Diffusion Equations. *Journal Kalam*. 7(2): 016-022.

[12] Saudi, A, and J. Sulaiman. 2012. Path Planning for Indoor mobile Robot using Half-Sweep SOR via Nine-Point Laplacian (HSSORL9L). *IOSR Journal of Mathematics*. 3(2): 01-07.

[13] Young, D. M. 1954. Iterative Methods for Solving Partial Difference Equations of Elliptic Type. *Transaction of The AMS-American Mathematical Society*. 76: 92-111.

[14] Young, D. M. 1971. *Iterative Solution of Large Sparse Systems*. London: Academic Press.

[15] Young, D. M. 1972. Second-Degree Iterative Methods for The Solution of Large Linear Systems. *Journal of Approximation Theory*. 15: 37-148.

[16] Hackbush, W. 1995. *Iterative Solution of Large Sparse Systems of Equations.* New York: Springer-Verlag.

[17] Saad, Y. 1996. *Iterative Method for Sparse Linear Systems.* Boston: International Thomas Publishing.

[18] Evans, D. J. 1985. Group Explicit Iterative Methods for Solving Large Linear Systems. *International Journal Computer Mathematics.* 17: 81-108.

[19] Yousif, W., and Evans, D. J. 1995. Explicit De-coupled Group Iterative methods and Their Implementations. *Parallel Algorithm and Applications.* 7: 53-71.

[20] Hadjidimos, A. 1978. *Accelerated Over Relaxation Method. Mathematics of Computation.* 32: 149-157

[21] Tian, H. 2003. Accelerated Over-relaxation Method for Rank Deficient Linear Systems. *Applied Mathematics and computation*. 14: 485-499.

[22] Sunarto, A., Sulaiman, J., and Saudi, A. 2013. SOR Method for Implicit Finite Difference Solution of Time-Fractional Diffusion Equations. *Borneo Science.* 34: 34-42.

[23] Sunarto, A., Sulaiman, J. and Saudi, A. 2014. Full-Sweep SOR Method for Solving Space-Fractional Diffusion Equations. *Australian journal of Basic and Applied Science*. 8: 153-158.

[24] Sunarto, A., Sulaiman, J. and Saudi, A. 2014. Solving The Time-Fractional Diffusion Equations By The Half-Sweep SOR Method. *Proceeding of International Conference of Advanced Informatics: Concept, Theory and Applications* (ICAICTA), Bandung, Indonesia: 20-21 August 2014. 1: 272-277.

[25] Yeyios, A. K. 1988. A Necessary Condition for The Convergence of The Accelerated Overrelaxation (AOR) *method. Journal of Computational and Applied Mathematics.* 26: 371-373.

[26] Sunarto, A., Sulaiman J. and Saudi, A. 2014. Approximate Solution for The Time-Fractional Diffusion Equations Using Quarter-Sweep Gauss-Seidel Method. *Proceeding of The 1st International Conference on Science and Technology for Sustainability (ICoStech)*, Batam, Indonesia. 22 October 2014. 1: 15-21.

[27] Sunarto, A., Sulaiman, J. and Saudi, A. 2015. Numerical Solution of The Time–Fractional Diffusion Equations By Using Quarter-Sweep SOR Iterative Method. *International of Journal Mathematical Engineering and Science (IJMES).* 3(2): 54-67.

[28] Sulaiman, J., Othman, M., and Hasan, M. K. 2004. Quarter-Sweep Iterative Alternating Decomposition Explicit Algorithm Applied to Diffusion Equations. *International Journal Computer Mathematics.* 81(12): 1559-1565.